DSM-Net: Disentangled Structured Mesh Net for Controllable Generation of Fine Geometry

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

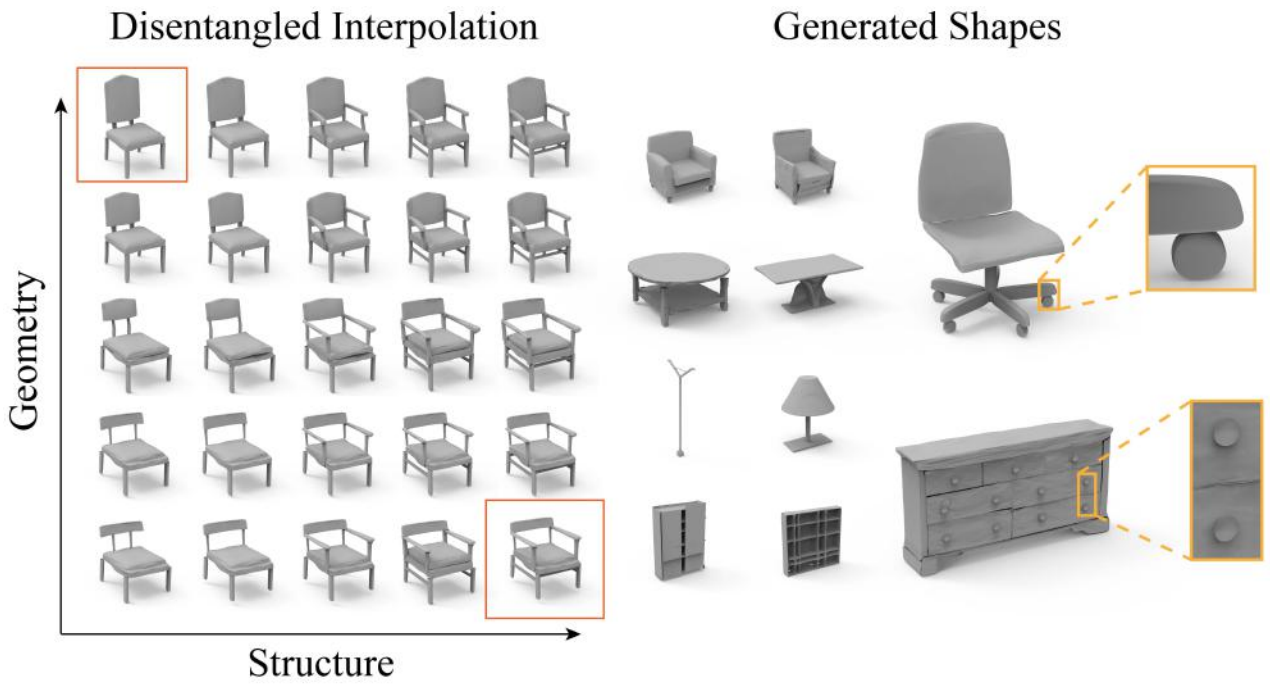


Fig. 1. Our deep generative network DSM-Net encodes 3D shapes with complex structure and fine geometry in a representation that leverages the synergy between geometry and structure, while disentangling these two aspects as much as possible. This enables novel modes of controllable generation for high-quality 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 DSM-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 DSM-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 geometric shapes with rich detail and complex structure, in a controllable manner. To tackle this, we introduce DSM-Net, a deep neural network that learns a disentangled structured 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 intuitive 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 generation applications, but also produces high-quality synthesized shapes, outperforming state-of-the-art methods.

CCS Concepts: • **Computing methodologies → Shape modeling**.

Additional Key Words and Phrases: 3D shape generation, disentangled representation, structure, geometry, hierarchies

∗ Authors contributed equally.
† Corresponding author.
Project webpage: http://geometrylearning.com/dsm-net/. This is the author's version of the work. It is posted here for your personal use. Not for redistribution.

1 INTRODUCTION

3D shapes are widely used in computer graphics and computer vision, with applications ranging from modeling, 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, to ensure high-level control of 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 representations have been widely studied in deep image generation, allowing different aspects, such as different facial attributes (expression, age, gender, etc.) to be manipulated separately, either in a supervised [Xiao et al. 2018a,b] or unsupervised [Chen et al. 2016] manner. For disentanglement of 3D shapes, existing works either focus on specific object categories such as human faces [Abrevaya et al. 2019] where explicit annotation is used for supervision, or are restricted to intrinsic/extrinsic decomposition, where shape geometry and poses are considered [Aumentado-Armstrong et al. 2019]. However, such methods are rather restrictive, and usually require the set of shapes to have point-to-point correspondence. None of these methods can handle the more general geometry and structure disentanglement we address in this work. Such disentangled and synergistic representations offer significant benefits, including controllable generation of new shapes, e.g., interpolating or transferring structure while keeping geometry unchanged, or manipulating geometry while retaining the structure.

Specifically, 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 of generating shapes with both complex structures and detailed geometry. They also have not addressed disentanglement of structure and geometry.

In this paper, we introduce Disentangled Structured Mesh Net (DSM-Net), a novel deep generative model which overcomes the above limitations. DSM-Net is based on the PartNet [Mo et al. 2019c] dataset with fine-grained, consistent part annotations aggregated into shape hierarchies. We follow the PT2PC [Mo et al. 2020] approach to group the PartNet data (e.g. chair, table, cabinet, etc.) according to their structure. However, our structure only includes the hierarchical graph of part semantics and relationships, excluding all geometric information, whereas the geometry hierarchy includes the detailed geometry and position of each part. Our network encodes 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 flow along the edges of hierarchical graphs and aggregate into two latent spaces, allowing these two key aspects to be encoded *separately* in a disentangled manner.

However, the latent codes from both spaces need to be correlated, to ensure the plausibility of the represented shape. To achieve this, we simultaneously train both structure and geometry VAEs. We further ensure that they communicate with each other: both hierarchies have bijective mappings bridging them at each level. During training, the structure communicates with the geometry and gives guidance on the generated part shapes. The geometry will follow the inter-part relationship edges with a message passing protocol. The geometry of parts also supplements the structure to provide reliable correspondences to the ground truth to facilitate training. The detailed geometry is encoded using a conditional VAE, where the structure context is used as the condition to further promote structure and geometry compatibility.

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, which is an intuitive process for shape modelers. New shapes can also be synthesized by mixing structure and geometry from different examples.

In summary, our DSM-Net makes the following key contributions:

- We propose a novel deep network that decomposes shape space into two disentangled latent spaces, encoding the geometry and structure of shapes. We incorporate communication between the geometry and structure, making them compatible on the generated shapes, while supporting novel synthesis applications that exploit independent control of structure and geometry.
- Our DSM-Net also allows high-quality shapes with complex structures and delicate geometric details to be effectively represented and synthesized, outperforming state-of-the-art methods.

Figure 1 demonstrates the capability of our DSM-Net to interpolate shapes with rich geometry and complex structure in the geometry and structure spaces, separately where each row shows interpolation of 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 on 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.

2 RELATED WORK

In recent years, researchers have been making great advances towards learning deep representations for 3D data and pushing the frontiers of 3D shape analysis, synthesis and modeling. A key research topic in 3D computer vision and graphics is how to represent, reconstruct and generate 3D shapes with complicated part structures and delicate geometric details.

In this section, we give a brief review on various kinds of 3D shape representations and provide a comprehensive discussion of recent advances on modeling 3D shape geometry and structure.

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. To name a few, recent works have developed deep learning frameworks for 3D voxel grids [Choy et al. 2019, 2016; Girdhar et al. 2016; Graham et al. 2018; Maturana and Scherer 2015; Riegler et al. 2017; Tatarchenko et al. 2017; Wu et al. 2017, 2016, 2015; Yan et al. 2016], multi-view 2D rendering of 3D data [Huang et al. 2017; Kalogerakis et al. 2017; Kanezaki et al. 2018; Lyu et al. 2020; Su et al. 2018, 2015], 3D point clouds [Achlioptas et al. 2018; Fan et al. 2017; Gadelha et al. 2018; Le et al. 2019; Li et al. 2018; Qi et al. 2017a,b; Shu et al. 2019; Valsesia et al. 2018; Yang et al. 2019, 2018; Zhao et al. 2019], 3D polygonal meshes [Chen et al. 2019a; Dai and Nießner 2019; Gkioxari et al. 2019; Groueix et al. 2018; Kanazawa et al. 2018; Nash et al. 2020; Sinha et al. 2017; Wang et al. 2018], and 3D implicit functions [Chabra et al. 2020; Chen and Zhang 2019; Chibane et al. 2020; Duan et al. 2020; Jiang et al. 2020; Mescheder et al. 2019; Park et al. 2019; Peng et al. 2020; Xu et al. 2019]. For more detailed discussion and comparison, we refer the readers to these survey papers [Ahmed et al. 2018; Bronstein et al. 2017; Ioannidou et al. 2017; Xiao et al. 2020].

More relevant to our work is the trend of part-based and structure-aware 3D shape representations. 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 [Chen et al. 2019b; Golovinskiy and Funkhouser 2009; Hu et al. 2012; Huang et al. 2011; Kalogerakis et al. 2017; Mo et al. 2019c; Tulsiani et al. 2017; Yi et al. 2017; Yu et al. 2019; Zhu et al. 2020; 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; Sung et al. 2017; Van Kaick et al. 2013; Wang et al. 2011; Wu et al. 2019b; Zhu et al. 2018a], and generating 3D shapes with part structures [Gao et al. 2019b; Kalogerakis et al. 2012; Li et al. 2019, 2017; Mo et al. 2020; Schor et al. 2019; Wu et al. 2019a]. We refer to these 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 generate 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,c], 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, beyond SDM-Net. 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 directly 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]. 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. 2019; Schor et al. 2019; Wu et al. 2019a] and sequential part instances [Sung et al. 2017; Wu et al. 2019b].

