DSG-Net: Learning Disentangled Structure and Geometry for 3D Shape Generation

JIE YANG*, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences
KAICHUN MO*, Stanford University
YU-KUN LAI, Cardiff University
LEONIDAS J. GUIBAS, Stanford University
LIN GAO†, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences

Disentangled Interpolation

Generated Shapes

Fig. 1. Our deep generative network DSG-Net encodes 3D shapes with complex structure and fine geometry in a representation that leverages the synergy between geometry and structure, while disentangling these two aspects as much as possible. This enables novel modes of controllable generation for high-quality shapes. Left: results of disentangled interpolation. Here, the top left and bottom right chairs (highlighted with red rectangles) are the input shapes. The remaining chairs are generated automatically with our DSG-Net, where in each row, the structure of the shapes is interpolated while keeping the geometry unchanged, whereas in each column, the geometry is interpolated while retaining the structure. Right: shape generation results with complex structure and fine geometry details by our DSG-Net. We show close-up views in dashed yellow rectangles to highlight local details.

3D shape generation is a fundamental operation in computer graphics. While significant progress has been made, especially with recent deep generative models, it remains a challenge to synthesize high-quality shapes with rich geometric details and complex structures, in a controllable manner. To tackle this, we introduce DSG-Net, a deep neural network that learns a disentangled structured & geometric mesh representation for 3D shapes, where two key aspects of shapes, geometry and structure, are encoded in a synergistic manner to ensure plausibility of the generated shapes, while also being disentangled as much as possible. This supports a range of novel shape generation applications with disentangled control, such as interpolation of structure (geometry) while keeping geometry (structure) unchanged. To achieve this, we simultaneously learn structure and geometry through variational autoencoders (VAEs) in a hierarchical manner for both, with bijective mappings at each level. In this manner, we effectively encode geometry and structure in separate latent spaces, while ensuring their compatibility: the structure is used to guide the geometry and vice versa. At the leaf level, the part geometry is represented using a conditional part VAE, to encode high-quality geometric details, guided by the structure context as the condition. Our method not only supports controllable
3D shapes are widely used in computer graphics and computer vision, with applications ranging from modeling, and recognition to rendering. Synthesizing high-quality shapes is therefore highly demanded for many downstream applications. Ideally, the synthesized shapes should be able to contain fine geometric details and complex structures, and the generation process needs to provide high-level control to ensure desired shapes are produced.

Shape generation has been extensively researched in recent years, benefiting especially from the capabilities of deep generative models. This has been true across a variety of 3D representations used to represent generated shapes, including point clouds, voxels, implicit fields, meshes, etc. However, existing methods still have limitations in representing both complex shape structure as well as geometry details, which is what is required for many downstream applications.

Moreover, for high-level control in shape generation, it is important to decompose shapes into multiple aspects that can be independently manipulated—typically geometry and structure (i.e. how different parts are related to form the overall shape). On the one hand, geometry and structure are synergistic: the structure of an object may restrict the specific geometric shapes that are plausible, and vice versa. On the other hand, to support high-level control, it is beneficial to derive a representation that disentangles these two aspects as much as possible. Such disentangled and synergistic representations offer significant benefits, including controlling generation of new shapes, e.g. interpolating or transferring structure while keeping geometry unchanged, or manipulating geometry while retaining the structure.

Disentangled representations have been widely studied in image generation, allowing different aspects, such as different facial attributes (e.g., expression, age, gender) to be manipulated separately, either in supervised [Xiao et al. 2018a,b] or unsupervised [Chen et al. 2016] manners. For disentanglement of 3D shapes, existing works either focus on specific data kinds such as human faces [Abrevaya et al. 2019], or are restricted to intrinsic/extrinsic decomposition, where shape geometry and poses are considered [Aumentado-Armstrong et al. 2019]. None of these methods can handle more general shape geometry and structure disentanglement.

Most existing deep shape generation works produce synthesized shapes as a whole. This makes it particularly difficult to control the generation, either in a topology or geometry-aware manner. Recently, some pioneering works have addressed this shortcoming by considering shape generation using parts and their compositions, leading to improved geometric detail [Gao et al. 2019b] and better handling of complex structure [Mo et al. 2019a]. However, neither is able to generate shapes with both complex structures and detailed geometry. Moreover, disentanglement of structure and geometry is not addressed in these works. A notable work SAG-Net [Wu et al. 2019] pioneers the study of 3D shape geometry and structure disentangled/controllable generation where an attention-based Gated Recurrent Unit (GRU) network is proposed to jointly encode shape geometry as voxel maps and structure as fully connected graphs into a single latent space. However, it remains challenging to deal with more fine-grained parts and hierarchical shape structures, as well as fine geometric details. And also, it achieves the controllable generation by the two-branch variational autoencoder (VAE) and optimization in a single latent space, rather than two independent latent spaces, which do not require optimizations for a disentangled generation.

In this paper, we introduce Disentangled Structure & Geometry Net (DSG-Net), a novel deep generative model which learns to generate high-quality shape meshes while disentangling the shape geometry and structure generation of two independent spaces as much as possible, enabling many novel disentangled shape manipulation applications, as shown in Figure 2. Our work uses the fine-grained shape part hierarchies in the PartNet [Mo et al. 2019b] dataset. In our disentangled shape representation, shape structure only includes the hierarchical part graphs with symbolic part semantics and relationships, whereas shape geometry contains the detailed part geometry. For our network design, we encode both the structure and geometry hierarchies with an n-ary tree using separate variational autoencoders (VAEs) with recursive neural network architectures. Both the geometry and structure information flows along the edges of hierarchical graphs and is aggregated into two latent spaces, allowing these two key aspects to be encoded separately in a disentangled manner. Furthermore, we design a Cycled Disentanglement mechanism to decompose the shape space into two separate spaces and further improve the performance of disentanglement. During the training, a new shape that combines the geometry and structure of any two input shapes will be synthesized, and our new design will encourage its geometry and structure to be the same as the two input shapes as much as possible. Compared to the original framework, it only decomposes the shape with two separate encoders without additional supervision. Our new designs are capable of improving the disentanglement performance efficiently. Though disentangled, the two latent spaces need to be correlated to ensure the plausibility of the generated shapes. We simultaneously train both structure and geometry VAEs while ensuring necessary communications between them: the geometry follows the part semantics and the inter-part relationship edges in the structure, while the structure requires knowing part geometry for training.

Our novel solution allows shapes with complex structure and delicate geometry to be represented and synthesized, outperforming state-of-the-art methods, e.g. [Gao et al. 2019b; Mo et al. 2019a]. The disentangled and synergistic formulation allows novel applications, such as shape generation and interpolation with separate control of structure and geometry. New shapes can also be synthesized by mixing structure and geometry from different examples.

Figure 1 demonstrates the capability of our DSG-Net to interpolate shapes with rich geometry and complex structure in the geometry.
and structure spaces, separately where each row shows interpolation of a structure while keeping geometry unchanged, and each column presents interpolation of geometry while retaining the same structure. Through extensive evaluations and comparisons with the state-of-the-art deep neural generative models, our method shows significant advantages and superiority in various shape categories. Our method supports traditional applications such as shape generation, synthesis, and interpolation, but now with independent control on the shape structure and geometry detail, facilitating the design process.

In summary, we make the following key contributions:
- We propose a novel DSG-Net that learns to decompose shape space into two disentangled latent spaces, encoding the geometry and structure of shapes respectively (as illustrated in Figure 2). Furthermore, we design a Cycled Disentanglement (CycD) mechanism to further disentangle the structure and geometry of a shape in a self-supervised manner;
- DSG-Net enables novel shape synthesis applications that exploit disentangled control of structure and geometry;
- DSG-Net also allows high-quality shapes with complex structure and fine geometric details to be effectively represented and synthesized, outperforming state-of-the-art methods.

2 RELATED WORK

3D shape generation is a key research topic in 3D computer vision and graphics. In this section, we give a brief review on recent advances in 3D shape representations, modeling 3D shape geometry and structure, as well as disentangled representation learning.