Recently, researchers have been investigating representing every shape as a hierarchy of parts, which extends part granularity to more fine-grained scales. The pioneering work to encode the 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. 2019c]. A concurrent work 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 adapts 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 is 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.

In the pioneering work SAG-Net [Wu et al. 2019a], both geometry and structure are encoded in the single latent code by an attention-based GRU network, and the geometry details are represented with a voxel-based representation and the graph structure is represented by a fully connected graph. Our DSM-NET is fundamentally different in that geometry and structure are encoded into separate latent codes in a hierarchical manner. The hierarchy of encoded geometry guided by the structure is the key in our work that achieves disentanglement while ensuring structure/geometry compatibility, which does not appear in SAG-NET. This novel design enables DSM-NET to disentangle geometry and structure while keeping the two informed of each other, and thus synthesize 3D mesh models with complex structure and compatible fine geometry, advancing the state-of-the-art in neural shape representations.

2.4 Shape Editing, Deformation and Transformation

Using deep learning to aid shape editing, deformation and transformation applications attracts much research attentions in recent years. To name a few, Yumer et al. [2016] learn semantic deformation over 3D voxel grids for deforming shapes subject to input user intents. 3DN [Wang et al. 2019b] learns to deform 3D meshes by predicting offsets for mesh vertices. NeuralCages [Wang et al. 2019a] learns to fit coarse cages outside shape meshes and conduct deformation over the cages. StructEdit [Mo et al. 2019b] learns a conditional variational autoencoder (cVAE) to generate plausible shape variations for a source shape and transfer editing operations among similar shapes. LOGAN [Yin et al. 2019] proposes a general framework to learn shape transforms from unpaired domains and demonstrates many interesting applications, such as 3D style transfer and generating shapes from skeletons. PT2PC [Mo et al. 2020] learns a conditional generative adversarial network that generates 3D shapes with geometric variations given a part-tree condition. Aumentado et al. [2019] proposes an unsupervised approach to learn disentangled representations for mammal and human point cloud shapes with factorization of the pose and intrinsic shapes.

In this paper, we learn a disentangled part hierarchies for shape geometry and structure, which enables many controllable shape editing and transformation applications, such as varying shape geometry (structure) while keeping the structure (geometry) unchanged, and shape re-synthesis combining the structure of one shape and the geometry feature of another shape.

2.5 Disentangled Analysis in Deep Learning

In the field of 2D image or 3D model processing, there are some pioneering research works on the *Disentangled Analysis*.

With the advancement of deep learning in the field of 2D images, many works aim to improve the generation quality and manipulate the generated images. Borrowing from the style transfer literature, the proposed architecture [Karras et al. 2019] enables intuitive, scale-specific control of the high-resolution image synthesis by automatic unsupervised separation of high level attributes. HoloGAN [Nguyen-Phuoc et al. 2019] improves the visual quality of generation and allows manipulations by the disentangle learning, which utilizes explicit 3D features to disentangle the shape and appearance in an end-to-end manner from unlabeled 2D images only. Also in the field of 3D shape processing, the generative modeling becomes a mainstream topic thanks to the deep learning and tremendous public 3D datasets, some of which contain rich textures for realism. 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] proposed a CFAN (conformal factor and normal) feature to achieve the geometric disentanglement (pose and identity of human shape) 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 for image and shape synthesis. For more photo-realistic image generation, it decomposes the process into three factors: shape, viewpoint, and texture.

Compared to the above works, our work displays a rather novel capability - *disentanglement of structure and geometry*. 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*. Our network can not only be used to generate shape with improved geometric details, but also allows to exploit independent control of structure and geometry with the disentangled latent space.

3 METHODOLOGY

As provided in the PartNet dataset [Mo et al. 2019c], every 3D shape is decomposed into semantically consistent part instances that are organized by an n-ary part hierarchy that covers parts at different granularities, ranging from coarse-grained parts (e.g. chair back, chair base) to fine-grained ones (e.g. chair back bars, chair legs). The part hierarchy also includes a rich set of part relationships shedding lights on the complicated shape structure, such as the vertical parent-child relations and the horizontal symmetry or adjacency relations. Such a part hierarchy provides a powerful representation

that describes complex structure and geometry details in a unified format.

In this paper, we propose a *disentangled* but *highly synergistic* hierarchical representation for shape geometry and structure (see Figure 2). We disentangle the unified PartNet [Mo et al. 2019c] part hierarchy into a *structure hierarchy*, which describes the symbolic part semantics and part relationships, and a *geometry hierarchy*, which contains the detailed part mesh geometry for the tree nodes. The structure and geometry hierarchies are disentangled to enable controllable shape editing in downstream applications, as we will show in Sec. 4, while still being highly coupled and synergistic in that the two hierarchies have a bijective part correspondence among the tree nodes and they are learned together so as to generate 3D shapes with compatible structure and geometry.

We use Recursive Neural Networks (RvNNs) to hierarchically encode and decode the structure and geometry part hierarchies. Different from StructureNet [Mo et al. 2019a], we propose to learn two separate but deeply coupled VAEs to encode the geometry and structure hierarchies into two latent spaces, producing a disentangled representation for shape structure and geometry. However, there are rich communications between the disentangled structure and geometry VAEs during both encoding and decoding procedures, since the part geometry is generated under the shape structure guidelines while the shape structure leverages the produced part geometry for effective training. Such communication is necessary to ensure the compatibility of the generated shape structure and geometry.

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. Finally, we present our 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.

3.1 Disentangled Shape Representation

We adapt the hierarchical part segmentation in PartNet [Mo et al. 2019c] 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 highly synergistic shape representation for shape structure and geometry, where we represent each 3D shape as a pair of a structure hierarchy and a geometry hierarchy. In our disentangled representation (see Figure 2), a structure hierarchy abstracts away the part geometry and only describes a symbolic part hierarchy with part structures and relationships, namely $(\langle l_1, l_2, \cdots, l_N \rangle, \mathbf{H}, \mathbf{R})$, while a geometry hierarchy describes the part geometry $\langle G_1, G_2, \cdots, G_N \rangle$. There is a bijective mapping between the tree nodes of the structure and geometry hierarchies

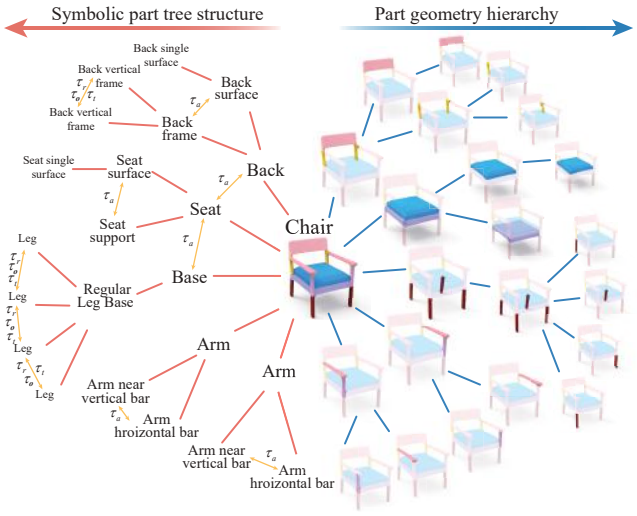


Fig. 2. An example showing the proposed disentangled but highly synergistic representation of shape geometry and structure hierarchies. There is a bijective mapping between the tree nodes in the two hierarchies. In the structure hierarchy, we consider symbolic part semantics and a rich set of part relationships (orange arrows), such as adjacency (τ_a), transnational symmetry (τ_t), reflective symmetry (τ_r) and rotational symmetry (τ_o). In the part geometry hierarchy, the part geometry is represented by mesh.

that 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_{box} with 5,402 vertices, we first calculate the oriented bounding box (OBB) B_i of each part P_i and deform G_{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 $X_i \in \mathbb{R}^{V \times 9}$ captures the local rotation and scale information in a one-ring neighbour 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 3 (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 3 (b), i.e., $G_i = (X_i, c_i)$.

Geometry Hierarchy. The geometry hierarchy for a 3D shape is a hierarchy of part geometries $\langle G_1, G_2, \cdots, G_N \rangle$ (see Figure 2 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 $(\langle l_1, l_2, \cdots, l_N \rangle, \mathbf{H}, \mathbf{R})$ as the structure representation for a shape, inspired by a recent work PT2PC [Mo et al. 2020]. Figure 2 (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 $\langle G_1, G_2, \cdots, G_N \rangle$ and the structure hierarchy $(\langle l_1, l_2, \cdots, l_N \rangle, \mathbf{H}, \mathbf{R})$ 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.

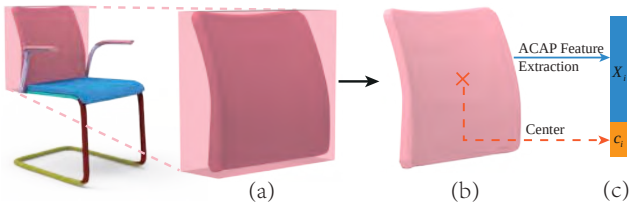


Fig. 3. We present (a) the non-rigid registration process that deforms a box mesh to a part geometry, (b) the deformed part mesh, and (c) our proposed part geometry representation, consisting of an ACAP deformation feature [Gao et al. 2019a] and a 3-dimensional part center vector.