2.1 3D Shape Representations

In contrast to reaching a great consensus on representing 2D images as pixel grids, researchers have been exploring a big variation of representations for 3D data. Recent works have developed deep learning frameworks for 3D voxel grids [Choy et al. 2016; Maturana and Scherer 2015], multi-view 2D rendering of 3D data [Kalogerakis et al. 2017; Su et al. 2018, 2015], 3D point clouds [Achlioptas et al. 2018; Fan et al. 2017; Li et al. 2018; Lin et al. 2021; Qi et al. 2017a,b], 3D polygonal meshes [Chen et al. 2020; Groueix et al. 2018; Wang et al. 2018], and 3D implicit functions [Chen and Zhang 2019; Mescheder et al. 2019; Park et al. 2019]. For more detailed discussion and comparison, we refer the readers to these surveys [Ahmed et al. 2018; Bronstein et al. 2017; Ioannidou et al. 2017; Jin et al. 2020; Xiao et al. 2020; Yuan et al. 2021].

There is a recent trend of studying part-based and structure-aware 3D shape representations, since 3D shapes naturally exhibit compositional part structures. Part-based shape modeling decomposes complicated shapes into simpler parts for geometric modeling and organizes parts as part sequences or part hierarchies that encode shape part relationships and structures. Many previous works investigated parsing 3D shapes into parts [Huang et al. 2011; Tulsiani et al. 2017; Yi et al. 2017a; Zou et al. 2017], representing 3D shapes as part sequences or hierarchies [Ganapathi-Subramanian et al. 2018; Kim et al. 2013; Mo et al. 2019a; Niu et al. 2018; Van Kaick et al. 2013; Wang et al. 2011; Wu et al. 2020; Zhu et al. 2018a], and generating 3D shapes with part structures [Gao et al. 2019b; Kalogerakis et al. 2012; Li et al. 2017; Mo et al. 2020; Wu et al. 2019]. We refer to survey papers [Chaudhuri et al. 2020; Mitra et al. 2014; Xu et al. 2016] for more comprehensive discussion.

2.2 Modeling Shape Geometry

There are several different approaches to generating detailed 3D shape geometry: direct methods, patch-based methods, deformation-based methods, and others. Direct methods exploit decoder networks that output 3D contents in direct feed-forward procedures. For instance, Choy et al. [2016] and Tatarchenko et al. [2017] directly generate 3D voxel grids using 3D convolutional neural networks. Fan et al. [2017] and Achlioptas et al. [2018] use Multi-layer Perceptrons (MLPs) to directly generate 3D point clouds. Patch-based methods generate 3D shapes by assembling many local 3D surface patches. AtlasNet [Groueix et al. 2018] and Deprelle et al. [2019] learn to reconstruct each 3D shape by a collection of local surface elements or point clouds. Recent papers [Genova et al. 2019; Jiang et al. 2020] learn local implicit functions that are aggregated together to generate 3D shapes. Deformation-based methods train neural networks to deform an initial shape template to the output shape. For example, FoldingNet [Yang et al. 2018] and Pixel2Mesh [Wang et al. 2018] learn to deform 2D grid surfaces and 3D sphere manifolds to reconstruct 3D target outputs.

In our paper, we choose a deformation-based mesh representation for leaf-node parts, where we deform a unified unit cube mesh with 5,402 vertices to describe leaf-node part geometry. Representing 3D shapes as fine-grained part hierarchies [Mo et al. 2019a,b], we find that it is effective and efficient for preserving geometry details for leaf-node parts, as previously shown in the recent works [Gao et al. 2019a,b]. Different from SDM-Net [Gao et al. 2019b], we introduce a structure-conditioned part geometry VAE, that substantially improves data efficiency and reconstruction performance. Second, we build up bijective mappings between the structure and geometry nodes for synergistic joint learning, which enables disentangled representations for shape structure and geometry. Compared to StructureNet [Mo et al. 2019a] that generates 3D point clouds for leaf-node parts, we find our method generates 3D part geometry with sharper edges and more details.
2.3 Modeling Shape Structure

3D objects, especially man-made ones, are highly compositional and structured. Previous works attempt to infer the underlying shape grammars [Chaudhuri et al. 2011; Kalogerakis et al. 2012; Wu et al. 2016], part-based templates [Ganapathi-Subramanian et al. 2018; Kim et al. 2013; Ovsjanikov et al. 2011], and shape programs [Sharma et al. 2018; Tian et al. 2019]. Then, Elena et al. [2018] proposed a structure-aware and voxel-based shape synthesis model that respects structure constraints (landmark points), which are predicted by a learned structure detector. There are also many papers investigating generating shapes in the part-by-part manner using consistent part semantics [Dubrovina et al. 2019; Li et al. 2020; Schor et al. 2019; Wu et al. 2019] and sequential part instances [Sung et al. 2017; Wu et al. 2020].

Recently, researchers have been investigating representing shapes as part hierarchy, extending part granularity to more fine-grained scales. The pioneering work to encode tree structure of object GRASS [Li et al. 2017] uses binary part hierarchies and advocates to use recursive neural networks (RvNN) to hierarchically encode and decode parts along the tree structure. A follow-up work StructureNet [Mo et al. 2019a] further extends the framework to handle n-ary part hierarchies with consistent part semantics for an object category. [Mo et al. 2019b], SDM-NET [Gao et al. 2019a] learns to generate structured meshes with deformable parts by leveraging a part graph with rich support and symmetry relations. Sun et al. [2019] and Paschalidou et al. [2020] explore learning hierarchical part decompositions in unsupervised settings.

Our work adopts the hierarchical part representation introduced in StructureNet [Mo et al. 2019a] that can represent ShapeNet [Chang et al. 2015] shapes with complicated structures and fine-grained leaf-node parts. Different from StructureNet where shape geometry and structure are jointly modeled in one RvNN, we learn a pair of separate geometry RvNN and structure RvNN in a disentangled but synergistic fashion, which enables exploring geometric (structural) changes while keeping shape structure (geometry) unchanged. We also find that by combining the state-of-the-art structure learning modules from StructureNet [Mo et al. 2019a] and the latest techniques in modeling detailed part geometry from SDM-Net [Gao et al. 2019b] in an effective way, we achieve the best from both worlds that beats both StructureNet and SDM-Net in performance.

2.4 Disentangled Analysis in Deep Learning

For 2D image generation, the architecture proposed in [Karras et al. 2019] enables intuitive, scale-specific control of high-level attributes for high-resolution image synthesis by automatic unsupervised separation. HoloGAN [Nguyen-Phuoc et al. 2019] improves the visual quality of generation and allows manipulations by utilizing explicit 3D features to disentangle the shape and appearance in an end-to-end manner from unlabeled 2D images only.

In 3D shape processing, generative modeling becomes a mainstream topic thanks to deep learning and tremendous public 3D datasets, some of which contain rich realistic textures. Levinson et al. [2019] propose a supervised generative model to achieve accurate disentanglement of pose and shape in a large-scale human mesh dataset, as well as successfully incorporating techniques such as pose and shape transfer. Moreover, CFAN-VAE [Tatro et al. 2020] proposes to use the intrinsic conformal factor and extrinsic normal feature to achieve geometric disentanglement (pose and identity of human shapes) in an unsupervised way. For general textured objects datasets, VON [Zhu et al. 2018b] presents a fully differentiable 3D-aware generative model with a disentangled 3D representation (shape, viewpoint, and texture) for image and shape synthesis.

Compared to the above works, our work displays a rather novel capability - disentanglement of structure and geometry of 3D shapes. In this work, we learn a disentangled structured mesh representation for 3D shapes, where the disentanglement is entirely between two explicitly defined factors, namely structure and geometry. Furthermore, the cycle consistency first enables the translation of images and shapes with unpaired examples in an unsupervised manner [Gao et al. 2018; Yi et al. 2017b; Zhu et al. 2017]. For the enhancement of disentangled learning, we adopt the cycle consistency into our framework to explicitly encourage latent-space disentanglement. Our network can not only be used to generate shapes with improved geometric details but also allows us to exploit independent control of structure and geometry with the disentangled latent spaces.

3 METHODOLOGY

In this work, every 3D shape is decomposed into semantically consistent part instances that are organized by an n-ary part hierarchy covering parts at different granularities, ranging from coarse-grained parts (e.g. chair back and base) to fine-grained ones (e.g. back bars and legs). We propose a disentangled but synergistic hierarchical representation (Figure 3) and a learning framework with our Cycled Disentanglement mechanism (Figure 5), enabling disentangled control of shape geometry and structure in the shape generation procedure.

In the following subsections, we first describe the detailed definitions for our disentangled shape representation of structure hierarchy and geometry hierarchy. Then, we introduce a conditional part geometry VAE on encoding and decoding the fine-grained part geometry using a unified deformable mesh. Next, we present our disentangled network architecture designs for the geometry and structure VAEs and discuss how to learn the disentangled shape geometry and structure latent spaces simultaneously where the geometry and structure VAEs guide the learning processes for each other. Finally, a post-processing procedure is introduced for result refinement.

3.1 Disentangled Shape Representation

We adopt the hierarchical part segmentation in PartNet [Mo et al. 2019b] for ShapeNet models [Chang et al. 2015], where each shape is decomposed into a set of parts P and organized in a part hierarchy H (i.e., the vertical parent-child part relationships) with rich part relationships R (i.e., the horizontal among-sibling symmetry or adjacency part relationships). Each part \( P_i \) is associated with a semantic label \( l_i \) (e.g. chair back, chair leg) defined for a certain object class, as well as the detailed part geometry \( G_i \).

We introduce a disentangled but synergistic shape representation for shape structure and geometry, where we represent each 3D shape...
There is a bijective mapping between the tree nodes in the two hierarchies. We consider symbolic part semantics and a rich set of part relationships (orange arrows), such as adjacency (\( \tau_3 \)), translational symmetry (\( \tau_1 \)), reflective symmetry (\( \tau_2 \)) and rotational symmetry (\( \tau_0 \)). In the part geometry hierarchy, the part geometry is represented by meshes as a pair of a structured hierarchy and a geometry hierarchy. In our disentangled representation (see Figure 3), a structure hierarchy abstracts away the part geometry and only describes a symbolic part hierarchy with part structures and relationships, namely \((\{l_1, l_2, \ldots, l_N\}, H, R)\), while a geometry hierarchy describes the part geometry \((G_1, G_2, \ldots, G_N)\). There is a bijective mapping between the tree nodes of the structure and geometry hierarchies where the part semantic label \(l_i\) defined in the structure hierarchy corresponds to the part geometry \(G_i\) included in the geometry hierarchy. Also, the geometry hierarchy implicitly follows the same part hierarchy \(H\) and part relationships \(R\) as specified in the structure hierarchy.

**Part Geometry Representation.** For each part geometry \(G_i\), we use a mesh representation to capture more geometric details, such as the decorative patterns and sharp boundary edges, than the point cloud representation used in StructureNet [Mo et al. 2019a]. Given a closed box mesh manifold \(G_{\text{box}}\) with \(V = 5402\) vertices, we first calculate the oriented bounding box (OBB) \(B_i\) of each part \(P_i\) and deform \(G_{\text{box}}\), initialized with the shape \(B_i\), to the target part geometry \(G_i\) by adjusting the vertex positions through a non-rigid registration procedure. Then, for each part, we use the ACAP (as-consistent-as-possible) feature [Gao et al. 2019a,b] \(X_i\) as the representation of the deformed box mesh. The ACAP feature \(X_i \in \mathbb{R}^{5402}\) captures the local rotation and scale information in a one-ring neighbor patch of every vertex on the mesh and is capable of capturing large-scale local geometric deformations (e.g. rotation greater than 180°). We show an example registration result in Figure 4 (a). For the detailed calculation, please refer to the work [Gao et al. 2019a]. Since the ACAP feature is invariant to spatial translation of the part, we incorporate an additional 3-dimensional vector to describe the part center \(c_i\). Overall, each part geometry is represented as a pair of an ACAP feature \(X_i\) and a part center vector \(c_i\), as shown in Figure 4 (b), i.e., \(G_i = (X_i, c_i)\).

**Geometry Hierarchy.** The geometry hierarchy for a 3D shape is a hierarchy of part geometries \((G_1, G_2, \ldots, G_N)\) (Figure 3 right). It decomposes a complicated shape geometry into a hierarchy of parts ranging from coarse-grained levels to fine-grained levels. Each part geometry \(G_i\) in the geometry hierarchy corresponds to a tree node in the structure hierarchy and gives a concrete geometric realization given the context of the entire shape structure to generate. The geometry hierarchy implicitly follows the structural hierarchy and part relationships \(H\) and \(R\) defined in the structure hierarchy.

**Structure Hierarchy.** We consider a symbolic structure hierarchy \((\{l_1, l_2, \ldots, l_N\}, H, R)\) as the structure representation for a shape, inspired by a recent work PT2PC [Mo et al. 2020]. Figure 3 (left) presents an example for the symbolic structure hierarchy. It only includes the semantic information of shape parts and the relationships between parts, while abstracting away the concrete part geometry. PT2PC learns to generate 3D point cloud shapes conditioned on a given symbolic structure hierarchy as a fixed skeleton for shape generation. In this work, we extend PT2PC to consider encoding and decoding the symbolic structure hierarchy and investigate its disentangled but synergistic relationship to the geometry hierarchy.

In the symbolic structure hierarchy, we represent each part with a semantic label \(l_i\) (e.g. chair back, chair leg) without having a concrete part geometry in the representation. We include the rich sets of part relationships defined in the PartNet dataset in the symbolic structure hierarchy representation. There are two kinds of part relationships: the vertical parent-child inclusion relationships (e.g. a chair back and its sub-component chair back bars), as defined in \(H\), and the horizontal among-sibling part symmetry and adjacency relationships (e.g. chair back bars have translational symmetry), as denoted in \(R\). We use the part relationships \(H\) and \(R\) as provided in StructureNet [Mo et al. 2019a].

**Coupling Geometry and Structure Hierarchies.** Even though we are attempting a disentangled shape representation, the structure and geometry need to be compatible with each other for generating plausible and realistic shapes. On the one hand, shape structure
provides a high-level guidance for part geometry. If four legs of a chair are specified to be symmetric to each other in the structure hierarchy, the four legs should have identical part geometry to satisfy the structural requirement. On the other hand, given a certain type of part geometry, only certain kinds of shape structures are possible. For example, it is nearly impossible to manufacture a swivel chair if no lift handle or gas cylinder parts are provided.

Concretely, in our disentangled shape representation, the geometry hierarchy \((G_1, G_2, \cdots, G_N)\) and the structure hierarchy \((\langle l_1, l_2, \cdots, l_N \rangle, H, R)\) of a shape are highly correlated and tightly coupled. There is a bijective mapping between each part geometry node \(G_i\) and the part structure symbolic node \(l_i\). We set up communication channels between the two hierarchies in the joint learning process. The geometry hierarchy uses the part hierarchy \(H\) and relationship \(R\) in the encoding and decoding stages for passing messages and synchronizing geometry generation among related nodes. To train the decoding stage of the structure hierarchy, we leverage the corresponding geometry nodes to help match the prediction to the ground-truth parts. Thus, the synergy between the structure and geometry hierarchies is essential for simultaneously learning the embedding spaces.

### 3.2 Conditional Part Geometry VAE

In the geometry hierarchy of a 3D shape, each part geometry \(G_i\) is represented as a pair of ACAP feature \(X_i \in \mathbb{R}^{V \times 9}\) and the part center \(c_i \in \mathbb{R}^3\). We propose a part geometry conditional variational autoencoder (VAE) with a conditional part geometry encoder \(\text{Enc}_{PG}\) that maps the part geometry \(G_i = (\hat{X}_i, c_i)\) into a 128-dimensional latent feature and a conditional part geometry decoder \(\text{Dec}_{PG}\) which reconstructs \(\hat{G}_i\) from the latent code. Both the encoder and decoder are conditioned on the part semantics and its current structural context, in order to generate part geometry that is synergistic to the current structure tree nodes. We use the mesh graph convolutional operator to aggregate the local features around the vertex, which is also suitable for shape analysis [Monti et al. 2017; Wang et al. 2020].

Figure 4 (b) illustrates the proposed part geometry conditional VAE architecture. The encoder network \(\text{Enc}_{PG}\) performs two sequential mesh graph convolutional operations over the \(X_i \in \mathbb{R}^{V \times 9}\) feature map within local one-ring neighborhood around each vertex, extracts a global part geometry feature via a single fully-connected layer, which is then concatenated with the part center vector \(c_i\), and finally predicts a 128-dim geometry feature \(\hat{f}^G_i\) for part \(P_i\).

The decoder network \(\text{Dec}_{PG}\) decodes the part ACAP feature \(\hat{X}_i\) and the part center \(c_i\) through fully-connected and mesh-based convolutional layers. Then, the decoded ACAP feature \(\hat{X}_i\) is applied on every vertex of the closed box mesh \(C_{\text{box}}\) to reconstruct the part mesh \(\hat{G}_i\) and the reconstructed center \(\hat{c}_i\) moves the part mesh to the correct position in the shape space. Both the encoder and decoder are conditioned on a structure code condition \(f^S_i\) summarizing certain part semantics and its structural context.