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 Enc_{PG} that maps the part geometry $G_i = (X_i, c_i)$ into a 128-dimensional latent feature and a conditional part geometry decoder 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 illustrates the proposed part geometry conditional VAE architecture. The encoder network 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 f_i^G for part P_i. The decoder network Dec_{PG} decodes the part ACAP feature $\hat{X}_i$ and the part center $\hat{c}_i$ 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 G_{box} to reconstruct the part mesh $\hat{G}_i$ and the reconstructed center $\hat{c}_i$ move the part mesh to the correct position in the shape space.

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_i^S. The reason is three-fold: firstly, PartNet gives far more part semantic labels than the SDM-NET data, where training separate networks for different part semantics is extremely costly and empirically hard to converge; secondly, the data sample for some rare part categories is not sufficient to train a separate network; lastly, our conditional PartVAE can be conditioned on structure codes summarizing the part semantics and sub-hierarchy information, allowing effective specialization for part geometry generation given different structure contexts.

In summary, the conditional encoder Enc_{PG} takes as inputs a part geometry $G_i = (X_i, c_i)$ and a structure code condition f_i^S summarizing certain part semantics and its structural context information and outputs a latent part embedding $f_i^G = Enc_{PG}(G_i, f_i^S)$. And the

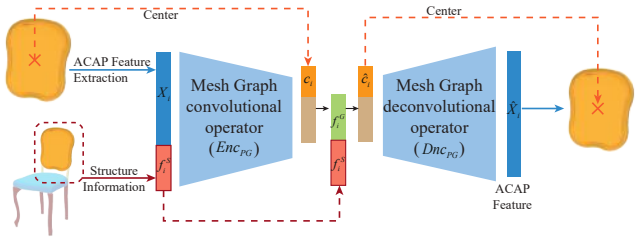


Fig. 4. The architecture of our conditional part geometry variational autoencoder. For a single part mesh geometry, the encoder maps the part ACAP feature and its center position into a 128-dimensional geometric latent code, while the decoder reconstructs the part geometry by decoding the ACAP feature and the center vector. Both networks are conditional on the part structure information along the structure hierarchy to generate specialized part geometry for different structure contexts.

conditional decoder Dec_{PG} learns to reconstruct $\hat{G}_i$ from a geometry latent code f_i^G and the current structural context information f_i^S, namely, $\hat{G}_i = (\hat{X}_i, \hat{c}_i) = Dec_{PG}(f_i^G, f_i^S)$. To train the proposed conditional PartVAE, we define the loss as follows.

$$\mathcal{L}_{\text{cond-PartVAE}} = \lambda_1 \mathcal{L}^{\text{recon}}_{\text{cond-PartVAE}} + \mathcal{L}^{\text{KL}}_{\text{cond-PartVAE}} \tag{1}$$

where $\mathcal{L}^{\text{recon}}_{\text{cond-PartVAE}} = \|\hat{X}_i - X_i\|_2^2 + \|\hat{c}_i - c_i\|_2^2$ is the reconstruction loss and $\mathcal{L}^{\text{KL}}_{\text{cond-PartVAE}}$ is the standard KL divergence loss to encourage the learned embedding space to be close to a unit multivariate Gaussian distribution.

3.3 Disentangled Geometry and Structure VAEs

To learn disentangled latent spaces for shape geometry and structure, we design two Variational Autoencoders (VAE) with Recursive Neural Network (RvNN) encoders and decoders that are trained in a disentangled but tightly coupled manner. Figure 5 provides an overview for the proposed disentangled VAEs. The geometry VAE (the blue part) and the structure VAE (the red part) learn two disentangled latent spaces for shape geometry and structure.

Though disentangled, the structure and geometry VAEs are jointly learned in a highly synergistic manner, where we build up a bijective mapping among the nodes between the structure and geometry hierarchies and allow communications across the two hierarchies. Such communications are necessary for the learning procedure since neither the structure hierarchy nor the geometry hierarchy contains sufficient information for the training.

3.3.1 *Structure VAE*

Given a structure hierarchy $(\langle l_1, l_2, \cdots, l_N \rangle, \mathbf{H}, \mathbf{R})$ 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_{RvS} propagates features from the leaf nodes to the root in a bottom-up manner according to the part hierarchy $\mathbf{H}$ and relationships $\mathbf{R}$. Inversely, the decoding process contains a recursive graph structure decoder Dec_{RvS} 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 $(\langle l_1, l_2, \cdots, l_N \rangle, \mathbf{H}, \mathbf{R})$, 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 in the encoding stage 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_i^S.

$$f_i^S = Enc_{PS}([l_i; d_i]) \tag{2}$$

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_{RvS} gathers all children node features, performs graph message-passing along the part relationships defined in $\mathbf{R}$ among the children nodes, and finally computes f_i^S by aggregating the children nodes' features. Specifically, we have

$$f_i^S = Enc_{RvS}\left(\left\{f_j^S\right\}_{(P_i,P_j)\in\mathbf{H}}, l_i, d_i\right) \tag{3}$$

where $(P_i, P_j) \in \mathbf{H}$ denotes that part P_j is a child of P_i. The module Enc_{RvS} 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_i^S 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.

We repeatedly apply the part structure encoder Enc_{RvS} until reaching the root node P_{root}. The final root node structure feature f^S_{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 unit multivariate Gaussian distribution.

Decoders. The decoding process of a structure VAE takes a structure latent code as input and recursively decodes a symbolic

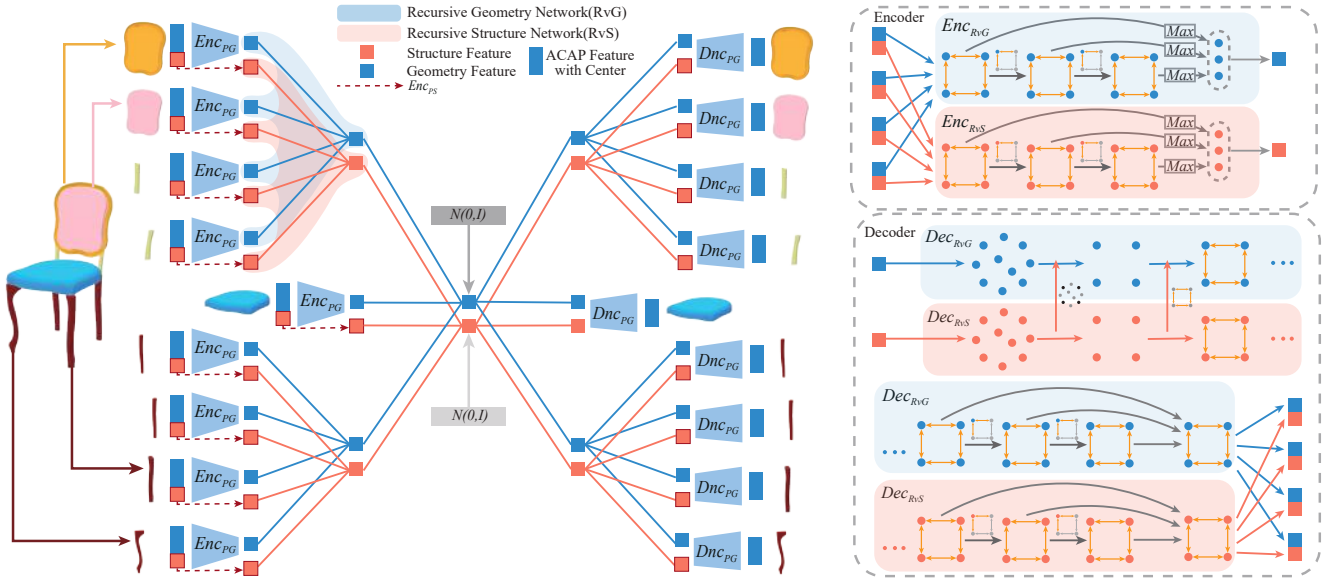


Fig. 5. We train two coupled variational autoencoders (VAEs) with recursive encoders and decoders and learn disentangled latent spaces for shape geometry and structure. The left figure illustrates the joint learning procedure of the structure VAE (shown in red) and the geometry VAE (shown in blue). In the encoding stages, the structure features summarize the symbolic part semantics and recursively compute sub-hierarchy structure contexts, while the geometry features encode the detailed part geometry for leaf nodes and propagate the geometry information along the same hierarchy. The decoding procedures of the VAEs are supervised to reconstruct the hierarchical structure and geometry information in an inverse manner. The right figure illustrates the shared message-passing mechanism used in both VAEs among related part nodes in the encoding (top) and decoding (bottom) stages, as well as the matching procedure for simultaneous training of the decoding stages for the two VAEs (middle). The blue and red nodes refer to the part nodes in the geometry and structure hierarchies respectively. For the encoding stage, there are two branches to aggregate the information (geometry/structure) of the same type siblings respectively. It performs several message-passing operations along the relation edges among the siblings and finally gathers information into a feature by max-pooling and FC layers for each branch. For the decoding stage, there are also two branches to decode one feature to its siblings for geometry and structure. It predicts existence and the edges among the existing nodes on structure branch. The geometry branch utilizes the predicted relationships. Based on this, the final node features of two branches will be updated by several message-passing operations.

structure hierarchy $(\langle \hat{l}_1, \hat{l}_2, \cdots, \hat{l}_N \rangle, \hat{\mathbf{H}}, \hat{\mathbf{R}})$ as the output. The part instance identifiers are not involved in the decoding procedure.