Different from SDM-NET where they train separate PartVAEs for different part semantics, we propose to use a single shared PartVAE to encode and decode shape part geometry that is conditional on the part structure information \(f^S_i\). The reason is three-fold: firstly,
3.3.1 Structure VAE

Given a structure hierarchy \((l_1, l_2, \cdots, l_N), H, R\) describing a symbolic tree with part semantics, hierarchy and relationships, the structure VAE is trained to learn a structure latent space. For the encoding process, a part structure encoder \(Enc_{PS}\) first summarizes the leaf-node part semantics and then a recursive graph structure encoder \(Enc_{RGS}\) propagates features from the leaf nodes to the root in a bottom-up manner according to the part hierarchy \(H\) and relationships \(R\). Inversely, the decoding process contains a recursive graph structure decoder \(Dec_{PS}\) that hierarchically predicts the structure features from the root to the leaf nodes in a top-down fashion and a part structure decoder \(Dec_{PS}\) that decodes part semantic labels for the leaf nodes.

The structure VAE uses a similar recursive neural network architecture to StructureNet [Mo et al. 2019a], but we are encoding and decoding symbolic structure hierarchies with no concrete part geometry. It is thus difficult to train the decoding procedure given no part geometry since we are not able to perform node matching between a set of decoded children and the set of ground-truth parts. To address this challenge, we borrow the corresponding part geometry decoded from the geometry VAE to perform the node matching for the training, where a communication channel between the structure and geometry VAEs is established. Below, we discuss more details on the four network components for the structure VAE.

**Encoders.** To encode a symbolic structure hierarchy represented as \((l_1, l_2, \cdots, l_N), H, R\), we need to introduce an additional part instance identifier for each part \(d_i\), where \(d_i = 0, 1, 2, \cdots\), similar to PT2PC [Mo et al. 2020]. Part instance identifiers help differentiate the part instances with the same part semantics for a parent node. For example, if a chair base contains four chair legs, we mark them with part instance identifiers 0, 1, 2, 3. The part instance identifiers are only necessary for the encoding stage and will be ignored in the decoding procedure.

For each leaf node part \(P_i\), the part structure encoder \(Enc_{PS}\) encodes the part semantics \(l_i\) and its part instance identifier \(d_i\) into a part structure latent code \(f^S_{i}\).

\[
    f^S_{i} = Enc_{PS} ([l_i; d_i])
\]

where \(Enc_{PS}\) is simply a fully-connected layer, \([; ;]\) denotes the vector concatenation, and we represent both \(d_i\) and \(l_i\) as one-hot vectors.

For the non-leaf part \(P_i\), the recursive graph structure encoder \(Enc_{RGS}\) gathers all children node features, performs graph message-passing along the part relationships defined in \(R\) among the children nodes, and finally computes \(f^S_{i}\) by aggregating the children nodes’ features, as illustrated in the red branch of Figure 6 (left). Namely,

\[
    f^S_{i} = Enc_{RGS} \left( \left\{ f^S_{j} \right\}_{(P_j, P_i) \in H}, l_i, d_i \right)
\]

where \((P_j, P_i) \in H\) denotes that part \(P_j\) is a child of \(P_i\). The module \(Enc_{RGS}\) is composed of two iterations of graph message-passing similar to StructureNet [Mo et al. 2019a], a max-pooling operation over the obtained node features and a fully-connected layer producing the part structure feature \(f^S_{i}\) given the pooled feature and the part identifiers \([l_i; d_i]\) for the part. Here, please note that the part instance identifiers are necessary, due to the max-pooling operation, to distinguish and count the different occurrences of part instances with the same part semantics, and such crucial information would otherwise be lost.

We repeatedly apply the part structure encoder \(Enc_{RGS}\) until reaching the root node \(f^S_{\text{root}}\). The final root node structure feature \(f^S_{\text{root}}\) is then mapped to the final structure embedding space through a fully-connected layer. We use a KL divergence loss to encourage the learned structure latent space to be close to a univariate Gaussian distribution.

**Decoders.** The decoding process of a structure VAE takes a structure latent code as input and recursively decodes a symbolic structure hierarchy \((l_1, l_2, \cdots, l_N), H, R\) as the output. The part instance identifiers are not involved in the decoding procedure.

The recursive graph structure decoder \(Dec_{RGS}\) consumes the parent structure feature \(f^S_{i}\) and infers a set of children node structure features \(\{f^S_{1, i}, f^S_{2, i}, \cdots, f^S_{N, i}\}\), where we assume there is a maximum of 10 children parts per parent node. Following StructureNet [Mo et al. 2019a], we predict a semantic label and an existence probability for each part, by another fully-connected layer followed by classification output layers. Besides the node prediction, by connecting all pairs of parts, we also predict a set of symmetric

\[
    f^S_{i} = Dec_{RGS} \left( \left\{ f^S_{j} \right\}_{(P_j, P_i) \in H}, l_i, d_i \right)
\]
or adjacent edges $\tilde{R}_i$ among the existing nodes. Along the predicted edges, node features \( \{ f^S_{j_k} \} \) are updated via two graph message-passing operations and finally we decode a set of structure part nodes \( \{ f^S_{j_k}, f^G_{j_k}, \cdots, f^S_{j_k} \} \), where $K_i$ denotes the number of existing nodes for part $P_i$. We refer the readers to StructureNet [Mo et al. 2019a] for more details. The red branch in Figure 6 (right) illustrates this process. More formally,

$$
\{ f^S_{j_k}, f^G_{j_k}, \cdots, f^S_{j_k} \} = \text{DecRes} \left( f^S \right)
$$

(4)

We repeat the recursive structure decoding procedure until reaching the leaf nodes. For a leaf node part $P_i$, the part structure decoder $\text{DecPS}$ simply decodes the part semantic label via a fully-connected layer followed by outputting a likelihood score for each part semantic label. Finally, we get

$$
\hat{P}_i = \text{DecPS} \left( f^S_i \right)
$$

(5)

To train the hierarchical decoding process, StructureNet [Mo et al. 2019a] predicts part geometry for the intermediate nodes and establishes a correspondence between the predicted set of parts and the ground-truth set of parts. However, it is difficult to directly adapt this training procedure to decode the symbolic structure hierarchy by matching the part semantic labels. We resolve this challenge by building a communication channel between the structure hierarchy and the geometry one and borrowing the corresponding part geometry decoded in the geometry VAE for the matching procedure.

In our implementation, we utilize the conditional part geometry decoder $\text{DecPG}$ introduced in Sec. 3.2 and predict an oriented bounding box (OBB) geometry $\tilde{B}_j$ for each part $P_j$ where $j = j_1, j_2, \cdots, j_{K_i}$. We choose to use the OBB geometry for the matching process instead of the mesh geometry $\tilde{G}_i$ since we observe a decreased accuracy for registering the box mesh $G_{box}$ to an intermediate part geometry, which is usually more complex than leaf-node parts.

To train the part existence scores, part edge predictions and the part semantic labels, we follow StructureNet [Mo et al. 2019a] and refer the readers to the paper for more details.

### 3.3.2 Geometry VAE

Given a geometry hierarchy \( \{ G_1, G_2, \cdots, G_N \} \) encoding the part geometry of shape parts, the geometry VAE learns to map the shape geometry to a geometry latent space, disentangled from the structure latent space. The geometry latent space is also modeled to be a unit multivariate Gaussian distribution.

The geometry VAE shares a similar network architecture to the structure VAE. The encoding process starts from extracting part geometry features for all leaf-node parts via a part geometry encoder $\text{EncPG}$ and then recursively propagates the geometry features along the hierarchy to the root node, summarizing the geometry information for the entire shape through a recursive graph part geometry encoder $\text{EncReg}$. For the decoding process, we first use a recursive graph geometry decoder $\text{DecReg}$ that hierarchically decodes the geometry features from the root to the leaf-node parts in an inverse recursive manner. Then, we leverage a part geometry decoder $\text{DecPS}$ to reconstruct the part geometry for leaf-node parts.

There are two communication channels that allow the synergistic structure hierarchy to guide the geometry VAE encoding and decoding procedures. Firstly, the part geometry encoder $\text{EncPG}$ and decoder $\text{DecPG}$ are conditioned on the structure context produced by the structure VAE, which allows for generating different kinds of part geometry according to different part semantics and shape structures. Secondly, the graph message-passing procedures in the recursive graph geometry encoder $\text{EncReg}$ and decoder $\text{DecReg}$ borrow the part hierarchy and relationships defined in the structure hierarchy.

As follows, we describe the encoding and decoding stages for learning geometry VAE in more detail.