The recursive graph structure decoder Dec_{RvS} consumes the parent structure feature $\hat{f}_i^S$ and infers a set of children node structure features $\{\tilde{f}_{i,1}^S, \tilde{f}_{i,2}^S, \cdots, \tilde{f}_{i,10}^S\}$, where we assume there are at maximum 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 or adjacent edges $\hat{\mathbf{R}}_i$ among the existing nodes. Along the predicted edges, node features $\{\tilde{f}_{i,k}^S\}_k$ are updated via two graph message-passing operations and finally we decode a set of structure part nodes $\{\hat{f}_{j_1}^S, \hat{f}_{j_2}^S, \cdots, \hat{f}_{j_{K_i}}^S\}$, 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. In summary, we have

$$\left\{\hat{f}_{j_1}^S, \hat{f}_{j_2}^S, \cdots, \hat{f}_{j_{K_i}}^S, \hat{\mathbf{R}}_i\right\} = Dec_{RvS}\left(\hat{f}_i^S\right) \tag{4}$$

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

$$\hat{l}_i = Dec_{PS}\left(\hat{f}_i^S\right) \tag{5}$$

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 resort to the conditional part geometry decoder Dec_{PG} introduced in Sec. 3.2 and predict an oriented bounding box geometry $\hat{B}_j$ for each part $\hat{P}_j$ 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 $\hat{G}_i$ 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. We use a KL divergence loss term to train the structure latent space to get closer to the unit multivariate Gaussian distribution.

3.3.2 Geometry VAE

Given a geometry hierarchy $\langle G_1, G_2, \cdots, G_N \rangle$ encoding the part geometry of shape parts, the geometry VAE learns to map the shape geometry to a geometry latent space, disengtangled 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 Enc_{PG} and then recursively propogates 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 Enc_{RvG}. For the decoding process, we first use a recursive graph geometry decoder Dec_{RvG} that hierarchically decodes the geometry features from the root to the leaf-node parts in an inversely recursive manner. Then, we leverage a part geometry decoder Dec_{PG} 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 Enc_{PG} and decoder Dec_{PG} are conditioned on the structure context produced by the structure VAE, which allows 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 Enc_{RvG} and decoder Dec_{RvG} 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 details.

Encoders. We start from encoding each leaf node part geometry $G_i = (X_i, c_i)$ into a latent part geometry feature space. We use the conditional part geometry encoder Enc_{PG} 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_i^G, namely,

$$f_i^G = Enc_{PG}\left([X_i; c_i], f_i^S\right) \tag{6}$$

The network is conditioned on the structure code f_i^S generated in the structure VAE, in order to gain some structural context on what is the semantics for the current part 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_i^G by aggregating its children geometry node features $\{f_j^G\}_j$ through the recursive graph geometry encoder Enc_{RvG}. Similar to the design of Enc_{RvS} 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_i^G, where we have

$$f_i^G = Enc_{RvG}\left(\left\{f_j^S\right\}_{(P_i,P_j)\in\mathbf{H}}\right) \tag{7}$$

Different from the recursive graph structure encoder Enc_{RvS} 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, as illustrated in Figure 5.

We repeatedly apply the recursive graph geometry encoder Enc_{RvG} until reaching the root node P_{root}. The final root node geometry feature f_{root}^G 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 $\langle \hat{G}_1, \hat{G}_2, \cdots, \hat{G}_N \rangle$ for a shape.

The recursive graph geometry decoder Dec_{RvG} takes the parent geometry feature $\hat{f}_i^G$ as input and decodes a set of children node geometry features $\{\tilde{f}_{i,1}^G, \tilde{f}_{i,2}^G, \cdots, \tilde{f}_{i,10}^G\}$. 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 $\hat{\mathbf{R}}_i$. The decoder Dec_{RvG} then produces a final set of children nodes with the predicted part geometry features.

$$\left\{\hat{f}_{j_1}^G, \hat{f}_{j_2}^G, \cdots, \hat{f}_{j_{K_i}}^G\right\} = Dec_{RvG}\left(\hat{f}_i^S, \hat{\mathbf{R}}_i\right) \tag{8}$$

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

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 $\hat{G}_i = (\hat{X}_i, \hat{c}_i)$ from an input part geometry feature $\hat{f}_i^G$. Formally, we have

$$\hat{G}_i = Dec_{PG}\left(\hat{f}_i^G, \hat{f}_i^S\right) \tag{9}$$

Notice that the network Dec_{PG} is conditioned on the part structure code $\hat{f}_i^S$ 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.

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. Not only we demonstrate the state-of-the-art performance for

Fig. 6. Example shapes in the PartNet dataset (left) and the synthetic dataset (right).

Table 1. We summarize the data statistics of the two datasets in our experiments. We use four categories from PartNet (chairs, tables, cabinets and lamps) for the majority of our experiments and one synthetic dataset (synchairs) for evaluating disentangled shape reconstruction.

DataSet	Chair	Table	Cabinet	Lamp	SynChair
#objects	4287	3967	667	653	10800
#training objects	3407	3285	481	478	8100
#test objects	880	682	186	175	2700

structured shape generative modeling, we also illustrate how our DSM-Net can generate shape meshes with controllable structure and geometry configurations.

In this section, we present extensive experiments on the tasks of shape reconstruction, generation and interpolation and show the superior performance of our proposed method on the PartNet dataset [Mo et al. 2019c] comparing to several strong baseline methods, including StructureNet [Mo et al. 2019a], SDM-Net [Gao et al. 2019b], IM-Net [Chen and Zhang 2019] and BSP-Net [Chen et al. 2019a]). 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 other unchanged. We further benchmark our performance for disentangled shape reconstruction on a synthetic dataset. All experiments were carried out on a computer with an i9-9900K CPU, 64GB RAM, and a GTX 2080Ti GPU.

4.1 Data Preparation

We primarily use the PartNet dataset [Mo et al. 2019c] for the majority of our experiments. PartNet provides fine-grained, multi-scale and hierarchical shape part segmentation for ShapeNet [Chang et al. 2015] models. We use the four biggest and commonly used object categories for our experiments: chairs, tables, cabinets and lamps. Table 1 summarizes the data statistics. We follow the official training and testing data splits. Figure 6 (left) shows example shapes in PartNet.

All the PartNet shapes from the same object category share a canonical part template with consistent part semantics. The vertical parent-child relationships are defined consistently according to the shared part semantics set. However, the horizontal pairwise part symmetry and adjacency relationships are detected from the part annotations that provide different part structures for different shapes. Also, the part hierarchies for complex shapes usually contain more part instances than the ones for simple shapes. We directly follow the part semantics, hierarchy and relationships introduced in StructureNet [Mo et al. 2019a], but we disentangle the unified part hierarchy into two disentangled but coupled structure and geometry hierarchies (see Figure 2). Following StructureNet [Mo et al. 2019a], we only use the shapes that each parent part has a maximum number of 10 children parts.

Moreover, for quantitatively evaluating the task of disentangled shape reconstruction, we further introduce a synthetic dataset that contains 10,800 shapes (see Figure 6 (right)) with 54 kinds of shape structures and 200 geometric variations. Each shape is generated by picking one shape structure and one geometric variation, granting us the access to the ground-truth shape synthesis outcome for every configuration pair. The 54 structures are generated by enumerating structural combinations of different back types, leg styles and whether the chair has arms or not. The 200 geometric variations are created by varying the global parameters for the part geometry (e.g. the width of legs, the height of the back). The dataset is divided into the training and testing sets with a ratio of 3:1. We will release the code and data for facilitating future research.

4.2 Implementations

For our network, we train the part geometry conditional VAE and the coupled hierarchical VAEs simultaneously. The part geometry conditional VAE is used to recover the part geometric details according to the structure context. Two coupled hierarchical VAEs is aim to learn two disentangled latent spaces for shape geometry and structure in a disentangled but tightly coupled manner. Training of the whole network is optimized by the Adam solver [Kingma and Ba 2014]. All learnable parameters are initialized randomly with Gaussian distribution. For the training of whole network, we set the batch size as 16 and learning rate starting from 0.001 and decaying every 100 steps with the decay rate set to 0.9, until the loss converge with about 1000 iterations.

4.3 Shape Reconstruction

In this section, we present the shape reconstruction performance of our DSM-Net and provide quantitative and qualitative comparisons to the state-of-the-art 3D shape generative models. Figure 7 shows the shape reconstruction results for our DSM-Net on the four shape categories in PartNet. We observe that our method successfully captures both the complex shape structure and the fine-grained geometry details. Next, we propose a novel task of disentangled shape reconstruction that takes two shapes as inputs and re-synthesize a novel shape with ingredients of the structure of one shape and the geometry of the other shape. We present qualitative results on PartNet and provide quantitative evaluations on the synthetic dataset where we are provided with the ground-truth re-synthesis outputs.

Baselines. We compare DSM-Net to four state-of-the-art methods for learning 3D shape representations: IM-Net [Chen and Zhang 2019], BSP-Net [Chen et al. 2019a], StructureNet [Mo et al. 2019a] and SDM-Net [Gao et al. 2019b]. IM-Net learns an implicit function representation for encoding 3D shapes, while BSP-Net puts attention

(a) Chair (b) Table (c) Cabinet (d) Lamp

Fig. 7. The gallery of shape reconstruction results on PartNet. For each set of results, the left column shows the ground-truth targets, and the right column presents our reconstruction results. We observe that our method can capture complex shape structures and detailed part geometry at the same time.

on designing a compact mesh representations for 3D shapes. They both represent shapes 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. All the results of four baselines are reproduced by their official pre-trained models on some sub-categories of ShapeNet. In Figure 8, we can clearly see that our method achieves the best in both worlds and reconstructs shapes with more accurate structure and more detailed geometry.

Metrics. We adapt two kinds of metrics for quantitative comparisons to baseline 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 refer the readers to Fan et al. [2017] and Mo et al. [2020] for more details on the definitions of the metrics.

Table 2. Shape reconstruction quantitative evaluations. We use two geometry metrics (CD and EMD) and one structure metric (HierInsSeg). DSM-Net achieves the best geometry performance compared to all baseline methods and gets the second place in terms of the structure reconstruction accuracy. We achieve comparable HierInsSeg score with StructureNet but beats it in terms of the geometry metrics by a large margin.

DataSet	Method	Geometry Metrics		Structure Metrics
		CD ↓	EMD ↓	HierInsSeg(HIS) ↓
Chair	StructureNet	0.00973	0.43912	**0.513472**
	IM-Net	0.004795	0.117533	0.689327
	BSP-Net	0.004117	0.109377	0.743159
	SDM-Net	0.006025	0.187308	0.845902
	Ours	**0.002394**	**0.081749**	0.534247
	GT			0.321953
Table	StructureNet	0.014632	0.568401	**0.973392**
	IM-Net	0.004993	0.17932	1.139477
	BSP-Net	0.004733	0.170344	1.203952
	SDM-Net	0.007891	0.223127	1.389521
	Ours	**0.004051**	**0.084911**	0.995801
	GT			0.652784

Results. Table 2 presents the quantitative comparisons between our method and the baseline methods. Our method outperforms all baseline methods in terms of the geometry metrics, indicating that DSM-Net better captures and reconstructs shape geometry. We also beats IM-Net, BSP-Net, and SDM-Net by significantly

(a) Input Shape (b) IM-Net (c) BSP-Net (d) SDM-Net (e) StructureNet (f) Ours

Fig. 8. Shape Reconstruction Comparison with the baseline methods. DSM-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 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 third row.

Table 3. Quantitative evaluations of disentangled shape reconstruction on the synthetic data. We compare to an ablated version of our method, namely ours (no edge), since there is no applicable published baseline methods for this novel task. We observe that allowing edge communications between the structure and geometry hierarchies is essential in learning good shape representations.

Method	Geometry Metrics		Structure Metrics
	CD ↓	EMD ↓	HierInsSeg(HIS) ↓
Ours (no edge)	0.001577	0.146813	1.959482
Ours	**0.001293**	**0.061322**	**1.867651**
GT			1.79281

large margins in terms of the structure metric HierInsSeg, while achieves comparable performance to StructureNet. Although SDM-Net utilizes the structure information, the shape segmentation used by SDM-Net is very coarse. Some complex structures do not exist in its results, so the performance of HierInsSeg score on SDM-Net is worst. Figure 8 shows the qualitative comparison with other methods on the table and chair shape category. It is easy to obverse that IM-Net, BSP-Net and SDM-Net fail to generates complicated shape structures, such as the chair back bars and the table leg stretchers, while our method can successfully capture these complex shape structures. Comparing with StructureNet, we reconstruct the shape geometry more accurately. For example, StructureNet fails to reconstruct the slanted back bars for the chair in the first row, and does not recover an accurate aspect ratio of the table top surface in the third row.

Disentangled Shape Reconstruction. Our methods learn two disentangled latent manifolds (structure and geometry) for shape 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, DSM-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 9 (left) shows a set of qualitative results we experiment with on the PartNet dataset. The shapes in each row share the same geometry code while the shapes in every column have the same shape structure feature. Here, the top left and bottom right chairs are the input. The remaining chairs are generated with our DSM-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. The figure demonstrate that out method is able to re-synthesize novel shapes with pairs of geometry and structure configurations.

(a) Chair (b) Synthetic Data

Fig. 9. Disentangled shape reconstruction and interpolation results on PartNet chairs and the synthetic data. Here, the top left and bottom right chairs are the input shapes. The remaining chairs are generated automatically with our DSM-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.

We further quantitatively benchmark the performance of DSM-Net for disentangled shape reconstruction on the synthetic dataset, where we have the access to the ground-truth reconstruction results given a pair of structure and geometry configurations. Table 3 shows the quantitative results on the synthetic data. Since there is no applicable baseline methods for this novel task, we compare with an ablated version of our method: ours (no 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. We see that removing edge communications provides us worse performance, which proves the importance of maintaining the synergy between the disentangled structure and geometry hierarchies. Figure 9 (right) presents some qualitative results on the synthetic data.

4.4 Shape Generation

The main goal of DSM-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 DSM-Net and perform qualitative comparisons to several state-of-the-art baseline methods. Quantitative evaluations and user-study results further validate our superior performance than baselines. Equipped with two disentangled latent spaces for shape structure and geometry, DSM-Net also enables a novel task of generating shapes with a given shape structure or geometry patterns.

Metrics. The shape generation task aims to generate more diverse shape with complex structure and geometry, which is to cover the data distribution as much as possible. Meanwhile, a good generative models should generate realistic shapes as much as possible. Following StructureNet [Mo et al. 2019a], we measure the shape generation performance by the coverage and quality scores. The coverage score computes the average distance from a real shape to the closed generated shape, while the quality scores calculates the average distance from a generated shape to the closed real shape. The coverage score reflects if 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 eight generated shapes for each of the four object categories in PartNet. These shapes are generated by randomly sampling on two latent spaces. Our results shows the diversity of the shape set from the structure to the geometry. For each shape category, In Figure 11, comparing with SDM-Net and StructureNet, we demonstrate that our method generates shapes with more complex structure and better geometry details. In this

Fig. 10. Shape generation results. We sample random Gaussian noise vectors and use our DSM-Net to generate realistic shapes with complex structures and detailed geometry. Here we show eight generation results for each of the four object categories in PartNet.

experiment, we randomly generate the shape with different methods and then select some similar shapes to compare the quality of the generated shapes. We observe that SDM-Net can not handle the complex structure and StructureNet performs worse than ours in capturing detailed shape geometry. We show quantitative evaluation results relative to the performance of DSM-Net (*i.e.* all the reported scores are divided by the corresponding DSM-Net scores for normalization) in Table 4, where we see clear improvements over the baseline methods.

In addition, we conduct a user study to further evaluate how realistic the generated shapes are for humans. We render the shapes into images with the same setting. For every user, we asked them 10 questions. For every question, we let the user rank the three algorithms according to three different criteria (geometry, structure and overall). We shuffle the order of the algorithms each time we present the question and generate shapes from the three methods randomly. We show the results of the user study in Table 5, where we observe that our generated shape perform the best on all three criteria. We also see clearly that StructureNet is at the second place for shape structure and SDM-Net achieves better for shape geometry.

Disentangled Shape Generation. DSM-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 patterns. We demonstrate that given an input shape, DSM-Net can extract the structure code from the shape and pairs 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 two controllable generation results in Figure 12. In the experiments, given an input shape, the geometry code and structure code are extracted by running it through the encoding procedures.

Table 4. Quantitative evaluations on shape generation. We report the coverage and quality scores relative to DSM-Net (*i.e.* all the reported scores are divided by the corresponding DSM-Net scores for normalization) under the geometry metric (Chamfer-Distance) and the structure metric (HierInsSeg), comparing to StructureNet and SDM-Net as two baseline methods. We observe that DSM-Net achieves the best performance across all metrics.

Method	Geometry		Structure	
	Coverage ↑	Quality ↑	Coverage ↑	Quality ↑
SDM-Net	0.587687	0.230641	0.422925	0.479782
StructureNet	0.702391	0.766193	0.760957	0.975336
Ours	**1.00000**	**1.00000**	**1.00000**	**1.00000**

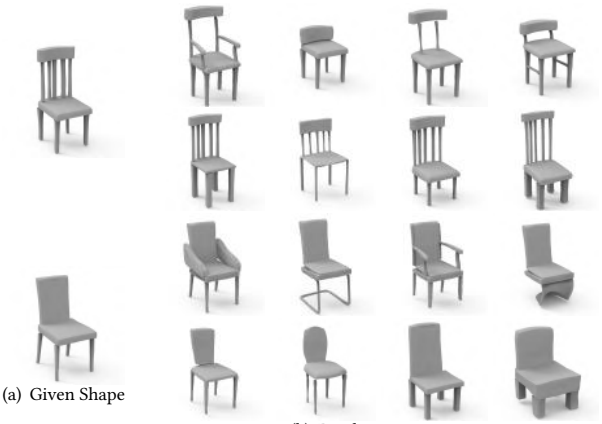
Table 5. User study results on shape generation. We show the average ranking score of the three methods: SDM-Net, StructureNet, and ours. The ranking ranges from 1 (the best) to 3 (the worst). The results are calculated based on 238 trials. We see that our method achieves the best in terms of both structure and geometry.

Method	Structure	Geometry	Overall
SDM-Net	2.6832	1.7853	1.7706
StructureNet	1.7448	2.6233	2.7014
Ours	**1.5721**	**1.5913**	**1.5279**

And then, we can keep one of them unchanged and randomly sample in another latent spaces. The two figures shows the controllable generation results. We see that when we preserve the geometry code, the chair legs usually maintain similar width and length to the input shape. And, when we keep the structure code unchanged, we are generating shapes with big geometric variations satisfying the

(a) SDM-Net (b) StructureNet (c) Ours

Fig. 11. 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. StructureNet fails at generating high-quality shape geometry and SDM-Net cannot generate shapes with complex part structures.