**Encoders.** We start from encoding each leaf node part geometry $G_1 = (X_i, c_i)$ into a latent part geometry feature space. We use the conditional part geometry encoder $\text{EncPG}$ introduced in Sec. 3.2 that maps the part $ACAP$ feature $X_i$ and the part center $c_i$ to a 128-dimensional feature $f^G_1$, namely,

$$
f^G_i = \text{EncPG} \left( X_i; c_i \right)
$$

(6)

The network is conditioned on the structure code $f^S_i$ generated in the structure VAE, in order to gain some structural context on what the semantics for the current part is and what role the part plays in generating the final shape.

For each sub-hierarchy of the part geometry, we recursively produce the intermediate part geometry node feature $f^G_i$ by aggregating its children geometry node features \( \{ f^G_j \} \) through the recursive graph geometry encoder $\text{EncReg}$, as illustrated in the blue branch of Figure 6 (left). Similar to the design of $\text{EncReg}$ for structure VAE, it performs two iterations of graph message-passing operations among the children geometry node features based on the part relationships between sibling part nodes, and conduct a simple max-pooling operation to compute $f^G_i$, where we have

$$
f^G_i = \text{EncReg} \left( \{ f^S \} \right)
$$

(7)

Different from $\text{EncReg}$ as shown in Eq. 3, we do not encode the part geometry for the non-leaf node since the geometry is more complex and the registration to a box mesh is less accurate. The increased geometric complexity also makes it harder to effectively embed them in a low-dimensional latent space. For the message-passing operations, we borrow the part relationships defined in the structure hierarchy. This is achieved by maintaining a bijective mapping among the tree nodes in the structure and geometry hierarchies. We repeatedly apply the recursive graph geometry encoder $\text{EncReg}$ until reaching the root node $P_{root}$. The final root node geometry feature $f^G_{root}$ is then mapped to the final geometry embedding space through a fully-connected layer.

**Decoders.** The decoding process of a geometry VAE takes a geometry latent code as input and recursively decodes a geometry hierarchy \( \{ \tilde{G}_1, \tilde{G}_2, \cdots, \tilde{G}_N \} \) for a shape.

As illustrated in the blue branch in Figure 6 (right), the recursive graph geometry decoder $\text{DecReg}$ takes the parent geometry feature $\tilde{G}_i$ as input and decodes a set of children node geometry features \( \{ f^G_1, f^G_2, \cdots, f^G_{10} \} \). Then, based on the structural predictions on
part existence scores, part semantic labels and part edge information from the synergistic structure VAE, we conduct two iterations of graph message-passing over the children node geometry features along the predicted pairwise part relationships \( \hat{R}_i \). The decoder \( Dec_{RG} \) then produces a final set of children nodes with the predicted part geometry features.

\[
\{ f_1^G, f_2^G, \cdots, f_{K_i}^G \} = Dec_{RG} \left( \hat{f}_i^S, \hat{R}_i \right)
\]  

where \( Dec_{RG} \) is conditioned on the decoded part relationships \( \hat{R}_i \) in the structure VAE and \( K_i \) denotes the number of existing part nodes predicted by the recursive graph structure decoder \( Dec_{RG} \).

We repeat the recursive graph geometry decoding procedure until reaching the leaf nodes. For a leaf node part \( \hat{P}_i \), we use the conditional part geometry decoder \( Dec_{PG} \) introduced in Sec. 3.2 that reconstructs \( \hat{G}_i = (\hat{X}_i, \hat{e}_i) \) from an input part geometry feature \( \hat{f}_i^G \). Formally, we have

\[
\hat{G}_i = Dec_{PG} \left( \hat{f}_i^G, \hat{S}_i \right)
\]  

Notice that the network \( Dec_{PG} \) is conditioned on the part structure code \( \hat{f}_i^G \) predicted in the coupled structure VAE decoding procedure.

The geometry VAE is trained jointly with the structure VAE and the conditional part geometry VAE. To supervise the reconstruction of the leaf-node part geometry in the decoding process, we simply adapt the loss terms defined in Eq. 1 from Sec. 3.2. We also add a KL divergence loss term to train the geometric latent space to get closer to the unit multivariate Gaussian distribution.

3.4 Cycled Disentanglement Mechanism (CycD) for Disentangled Geometry & Structure VAEs

Furthermore, in order to ensure that the geometry and structure are effectively disentangled and improve the performance of disentanglement, we propose a new Cycled Disentanglement mechanism to further disentangle the geometry and structure of shapes. Figure 7 illustrates our pipeline.

For any two input shapes (A, B), our new framework extracts their geometry and structure features \( \{ G_A, S_A \}, \{ G_B, S_B \} \) by DSG-Net. Then, we can synthesize a new shape C by combining features from structure and geometry of the two shapes, such as \( S_A \) and \( G_B \). So, the newly synthesized shape C has the geometric features from shape A and the structural features from shape B. The disentangled performance of our framework can be ensured and improved via paired self-supervised losses. During training, the loss terms \( L_{struct}, L_{geo} \) encourage the geometry code \( G_C \) and structure code \( S_C \) of Shape C to be the same as \( G_B \) and \( S_A \) under the MSE metric, i.e.,

\[
L_{struct} = ||S_C - S_A||^2_2, \quad L_{geo} = ||G_C - G_B||^2_2
\]  

Furthermore, the geometry code \( G_C \) and structure code \( S_C \) of shape C can be used to reconstruct the original shape A and B as \( A', B' \). Hence, in addition to the constraints on the shape geometry and structure codes, we have added the supervisions on the shapes \( A', B' \), which aims to make the disentanglement more successful in a shape-aware manner.

3.5 Post-Processing for Detached Parts

Though explicitly considering part relationships as soft constraints in the modeling already helps generate shapes whose parts are well structured and connected, we still observe some occasional failure cases of floating parts or part disconnections. We thus propose to use a post-processing module that directly optimizes the position of each part and further resolves the issue of detached parts.

Concretely, we fix the pre-trained parameters of the encoder and decoder, except for the center prediction MLP in the decoder network, so that we only optimize the center position of each part. Then, taking as input an object with disconnected parts, we optimize for the final location for each part by adjusting the part center positions. We employ two loss terms for the optimization: a structure loss, which enforces the adjacent parts to get closer to each other and the symmetric parts to satisfy the symmetric constraints, and an identity loss, which encourages the optimized shape to be similar to the input shape geometry, avoiding the degraded case that all parts are clustered together. During the optimization process, we balance the two loss terms with weights 1 and 100.

4 EXPERIMENTS

Learning disentangled latent spaces for shape structure and geometry allows us to generate high-quality 3D shape meshes with complex structure and detailed geometry in a controllable manner. DSG-Net not only demonstrates the state-of-the-art performance for structured shape generative modeling, but also enables generating shape meshes with controllable structure and geometry factors.

In this section, we present extensive experiments on the tasks of shape reconstruction, generation and interpolation, where we show the superior performance of our proposed method on the PartNet dataset [Mo et al. 2019b], compared to several strong baselines, including StructureNet [Mo et al. 2019a], SDM-Net [Gao et al. 2019b], IM-Net [Chen and Zhang 2019] and BSP-Net [Chen et al. 2020]. We also propose and formulate the tasks of disentangled shape reconstruction, generation and interpolation, where we manipulate one factor of shape structure and geometry while keeping the
4.2 Shape Reconstruction

In this section, we present the shape reconstruction performance of our DSG-Net and provide quantitative and qualitative comparisons to the state-of-the-art 3D shape generative models. Figure 8 shows the shape reconstruction results for our DSG-Net on the four shape categories in PartNet. We present more reconstructed results in the supplementary material. We observe that our method successfully captures both the complex shape structure and the fine-grained geometry details. Next, we present qualitative results on PartNet and provide quantitative evaluations on the synthetic dataset where we are provided with the ground-truth re-synthesized outputs. Please refer to the supplementary material for more results on shape reconstruction.

Baselines. We compare DSG-Net to four state-of-the-art methods for learning 3D shape representations – IM-Net [Chen and Zhang 2019], BSP-Net [Chen et al. 2020], StructureNet [Mo et al. 2019a], and SDM-Net [Gao et al. 2019b], as well as an ablated version (SN + Mesh) of our method. IM-Net learns an implicit function representation for encoding 3D shapes, while BSP-Net puts attention on designing a compact mesh representation for 3D shapes. They both represent a shape as a whole, without explicit modeling of shape parts and structures. StructureNet and SDM-Net are more relevant baselines to our method since they both explicitly represent shapes as part hierarchies. StructureNet uses point cloud representation for the part geometry, which we empirically find less effective on generating fine-grained shape geometry details. SDM-Net represents shapes with shallower part hierarchies, which prevents it from generating shapes with complicated structures. The SN + Mesh is a naive combination of StructureNet [Mo et al. 2019a] backbone and ACAP mesh representation [Gao et al. 2019a,b], to validate that our proposed disentangled structure and geometry representation and the cycled disentanglement indeed help improve the performance for learning 3D shape generative models. For the performance of SN+Mesh, please refer to supplementary material. All the methods are trained on the same data for the four object categories. We also compare to SAGNet [Wu et al. 2019] in the supplementary material.

Metrics. We adopt two kinds of metrics for quantitative comparisons to alternative methods: the geometry metrics and the structure metrics. For the geometry metrics, we compare the reconstructed shapes against the input shapes without explicitly considering the shape parts and structures. We follow the commonly used metrics in the literature: Chamfer Distance (CD) [Barrow et al. 1977] and Earth Mover’s Distance (EMD) [Rubner et al. 2000]. The CD and EMD are two permutation-invariant metrics for evaluating the difference of two unordered point sets, which have been used in the literature [Fan et al. 2017]. The CD measures the nearest distance for each point in one set to another point set. The EMD solves an optimization for bijective mapping between two point sets. For the structure metric, we use the HierInsSeg score proposed in PT2PC [Mo et al. 2020]. To compute the HierInsSeg score, Mo et al. [2020] first parse the reconstructed shape point cloud into the PartNet part hierarchy leveraging a pre-trained shape hierarchical instance segmentation network, and then compute the normalized tree-editing distance between the reconstructed and ground-truth part hierarchies. We
Table 1. Shape reconstruction quantitative evaluations. We use two geometry metrics (CD and EMD) and one structure metric (HierInsSeg). DSG-Net achieves the best geometry and structure performance compared to all baseline methods. Thanks to the cycled disentanglement, DSG-Net achieves the best HierInsSeg scores, followed by StructureNet, and DSG-Net outperforms them in terms of the geometric metrics by a large margin.

| DataSet | Method  | Geometry Metrics | Structure Metrics |
|---------|---------|------------------|-------------------|
|         |         | CD x 10^{-3}    | EMD x 10^{-2} | HierInsSeg (HIS) |
| Chair   | StructureNet | 9.34          | 6.45             | 0.51              |
|         | IM-Net  | 3.39          | 1.45             | 0.60              |
|         | BSP-Net | 8.27          | 2.07             | 0.78              |
|         | SDM-Net | 8.64          | 3.15             | 0.94              |
|         | Ours    | 1.98          | 0.73             | 0.39              |
|         | GT      |                |                  | 0.32              |
| Table   | StructureNet | 14.63         | 5.68             | 0.97              |
|         | IM-Net  | 5.04          | 2.08             | 1.13              |
|         | BSP-Net | 10.62         | 4.11             | 1.20              |
|         | SDM-Net | 9.73          | 4.63             | 1.38              |
|         | Ours    | 3.42          | 0.75             | 0.85              |
|         | GT      |                |                  | 0.65              |
| Cabinet | StructureNet | 16.34         | 5.74             | 0.57              |
|         | IM-Net  | 4.73          | 3.82             | 0.72              |
|         | BSP-Net | 6.67          | 4.65             | 0.84              |
|         | SDM-Net | 18.02         | 7.9              | 1.38              |
|         | Ours    | 2.96          | 0.97             | 0.45              |
|         | GT      |                |                  | 0.35              |
| Lamp    | StructureNet | 17.31         | 7.12             | 0.71              |
|         | IM-Net  | 13.20         | 5.11             | 0.73              |
|         | BSP-Net | 17.17         | 7.56             | 0.98              |
|         | SDM-Net | 51.21         | 8.72             | 0.76              |
|         | Ours    | 7.15          | 1.63             | 0.61              |
|         | GT      |                |                  | 0.54              |

refer the readers to Fan et al. [2017] and Mo et al. [2020] for more details on the definitions of the metrics.

Results. Table 1 presents the quantitative comparisons between DSG-Net and the alternative methods. Our method performs the best on the geometry metrics, indicating that DSG-Net captures better shape geometry. Our DSG-Net also outperforms IM-Net, BSP-Net, and SDM-Net on the structure metric by significantly large margins, while achieving slightly better performance than StructureNet thanks to our cycled disentanglement. We observe that StructureNet achieves a comparable HierInsSeg score since it tends to generate parts that are more disconnected as shown in Figure 9 (e), which is beneficial to make the part structure clearer but is detrimental to the overall shape geometric appearance. Figure 9 presents the qualitative comparison to the baseline methods. It is clear to observe that IM-Net, BSP-Net and SDM-Net fail to generate complicated shape structures, while our method can successfully capture these complex shape structures. Compared with StructureNet, we reconstruct the shape geometry more accurately.

Fig. 9. Shape reconstruction comparison with the baseline methods. DSG-Net can reconstruct high-quality shape meshes with complex shape structures and detailed part geometry. IM-Net, BSP-Net, and SDM-Net fail to reconstruct the complicated shape structures (e.g. chair back bars and table leg stretchers), while StructureNet (SN) generates point cloud shapes with less part geometry details and inaccurate part geometry. For instance, StructureNet fails to reconstruct the slanted bars for the chair in the first row and loses accuracy for the aspect ratio of the table top surface in the last row.

Fig. 10. Shape generation results. We sample random Gaussian noises in both latent spaces of shape structure and geometry and use DSG-Net to generate realistic shapes with complex structures and detailed geometry. We show six generation results for each of the four object categories in PartNet.

4.3 Shape Generation

The main goal of DSG-Net is to generate high-quality shapes with complex structures and fine-grained geometry. Given a noise vector sampled from a unit Gaussian distribution, a 3D shape generative model maps it to a realistic 3D shape. We evaluate the shape generation performance of DSG-Net and compare to several state-of-the-art baseline methods. Quantitative evaluations and user-study results further validate our superior performance over baselines. Equipped with two disentangled latent spaces for shape structure and geometry, DSG-Net also enables a novel task of generating
Fig. 11. Examples demonstrating generation of novel shapes. We present top-five retrieved shapes in the training sets, with CD as the retrieval metric, to our generated results shown in the left most column. Our generated shapes are different from the shapes in training sets, showing that DSG-Net does notSimply overfit the training data.

Fig. 12. Qualitative comparisons on shape generation. We compare our generated shapes to the baseline methods and show that our method learns to generate shapes with complex structures and fine-grained geometry. For each example, we show shapes generated by alternative methods which are nearest to our results under the CD metric. We can observe that, compared to our DSG-Net, StructureNet fails to generate high-quality shape geometry, SDM-Net cannot generate shapes with complex part structures, and the SN+Mesh is not able to generate shapes with compatible structure relationships.

shapes with a given structure or geometry pattern. Please refer to the supplementary material for more results on shape generation.

Metrics. The shape generation task aims to generate diverse and realistic shapes with complex structure and geometry. Following StructureNet [Mo et al. 2019a], we measure shape generation performance by the coverage and quality scores. The coverage score computes the average distance from a real shape to the closest generated shape, while the quality score calculates the average distance from a generated shape to the closest real shape. The coverage score reflects the diversity of the generated results is large enough to cover all real samples, and the quality score measures if the generated results contain bad examples that are far from the real data distribution. To compare with the baseline methods, we generate 1000 shapes and compute the coverage and quality scores regarding the geometry metric (CD) and the structure metric (HierInsSeg).

Results. Figure 10 shows our shape generation results, where we observe the complicated structure and fine mesh geometry are generated at the same time. In Figure 11, we further validate the novelty of the generated shapes by comparing them to the top five retrieved training shapes. In Figure 12, we compare our method to SDM-Net, StructureNet and our ablated version (SN+Mesh), where we clearly see that DSG-Net generates better shape geometry than StructureNet and produces shapes with more complicated structures than SDM-Net. We further show quantitative comparisons in Table 2, where we see that DSG-Net obtains clear improvements over the baselines. In addition, we conduct a user study to further evaluate how realistic the generated shapes are for humans. We render the shapes into images and ask the users to rank the three algorithms according to three different criteria (geometry, structure and overall). In Table 3, we observe that our generated shapes perform the best to human users in all the three criteria. See supplementary for more results.