(a) Given Shape (b) Random generation

Fig. 12. Qualitative results for disentangled shape generation. Given an input shape (a), we extract the geometry code and structure code. We fix one of them, we random sample on the other latent space to generate the 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.

same symbolic structure hierarchy. This allows novel applications such as exploring plausible geometry variations for a given shape structure and editing shape structures while keeping similar shape geometric patterns.

4.5 Shape Interpolation

If a shape generative model can learn a smooth latent space for shape embedding, it allows users to create novel shapes by interpolating the given shapes on the latent manifold. We evaluate our DSM-Net for interpolating between shape pairs and demonstrate that our network learns a smooth latent space for shape interpolation.

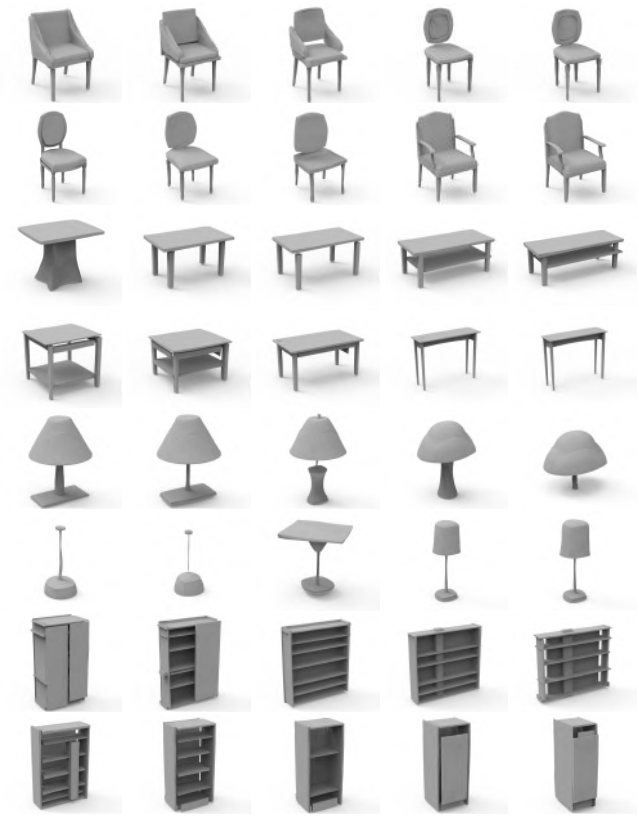


Fig. 13. Shape interpolation results on the four PartNet categories. We linearly interpolate between both the structure and geometry features of the two shapes. In the interpolated steps, we see both continuous geometry variations and discrete structure changes.

Moreover, with the help of our learned two 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.

Shape Interpolation Results. Figure 13 shows some interpolated results on four shape categories by interpolating both structure and geometry latent spaces jointly. All of interpolated results exhibits the geometric changes and structure changes. The interpolation in our learned two latent spaces lead to much more valid and functional shapes. For each interpolated step, we see both continuous geometry variations and discrete structure changes. For example, in the first row, the armrest become smaller and then disappear while the backrest change from square to round fashion in a more natural manner. In the second row, the backrest gradually becomes square, while the supporter disappears form the first chair to the second chair.

Disentangled Shape Interpolation Results. Our disentangled representations for shape structure and geometry also allows us to achieve the controllable interpolation between two shapes while keep structure or geometry unchanged. Figure 14 shows the controllable interpolation results between two shapes. Given a pair

of source and target shapes (a, b), we extract the geometry and structure code for both shapes. Then, we perform interpolation in the structure or geometry latent space while using the code of shape b in the other space. From the results, we find that our interpolation result is controllable and every interpolated shape is very realistic and reasonable. And, we see a clear disentanglement of the shape structure and geometry in the interpolated results.

4.6 Ablation Studies

We perform two ablation studies to demonstrate the necessity of the key components in our method. First, 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. Then, we validate the design choice of learning a unified conditional part geometry VAE, instead of training separate VAEs for each part semantics as used in SDM-Net [Gao et al. 2019b].

Table 6. Quantitative shape reconstruction performance comparing our full pipeline and two ablated versions: one version (Ours (no edge)) that removes the edge components and the graph message-passing modules, another version (StructureNet + Mesh) that naively combines the StructureNet backbone and SDM-Net ACAP mesh representation. We observe worse performances when we remove edges from the part hierarchies or naively replace the point cloud representation with SDM-Net ACAP mesh representation.

DataSet	Method	Geometry Metrics		Structure Metrics
		CD ↓	EMD ↓	HierInsSeg(HIS) ↓
Chair	StructureNet(SN)	0.00973	0.43912	**0.513472**
	SN + Mesh	0.003317	0.089114	0.514022
	Ours (no edge)	0.003784	0.099193	0.624833
	Ours	**0.002394**	**0.081749**	0.534247
	GT			0.321953
Table	StructureNet(SN)	0.014632	0.568401	**0.973392**
	SN + Mesh	0.004789	0.095722	0.969731
	Ours (no edge)	0.005341	0.106832	1.079425
	Ours	**0.004051**	**0.084911**	0.995801
	GT			0.652784
Cabinet	StructureNet(SN)	0.016342	0.574291	**0.579321**
	SN + Mesh	0.004019	0.159338	0.583601
	Ours (no edge)	0.004839	0.193373	0.697722
	Ours	**0.003498**	**0.110329**	0.589732
	GT			0.357964
Lamp	StructureNet(SN)	0.017311	0.712117	0.701172
	SN + Mesh	0.012937	9.259733	0.713094
	Ours (no edge)	0.013042	0.367712	0.765211
	Ours	**0.010428**	**0.19736**	**0.693458**
	GT			0.547783

Removing Part Relationships and Edges. Our network explicitly models the part relationships as horizontal edges among sibling nodes in the shape part hierarchy. Graph message-passing operations are conducted along the edges in both encoding and decoding stages. In this experiment, we compare to a no-edge version of our network where we remove the edge components and the message-passing modules. In Table 6, we see that removing the edge components gives worse results than our full pipeline. Figure 15

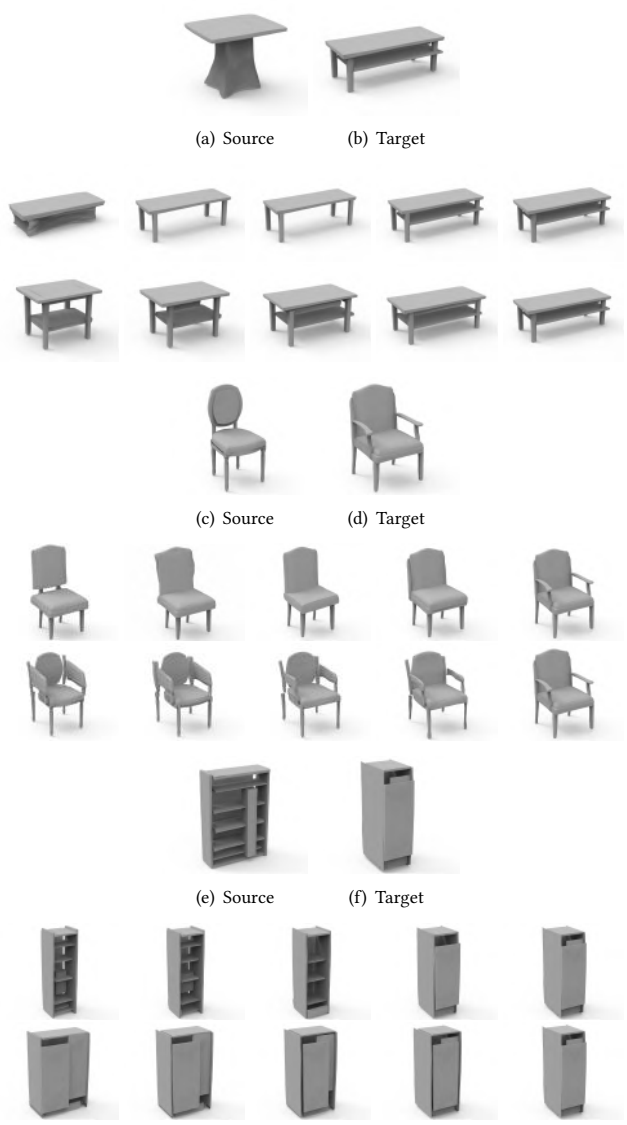


Fig. 14. Qualitative results for disentangled shape interpolation. (a,c,e) and (b,d,f) respectively show the source and target shapes. The following two rows present the interpolation result in one latent space (geometry or structure) while using the code of the target shape in the other latent space. Concretely, the first row interpolates the structure between two shapes while fixing the geometry code of target shape and the second row interpolates the geometry between two shapes while fixing the structure code of target shape. We see a clear disentanglement of the shape structure and geometry in the interpolated results.