Disentangled Shape Generation. DSG-Net learns two disentangled latent spaces for modeling shape structure and geometry, which enables a novel task of generating shapes with a given shape structure or geometry pattern. We demonstrate that given an input shape, DSG-Net can extract the structure code from the shape and pair it with a random geometry code, which allows us to explore shape geometry variations satisfying a certain shape structure. It also works well to explore structure variations while keeping the geometry code unchanged. We show four controllable generation results on the four categories in Figure 13. In the experiments, given an input shape, the geometry code and structure code are extracted.
DSG-Net: Learning Disentangled Structure and Geometry for 3D Shape Generation

Fig. 13. Qualitative results for disentangled shape generation on four categories. Given an input shape (a), we extract the geometry code and structure code. We fix one of them, and randomly sample code in the other latent space to generate new shapes (b). For the first row of (b), we keep the geometry code unchanged and randomly explore the structure latent space. And, for the second row, we keep the structure code unchanged and randomly sample over the geometry latent space. For the table example, we can see that the generated shapes share a similar size and length of table surface and legs in the first row but the other parts and shape structure are changing, while in the second row, the geometry of all parts is changing but the structure remains unchanged. We can also observe similar results for the cabinet and lamp examples.

by running it through the encoding procedures. And then, we can keep one of them unchanged and randomly sample in the other latent space. We see that when we preserve the geometry code, the chair/table legs usually maintain similar width and length to the input shapes. And, when we keep the structure code unchanged, we are generating shapes with geometric variations but satisfying the same symbolic structure hierarchy. Please refer to supplementary material for more disentangled generation results.

Fig. 14. Shape interpolation results. We linearly interpolate between input shape pairs (the left most and the right most shapes) jointly in the structure and geometry latent spaces. We see both continuous geometry variations and discrete structure changes. For the chair examples, in the first row, we see that the armrests become smaller and then disappear while the backrest changes from a square to round fashion in a more natural manner, while in the second row, the backrest gradually becomes square, while the supporter disappears form the first chair to the second chair. We observe similar behaviors for the table, lamp, and cabinet results.

4.4 Shape Interpolation

We evaluate our DSG-Net on shape interpolation and demonstrate that our network learns smooth latent spaces. Next, we propose a novel task of disentangled shape reconstruction that takes two shapes as inputs to re-synthesize a novel shape with ingredients of the structure of one shape and the geometry of the other shape. Moreover, with the help of the novel cycled disentanglement and our learned disentangled latent spaces for shape structure and geometry, we can also achieve controllable interpolation between two shapes, varying shape structure while keeping geometry unchanged and vice versa. Please refer to the supplementary material for more results on shape interpolation.

Results. Figure 14 shows some interpolated results that interpolate jointly in the structure and geometry latent spaces. We see both continuous geometry variations and discrete structure changes in the interpolation, which validates that DSG-Net learns a smooth manifold for shape generation.

Disentangled Shape Interpolation. Our disentangled representations for shape structure and geometry also allow us to achieve controllable interpolation between two shapes that one may keep the structure or geometry factor unchanged while interpolating the other factor. Figure 16 shows some disentangled shape interpolation results. From the results, we find that DSG-Net can achieve disentangled shape interpolation and every interpolated step produces a very realistic and reasonable shape.

Disentangled Shape Reconstruction. Our methods learn two disentangled latent manifolds (structure and geometry) for shape
Fig. 15. Disentangled shape reconstruction and interpolation results on PartNet Cabinet and Table. Here, the top left and bottom right shapes (highlighted with orange boxes) are the input shapes. The remaining shapes are generated automatically with our DSG-Net, where in each row, the structure of the shapes is interpolated while keeping the geometry factor unchanged, whereas in each column, the geometry is interpolated while retaining the structure. The vertical axis and horizontal axis represent the variation of structure and geometry respectively.

Table 4. Quantitative evaluations of disentangled shape reconstruction on the synthetic data (for details please refer to Sec. 4.1). We compare two ablated versions of our method, namely ours (w/o edge) and ours (w/o CycD) since there is no applicable external baseline method for this novel task. We observe that allowing edge communications between the structure and geometry hierarchies and novel cycled disentanglement are essential.

| Method         | Geometry Metrics | Structure Metrics |
|----------------|------------------|-------------------|
|                | $CD_{\times 10^{-3}}$ | $EMD_{\times 10^{-2}}$ | HierInsSeg(HIS) |
| Ours (w/o edge) | 1.50             | 1.39              | 1.92          |
| Ours (w/o CycD) | 1.29             | 0.61              | 1.87          |
| Ours           | **1.02**         | **0.58**          | **1.43**      |
| GT             |                  |                   | **1.19**      |

4.5 Ablation Studies

We perform four sets of ablation studies to demonstrate the necessity and effectiveness of the key components and training strategies for representations, which opens up new possibilities for controllable shape editing and re-synthesis tasks. Given two input shapes, one can push the two shapes through our structure and geometry VAE encoders and obtains the structure and geometry features for both shapes. Then, by re-combining the structure code of one shape and the geometry code of the other shape, DSG-Net is able to re-synthesize a novel shape that follows the structure of the first shape and the geometry of the second shape.

Figure 15 shows a set of qualitative results on the PartNet dataset (Cabinet and Table). The shapes in each row share the same geometry code while the shapes in every column have the same structural feature. Here, the top left and bottom right shapes are the inputs. The remaining shapes are generated with our DSG-Net, where in each row, the structure of the shapes is interpolated while keeping the geometry factor unchanged, whereas in each column, the geometry is interpolated while retaining the structure. The figure demonstrates that our method is able to re-synthesize novel shapes with pairs of geometry and structure configurations.

We further quantitatively benchmark the performance of DSG-Net for disentangled shape reconstruction on the synthetic dataset, where we have access to the ground-truth reconstruction results given a pair of structure and geometry configurations. Table 4 shows the quantitative results of the synthetic data. Since there is no applicable baseline method for this novel task, we compare two ablated versions of our method: 1. ours (w/o edge), where we ignore the part relationships from the part hierarchies and remove the graph message-passing procedures, which further reduces the communication between the structure and geometry; 2. ours (w/o CycD), where we remove the novel cycled disentanglement and losses ($L_{struct}$ and $L_{geo}$), which further reduces the capability of the disentanglement of geometry and structure. We see that removing edge communications or cycled disentanglement provides us with worse performance, which proves the importance of maintaining the synergy between the disentangled structure and geometry hierarchies. For more results on disentangled interpolation on Lamp and Chair categories and some qualitative results on the synthetic data, please refer to supplementary material.
We observe similar performances for the two strategies. We take the performance for learning 3D shape generative models. We validate the design choice of learning a unified conditional part geometry VAE, instead of training separate VAEs for each part. First, the evaluation on the post-processing procedure shows its effectiveness for resolving the issue of detached parts. Then, we compare cascaded training and end-to-end training for the part geometry VAE and the disentangled backbone VAEs. We validate similar performances for the two strategies. We take the end-to-end training approach due to its simplicity. We also validate the design choice of learning a unified conditional part geometry VAE, instead of training separate VAEs for each part. Finally, we demonstrate that explicitly considering part relationships and conducting graph message-passing operations along the edges are important. Removing the edge components from our network gives significantly worse results. In terms of the cycled disentanglement, we also demonstrate its importance for our disentangled representation of shape structure and geometry on shape reconstruction, including the disentangled shape reconstruction on synthetic data. Moreover, the ablated version (StructureNet + Mesh) is a naive combination of StructureNet [Mo et al. 2019a] backbone and ACAP mesh representation [Gao et al. 2019a,b]. We also evaluate it to validate that our proposed disentangled structure and geometry representation and the cycled disentanglement indeed help improve the performance for learning 3D shape generative models. 

\section{5 LIMITATIONS AND FUTURE WORKS}

Our method depends on heavily annotated shape structure hierarchies and fine-grained part geometric annotations for a large-scale collection of 3D shapes as input to our networks. It is a non-trivial task to obtain such data from automatic algorithms. One may consider predicting such hierarchies by training hierarchical part instance segmentation networks (as shown in PartNet [Mo et al. 2019b] Sec. 5.3 and StructureNet [Mo et al. 2019a] shape abstraction experiments). But, these methods all require a large-scale training dataset of fine-grained part and structure annotations. For unsupervised methods, although recent works, e.g. Cuboid Abstraction [Sun et al. 2019], show promising results for learning such fine-grained shape parts and structures, it still remains a challenging topic in the research community.

Finally, although our method achieves the state-of-the-art performance in generating shapes with fine geometry and complex structure, our method still has some failure cases as shown in Figure 18: (1) some generated shapes may have detached parts and asymmetric parts (c), especially for rare shapes, as well as missing parts (a), detached parts (b), extra parts (d), duplicate parts (e), or incompatible size of parts (f), etc.; (2) our network restricts the maximum number of siblings to 10, which is the same as StructureNet. This is to improve the efficiency of memory consumption. But as a result, our network cannot handle shapes with more than 10 siblings for a part. Future works can work on addressing these issues.

\section{6 CONCLUSION}

In this paper, we have presented DSG-Net, a novel deep generative model that learns to represent and generate 3D shapes in disentangled latent spaces of geometry and structure, while considering their
synergy to ensure the plausibility of generated shapes. Through extensive evaluation, our method produces high-quality shapes with complex structures and fine geometric details, outperforming state-of-the-art methods. Our method also enables disentangled control of geometry and structure in shape generation, supporting novel applications such as interpolation of geometry (structure) while keeping structure (geometry) unchanged.

ACKNOWLEDGMENTS
This work was supported by the Beijing Municipal Natural Science Foundation for Distinguished Young Scholars (No. JQ21013), the National Natural Science Foundation of China (No. 6206136007 and No. 61872440), Royal Society Newton Advanced Fellowship (No. NAFVR2.192151) and the Youth Innovation Promotion Association CAS. Kaichun Mo and Leonidas J. Guibas were supported by a grant from the Samsung GRO program, NSF grant CHS-1528025, a Vannevar Bush Faculty fellowship, and gifts from the Autodesk and Snap corporations.

REFERENCES
Victoria Fernández Abrevaya, Adnane Boukhayma, Stefanie Wuhrer, and Edmond Boyer. 2019. A Decoupled 3D Facial Shape Model by Adversarial Training. In IEEE/CVF International Conference on Computer Vision. 9418–9427.

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. 2018. Learning Representations and Generative Models for 3D Point Clouds. In International conference on machine learning. PMLR. 40–49.

Enam Ahmed, Alexandre Saint, Abd Ell Rahman Shiabayek, Keerinya Cherenkova, Rig Das, Gile Gusev, Djamilia Aouada, and Bjorn Ottersten. 2018. Deep learning advances on different 3D data representations: A survey. arXiv preprint arXiv:1808.01462 (2018).

Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan D Jepson, and Sven J Dickinson. 2019. Geometric Disentanglement for Generative Latent Shape Models. In IEEE/CVF International Conference on Computer Vision. 8180–8189.

Elena Balashova, Vivek Singh, Jiangping Wang, Brian Teixeira, Terrence Chen, and Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. 2020. Complementary generative models for high-fidelity model completion. arXiv preprint arXiv:2008.00549 (2020).

Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J. Guibas, and Lin Gao. 2019b. Towards data-driven shape generation. In ACM Transactions on Graphics (TOG) 38, 6 (2019). 179:1–179:18.

Giorgio Patane, David Vlasic, and Leonidas J. Guibas. 2018. Learning to generate parts from point clouds. In Proceedings: Image Understanding Workshop 21–27.

Vladimir G. Kim, Wilmot Li, Niloy J. Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Evangelos Kalogerakis. 2012. Probabilistic reasoning for assembly-based 3D modeling. ACM Transactions on Graphics (TOG) 31, 6 (2012), 237:1–237:15.

Yonghe Xing, Russ Pavlicek, Daniel Ritchie, Jiajun Wu, Kai Xu, and Hao Zhang. 2020. Learning Generative Models of 3D Structures. Computer Graphics Forum (EuroGraphics STAR) (2020).

Jun Li, Chengjie Niu, and Kai Xu. 2020. Learning part generation and assembly for 3D part prediction. arXiv preprint arXiv:2007.05577 (2020).

Juxing Liu, Jiangwei Huang, and Lin Gao. 2018. Learning implicit fields for generative shape modeling. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 5939–5948.

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 2016. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In European Conference on Computer Vision. Springer, 628–644.

Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C. Russell, and Matthieu Aubry. 2019. Learning Elementary Structures for 3D Shape Generation and Matching. Advances in Neural Information Processing Systems 32 (2019).

Anastasia Dubrovin, Fei Xia, Panos Achlioptas, Mira Shahal, and Leonidas J. Guibas. 2019. Composite Shape Modeling via Latent Space Factorization. IEEE/CVF International Conference on Computer Vision (2019), 8139–8148.

Haoqiang Fan, Hao Su, and Leonidas J. Guibas. 2017. A point set generation network for 3D object reconstruction from a single image. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 605–613.

Vignesh Ganapathi Subramanian, Olga Diamanti, Soeren Pirk, Chengcheng Tang, Matthias Nießner, and Leonidas J. Guibas. 2018. Parsing geometry using structure-aware shape templates. In International Conference on 3D Vision. IEEE, 672–681.

Lin Gao, Yu-Kun Lai, Jie Yang, Ling-Xiao Zhang, Shihong Xia, and Lei Kobbelt. 2019a. Sparse data driven mesh deformation. IEEE Transactions on Visualization and Computer Graphics 27, 3 (2019), 2085–2100.

Lin Gao, Jie Yang, Yi-Ling Qiao, Yu-Kun Lai, Paul L. Rosin, Weiwei Xu, and Shihong Xia. 2018. Automatic Unpaired Shape Deformation Transfer. ACM Transactions on Graphics (TOG) 37, 6 (2018), 237:1–237:15.

Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao(Richard) Zhang. 2019b. SDM-NET: Deep Generative Network for Structured Deformable Mesh. ACM Transactions on Graphics (TOG) 38, 6 (2019), 243:1–243:15.

Kyle Genergy, Forrest Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas A. Funkhouser. 2019. Learning Shape Templates With Structured Implicit Functions. IEEE International Conference on Computer Vision (2019), 7153–7163.

Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C. Russell, and Matthieu Aubry. 2018. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 216–224.

Qixing Huang, Vladlen Koltun, and Leonidas J. Guibas. 2011. Joint Shape Segmentation with Linear Programming. ACM Transactions on Graphics (TOG) 30, 6 (2011), 1–12.

Anastasia Ioannidou, Elvisert Cheliotzis, Spiros Nikolopoulos, and Ioannis Kompatsiaris. 2017. Deep learning advances in computer vision with 3D data: A survey. ACM Computing Surveys (CSUR) 50, 2 (2017), 1–38.

Chenyu Jiang, Arnavesh Sud, Anmesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. 2019. Learning implicit fields for generative shape modeling. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 6001–6010.

Yiwei Jun, Qiuying Jiang, and Ming Cai. 2020. 3D reconstruction using deep learning: a survey. Communications in Information and Systems 20, 4 (2020), 389–413.

Eman Ahmed, Alexandre Saint, Abd Ell Rahman Shiabayek, Keerinya Cherenkova, Rig Das, Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. 2019b. Towards data-driven shape generation. In ACM Transactions on Graphics (TOG) 31, 4 (2012), 55:1–55:11.

Tero Karas, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 4403–4410.

Vladimir G. Kim, Wilmot Li, Niley J. Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. 2013. Learning Part-Based Templates from Large Collections of 3D Shapes. ACM Transactions on Graphics (TOG) 32, 4 (2013), 70:1–70:12.

Jake Levinson, Arnavesh Sud, and Anmesh Makadia. 2019. Latent feature disentanglement for 3D meshes. arXiv preprint arXiv:1906.03281 (2019).

Chun-Liang Li, Manzil Zaheer, Yi-Ling Qiao, Yu-Kun Lai, Paul L. Rosin, Weiwei Xu, and Shihong Xia. 2019. Learning part generation and assembly for 3D part prediction. arXiv preprint arXiv:1906.05795 (2019).

Jun Li, Chengjie Niu, and Kai Xu. 2020. Learning part generation and assembly for structure-aware shape synthesis. In AAAI Conference on Artificial Intelligence, Vol. 34. 11362–11369.

Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas J. Guibas. 2017. GRASS: Generative recursive autoencoders for shape structures. ACM Transactions on Graphics (TOG) 36, 4 (2017), 52:1–52:14.

Liqiang Lin, Pengbo Huang, Fuyou Xue, Kai Xu, Daniel Cohen-Or, and Hui Huang. 2021. Haussdorf point cloud reconstruction with geometric priors. Science China Information Sciences 64, 11 (2021), 1–13.

Daniel Maturana and Sebastian Scherer. 2015. VoxNet: A 3D convolutional neural network for real-time object recognition. In IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 922–928.

Lars Mescheder, Michael Oechsle, Michael Niemann, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3D reconstruction in function space. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 4460–4470.