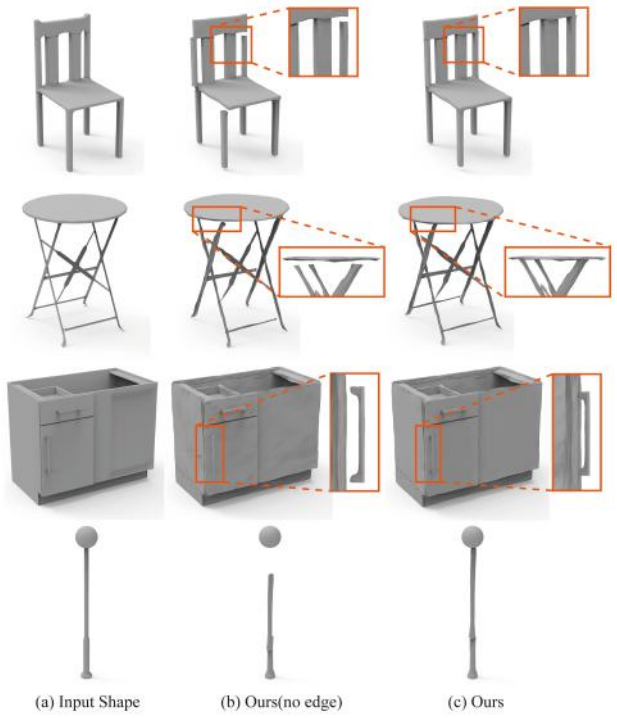


Fig. 15. Qualitative comparison on shape reconstruction with the no-edge version of our method. We can see that removing edges introduces disconnected parts in the reconstructed shapes.

illustrates three example reconstructed shapes for our method with and without edge components, where we see clearly that removing edge components creates more artifacts, such as disconnected parts.

Naive combination of StructureNet + SDM-Net ACAP mesh representation. Our network focuses on the shape structure and geometry disentanglement for controllable generation of meshes with fine geometry and complicated structure. If we do not consider the applications that are enabled by the disentangled design, such as disentangled shape generation and interpolation we presented in previous sections, there is a baseline that naively combines the StructureNet structure generation and the detailed part geometry representation (ACAP) for leaf-node mesh generation. Compared to our network with two disentangled structure and geometry VAEs, this baseline method uses one shared backbone for both. In Table 6, we clearly see that this naive baseline (SN + Mesh) obtains worse results than our DSM-Net.

Training Separate Part Geometry VAEs. In our network design, for encoding and decoding leaf-node part geometry, we train a unified part geometry VAE that is conditional on the part semantic labels and structure contexts. SDM-Net, however, proposes to use separate part geometry VAEs for different part semantics. We argue that while it is preferable in the SDM-Net experiments, it is very costly and ineffective to train separate networks on the PartNet data, where we have more fine-grained part categories than the SDM-Net dataset. In Table 7, we try the alternative method of training our network

Fig. 16. We compare the reconstructed results for a chair using a unified conditional part geometry VAE or using separate VAEs for different semantic parts. We find that using a single VAE reconstructs part geometry with more fine-grained part geometry details, e.g. the curvy armrests and the delicate sofa foots.

with 57 separate part geometry VAEs for PartNet chairs and show that the performance of shape reconstruction is significantly worse than training a unified conditional part geometry VAE. Figure 16 compares the reconstructed shapes of a chair and we see that a unified part geometry VAE learns to reconstruct part geometry with more fine-grained part geometry details, e.g. the curvy armrests and the delicate sofa foots.

Cascaded v.s. End-to-end Training. In our network, we have multiple VAEs for predicting the structure and geometry of shape and part geometric details. In our method, we train all network modules, including the part geometry VAE and the coupled hierarchical graph VAEs, in an end-to-end manner. We compare to a cascaded training scheme where we first train the part geometry VAE and then train the rest of our model. In Table 8, we evaluate the influence of two training strategies on the chair category. We observe similar performance for the two training schemes. So finally, we picked the end-to-end solution for simplicity.

Table 7. Shape reconstruction performance by our DSM-Net with using a single conditional part geometry VAE for all semantic parts or using separate VAEs for different semantic parts. We see that training one single conditional part geometry VAE is more data-efficient and thus works much better on the PartNet dataset.

Method	Geometry Metrics		Structure Metrics
	CD ↓	EMD ↓	HierInsSeg(HIS) ↓
Ours (not share one PG VAE)	0.015722	0.627309	0.972835
Ours (share one PG VAE)	**0.002394**	**0.081749**	**0.534247**
GT			0.321953

Table 8. Shape reconstruction quantitative evaluations on training strategy. We evaluate two training strategies on chair category: 1.Cascaded Training: Pre-train the PG VAE firstly, then train the coupled hierarchical VAEs based on the pre-trained PG VAE; 2. End-to-end Training: train two networks simultaneously. The separate training has similar performance to our task on geometry. So for simplicity of training network, we choose the end-to-end solution.

Method	Geometry Metrics		Structure Metrics
	CD ↓	EMD ↓	HierInsSeg(HIS) ↓
Ours (Cascaded Training)	0.002207	0.080533	0.537924
Ours (End-to-end Training)	0.002394	0.081749	0.534247
GT			0.321953

5 LIMITATIONS AND FUTURE WORKS

Our method depends on heavily annotated shape hierarchies and fine-grained part annotations for a large-scale of 3D shapes as inputs to our networks. It is a non-trivial task to obtain such data from automatic algorithms. One may consider to predict such hierarchies from training hierarchical part instance segmentation networks (as shown in PartNet [Mo et al. 2019c] 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.

In our work, we explicitly define the structure and geometry hierarchies and supervise the networks with the annotated data. It would be interesting or fresh if the network can learn to disentangle the shape structure and geometry representations automatically. It is very challenging how to learn 3D shape disentanglement in a fully unsupervised manner. We hope that our fully-supervised version can bring people's attention to this topic and future works can try to reduce the supervision.

6 CONCLUSION

In this paper, we have presented DSM-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 plausibility of generated shapes. Through extensive evaluation, our method produces high-quality shapes with complex structure and fine geometric details, outperforming state-of-the-art methods. Our method also enables intuitive and flexible 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 Royal Society Newton Advanced Fellowship (No. NAF\R2\192151), National Natural Science Foundation of China (No. 61872440 and No. 61828204), Beijing Municipal Natural Science Foundation (No. L182016), 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, ICCV*. 9418–9427.

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. 2018. Learning Representations and Generative Models for 3D Point Clouds. In *ICML*. 40–49.

Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Gleb Gusev, Djamila Aouada, and Björn Ottersten. 2018. Deep learning advances on different 3D data representations: A survey. *arXiv preprint arXiv:1808.01462* 1 (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, ICCV*. 8180–8189.

Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. 1977. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings: Image Understanding Workshop*. Science Applications, Inc Arlington, VA, 21–27.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine* 34, 4 (2017), 18–42.

Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. 2020. Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction. *arXiv preprint arXiv:2003.10983* (2020).

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012* (2015).

Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas J. Guibas, and Vladlen Koltun. 2011. Probabilistic reasoning for assembly-based 3D modeling. *ACM Trans. Graph.* 30 (2011), 35.

Siddhartha Chaudhuri, Daniel Ritchie, Jiajun Wu, Kai Xu, and Hao Zhang. 2020. Learning Generative Models of 3D Structures. *Computer Graphics Forum (Eurographics STAR)* (2020).

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*. 2172–2180.

Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. 2019a. BSP-Net: Generating Compact Meshes via Binary Space Partitioning. *arXiv preprint arXiv:1911.06971* (2019).

Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. 2019b. BAE-NET: branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 8490–8499.

Zhiqin Chen and Hao Zhang. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5939–5948.

Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. 2020. Implicit functions in feature space for 3D shape reconstruction and completion. *arXiv preprint arXiv:2003.01456* (2020).

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 2019. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3075–3084.

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 *ECCV*. Springer, 628–644.

Angela Dai and Matthias Nießner. 2019. Scan2mesh: From unstructured range scans to 3D meshes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5574–5583.

Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 2019. Learning Elementary Structures for 3D Shape Generation and Matching. *NeurIPS* (2019).

Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J. Guibas. 2020. Curriculum DeepSDF. arXiv:2003.08593 [cs.CV]

Anastasia Dubrovina, Fei Xia, Panos Achlioptas, Mira Shalah, and Leonidas J. Guibas. 2019. Composite Shape Modeling via Latent Space Factorization. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (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 *Proceedings of the IEEE conference on computer vision and pattern recognition*. 605–613.

Matheus Gadelha, Rui Wang, and Subhransu Maji. 2018. Multiresolution tree networks for 3D point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 103–118.

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

Lin Gao, Yu-Kun Lai, Jie Yang, Zhang Ling-Xiao, Shihong Xia, and Leif Kobbelt. 2019a. Sparse data driven mesh deformation. *IEEE transactions on visualization and computer graphics* (2019).

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 (Proceedings of ACM SIGGRAPH Asia 2019)* 38, 6 (2019), 243:1–243:15.

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

Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. 2016. Learning a predictable and generative vector representation for objects. In *ECCV*. Springer, 484–499.

Georgia Gkioxari, Jitendra Malik, and Justin Johnson. 2019. Mesh R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*. 9785–9795.

Aleksey Golovinskiy and Thomas Funkhouser. 2009. Consistent segmentation of 3D models. *Computers & Graphics* 33, 3 (2009), 262–269.

Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 2018. 3D semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 9224–9232.

Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 2018. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *CVPR*.

Ruizhen Hu, Lubin Fan, and Ligang Liu. 2012. Co-segmentation of 3D shapes via subspace clustering. In *Computer graphics forum*, Vol. 31. Wiley Online Library, 1703–1713.

Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G Kim, and Ersin Yumer. 2017. Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Transactions on Graphics (TOG)* 37, 1 (2017), 1–14.

Qixing Huang, Vladlen Koltun, and Leonidas Guibas. 2011. Joint shape segmentation with linear programming. In *Proceedings of the 2011 SIGGRAPH Asia Conference*. 1–12.

Anastasia Ioannidou, Elisavet Chatzilari, 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.

Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. 2020. Local Implicit Grid Representations for 3D Scenes. *arXiv preprint arXiv:2003.08981* (2020).

Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 2017. 3D shape segmentation with projective convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3779–3788.

Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. 2012. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–11.

Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7122–7131.

Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. 2018. RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5010–5019.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4401–4410.

Vladimir G. Kim, Wilmot Li, Niloy J. Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. 2013. Learning Part-based Templates from Large Collections of 3D Shapes. *Transactions on Graphics (Proc. of SIGGRAPH)* 32, 4 (2013).

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Eric-Tuan Le, Iasonas Kokkinos, and Niloy J Mitra. 2019. Going Deeper with Point Networks. *arXiv preprint arXiv:1907.00960* (2019).

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

Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. 2018. Point cloud GAN. *arXiv preprint arXiv:1810.05795* (2018).

Jun Li, Chengjie Niu, and Kai Xu. 2019. Learning part generation and assembly for structure-aware shape synthesis. *arXiv preprint arXiv:1906.06693* (2019).

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

Yecheng Lyu, Xinming Huang, and Ziming Zhang. 2020. Learning to Segment 3D Point Clouds in 2D Image Space. *arXiv preprint arXiv:2003.05593* (2020).

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

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

Niloy J Mitra, Michael Wand, Hao Zhang, Daniel Cohen-Or, Vladimir Kim, and Qi-Xing Huang. 2014. Structure-aware shape processing. In *ACM SIGGRAPH 2014 Courses*. 1–21.

Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. 2019a. StructureNet: Hierarchical Graph Networks for 3D Shape Generation. *ACM Transactions on Graphics (TOG), Siggraph Asia 2019* 38, 6 (2019), Article 242.

Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. 2019b. StructEdit: Learning Structural Shape Variations. *arXiv preprint arXiv:1911.11098* (2019).

Kaichun Mo, He Wang, Xinchen Yan, and Leonidas J Guibas. 2020. PT2PC: Learning to Generate 3D Point Cloud Shapes from Part Tree Conditions. *arXiv preprint arXiv:2003.08624* (2020).

Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. 2019c. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 909–918.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5115–5124.

Charlie Nash, Yaroslav Ganin, SM Eslami, and Peter W Battaglia. 2020. PolyGen: An autoregressive generative model of 3D meshes. *arXiv preprint arXiv:2002.10880* (2020).

Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE International Conference on Computer Vision*. 7588–7597.

Chengjie Niu, Jun Li, and Kai Xu. 2018. Im2struct: Recovering 3D shape structure from a single RGB image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4521–4529.

Maks Ovsjanikov, Wilmot Li, Leonidas J. Guibas, and Niloy Jyoti Mitra. 2011. Exploration of continuous variability in collections of 3D shapes. *ACM Trans. Graph.* 30 (2011), 33.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 165–174.

Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. 2020. Learning Unsupervised Hierarchical Part Decomposition of 3D Objects from a Single RGB Image. *ArXiv* abs/2004.01176 (2020).

Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. 2020. Convolutional occupancy networks. *arXiv preprint arXiv:2003.04618* (2020).

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*. 5099–5108.

Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. 2017. OctNet: Learning deep 3D representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3577–3586.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40, 2 (2000), 99–121.

Nadav Schor, Oren Katzir, Hao Zhang, and Daniel Cohen-Or. 2019. CompoNet: Learning to Generate the Unseen by Part Synthesis and Composition. In *Proceedings of the IEEE International Conference on Computer Vision*. 8759–8768.

Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. 2018. CSGNet: Neural Shape Parser for Constructive Solid Geometry. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 5515–5523.

Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 2019. 3D point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE International Conference on Computer Vision*. 3859–3868.

Ayan Sinha, Asim Unmesh, Qixing Huang, and Karthik Ramani. 2017. SurfNet: Generating 3D shape surfaces using deep residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6040–6049.

Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. 2018. SplatNet: Sparse lattice networks for point cloud

processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2530–2539.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE international conference on computer vision*. 945–953.

Chun-Yu Sun and Qian-Fang Zou. 2019. Learning adaptive hierarchical cuboid abstractions of 3D shape collections. *ACM Transactions on Graphics (TOG)* 38 (2019), 1 – 13.

Chun-Yu Sun, Qian-Fang Zou, Xin Tong, and Yang Liu. 2019. Learning adaptive hierarchical cuboid abstractions of 3d shape collections. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–13.

Minhyuk Sung, Hao Su, Vladimir G Kim, Siddhartha Chaudhuri, and Leonidas Guibas. 2017. ComplementMe: weakly-supervised component suggestions for 3D modeling. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–12.

Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. 2017. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *Proceedings of the IEEE International Conference on Computer Vision*. 2088–2096.

N Joseph Tatro, Stefan C Schonsheck, and Rongjie Lai. 2020. Unsupervised Geometric Disentanglement for Surfaces via CFAN-VAE. *arXiv preprint arXiv:2005.11622* (2020).

Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. 2019. Learning to Infer and Execute 3D Shape Programs. *ArXiv* abs/1901.02875 (2019).

Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. 2017. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2635–2643.

Diego Valsesia, Giulia Fracastoro, and Enrico Magli. 2018. Learning Localized Generative Models for 3D Point Clouds via Graph Convolution. (2018).

Oliver Van Kaick, Kai Xu, Hao Zhang, Yanzhen Wang, Shuyang Sun, Ariel Shamir, and Daniel Cohen-Or. 2013. Co-hierarchical analysis of shape structures. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. 2018. Pixel2mesh: Generating 3D mesh models from single RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 52–67.

Weiyue Wang, Duygu Ceylan, Radomír Mech, and Ulrich Neumann. 2019b. 3DN: 3D Deformation Network. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 1038–1046.

Yifan Wang, Noam Aigerman, Vladimir G. Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. 2019a. Neural Cages for Detail-Preserving 3D Deformations. *ArXiv* abs/1912.06395 (2019).

Yiqun Wang, Jing Ren, Dong-Ming Yan, Jianwei Guo, Xiaopeng Zhang, and Peter Wonka. 2020. MGCN: Descriptor Learning using Multiscale GCNs. *ACM Trans. on Graphics (Proc. SIGGRAPH)* (2020).

Yanzhen Wang, Kai Xu, Jun Li, Hao Zhang, Ariel Shamir, Ligang Liu, Zhiquan Cheng, and Yueshan Xiong. 2011. Symmetry hierarchy of man-made objects. In *Computer graphics forum*, Vol. 30. Wiley Online Library, 287–296.

Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. 2017. MarrNet: 3D shape reconstruction via 2.5D sketches. In *NIPS*. 540–550.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. 2016. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *NIPS*. 82–90.

Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. 2019b. PQ-NET: A Generative Part Seq2Seq Network for 3D Shapes. *arXiv preprint arXiv:1911.10949* (2019).

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*. 1912–1920.

Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. 2019a. SAGNet: Structure-aware generative network for 3D-shape modeling. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

Taihong Xiao, Jiapeng Hong, and Jinwen Ma. 2018a. DNA-GAN: Learning Disentangled Representations from Multi-Attribute Images. In *ICLR Workshops*.

Taihong Xiao, Jiapeng Hong, and Jinwen Ma. 2018b. ELEGANT: Exchanging Latent Encodings with GAN for Transferring Multiple Face Attributes. In *ECCV*.

Yun-Peng Xiao, Yu-Kun Lai, Fang-Lue Zhang, Chunpeng Li, and Lin Gao. 2020. A Survey on Deep Geometry Learning: From a Representation Perspective. *arXiv preprint arXiv:2002.07995* (2020).

Kai Xu, Vladimir G Kim, Qixing Huang, Niloy Mitra, and Evangelos Kalogerakis. 2016. Data-driven shape analysis and processing. In *SIGGRAPH ASIA 2016 Courses*. 1–38.

Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 2019. DISN: Deep implicit surface network for high-quality single-view 3D reconstruction. In *Advances in Neural Information Processing Systems*. 490–500.

Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. 2016. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *NIPS*. 1696–1704.

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. 2019. PointFlow: 3D point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision*. 4541–4550.

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. 2018. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 206–215.

Li Yi, Leonidas Guibas, Aaron Hertzmann, Vladimir G Kim, Hao Su, and Ersin Yumer. 2017. Learning hierarchical shape segmentation and labeling from online repositories. *arXiv preprint arXiv:1705.01661* (2017).

Kangxue Yin, Zhiqin Chen, Hui Huang, Daniel Cohen-Or, and Hao Zhang. 2019. LOGAN: Unpaired Shape Transform in Latent Overcomplete Space. *ACM Transactions on Graphics(Special Issue of SIGGRAPH Asia)* 38, 6 (2019), 198:1–198:13.

Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. 2019. PartNet: A Recursive Part Decomposition Network for Fine-grained and Hierarchical Shape Segmentation. In *CVPR*. to appear.

Mehmet Ersin Yümer and Niloy Jyoti Mitra. 2016. Learning Semantic Deformation Flows with 3D Convolutional Networks. In *ECCV*.

Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 2019. 3D point capsule networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1009–1018.

Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas J. Guibas, and Hao Zhang. 2020. AdaCoSeg: Adaptive Shape Co-Segmentation with Group Consistency Loss. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Renjiao Yi, and Hao Zhang. 2018a. SCORES: Shape composition with recursive substructure priors. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–14.

Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. 2018b. Visual object networks: Image generation with disentangled 3D representations. In *Advances in neural information processing systems*. 118–129.

Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 2017. 3D-PRNN: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 900–909.