Neural Template: Topology-aware Reconstruction and Disentangled Generation of 3D Meshes

Ka-Hei Hui[1*] Ruihui Li[2,1*] Jingyu Hu[1] Chi-Wing Fu[1]
[1]The Chinese University of Hong Kong [2] Hunan University
{khhui,jyhu,cwfu}@cse.cuhk.edu.hk liruihui@hnu.edu.cn

Abstract

This paper introduces a novel framework called DT-Net for 3D mesh reconstruction and generation via Disentangled Topology. Beyond previous works, we learn a topology-aware neural template specific to each input then deform the template to reconstruct a detailed mesh while preserving the learned topology. One key insight is to decouple the complex mesh reconstruction into two sub-tasks: topology formulation and shape deformation. Thanks to the decoupling, DT-Net implicitly learns a disentangled representation for the topology and shape in the latent space. Hence, it can enable novel disentangled controls for supporting various shape generation applications, e.g.*, remix the topologies of 3D objects, that are not achievable by previous reconstruction works. Extensive experimental results demonstrate that our method[1] is able to produce high-quality meshes, particularly with diverse topologies, as compared with the state-of-the-art methods.*

1. Introduction

Polygonal meshes, as a compact 3D shape representation, are widely used in many applications, such as modeling, rendering, and animation. In recent years, generative modeling and reconstruction of 3D meshes has received increasing interest and we may also guide the generative process by using various forms of input, *e.g.*, images [22,43,63] and point sets [10,14,25]. Yet, typical challenges remain—how to deal with the diverse topologies of 3D meshes, and also how to effectively provide high-level controls for new shape generation, *e.g.*, in a topology-aware manner.

To directly reconstruct a 3D mesh, one popular scheme is to learn to deform the vertices of an initial template [5, 37,43,52,56,62,63], *e.g.*, a manually-defined skeleton or a universal sphere, into the target mesh. However, the topologies of the final reconstructed meshes are typically limited by the template model. To address this, other works learn to cover a 3D mesh with planar or curved patches [22,64]; yet, the visual quality is often tampered due to the patch misalignment, so the resulting meshes often have rough surface appearance. While other 3D representations such as voxels [20,61,65,66,71,72], point clouds [2,16,28], and implicit functions [3,21,39,44,55] have been explored, these representations typically require conversions to meshes via a post-processing step for supporting visual applications.

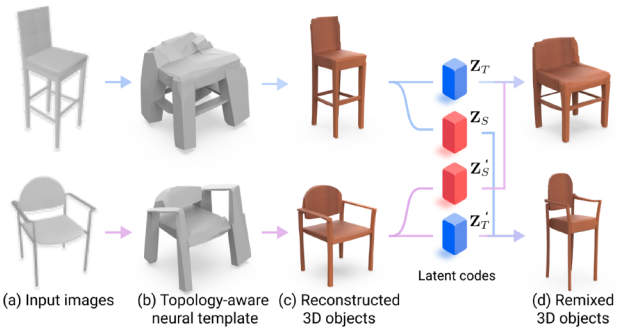

Figure 1. Our DT-Net learns to construct a *topology-aware neural template* (b) adapted to the input (a) and then deform it towards an accurate 3D mesh while preserving the initial (learned) topology. This decoupled design enables a disentangled latent representation of *topology* ($\mathbf{Z}_T$) and *shape* ($\mathbf{Z}_S$), promoting controllable 3D mesh generation, *e.g.*, remixing codes for object re-synthesis.

Another drawback is that most works focus on capturing the mesh geometry directly in a single step, without providing high-level interpretability, *e.g.*, structure or topology. So, it is particularly hard to control the mesh generation process. Some recent works tried to address this shortcoming by generating objects using parts and parts composition, *e.g.*, in terms of voxels [68], point clouds [40,67], and meshes [17]. While the approach allows certain part-aware generation, these works highly rely on the availability and the quality of the extra parts annotations.

In this paper, we present a novel framework, namely *DT-Net*, for 3D mesh reconstruction and generation via disentangled topology (DT). Distinctively, DT-Net enables the reconstruction of high-quality 3D meshes with diverse topologies, well-adapting to the input, *e.g.*, images or voxels. Also, our novel design facilitates controllability in the generative process, since DT-Net implicitly learns a disentangled latent representation for the topology and shape. Therefore, we can achieve disentangled mesh generations with separate topology and shape manipulations.

*Joint first authors

[1]Code available at https://github.com/edward1997104/Neural-Template.

Figure 1 illustrates the pipeline of DT-Net. Beyond previous works, we learn a topology-aware neural template (*e.g.*, genus of chairs) that fits each input then deform the template to reconstruct a detailed mesh. A key insight behind our design is that we decouple the mesh reconstruction into two sub-tasks: (i) topology formation for adapting various topologies; and (ii) shape deformation for reconstructing accurate objects while respecting their initial topologies. Our decoupling scheme eases the learning process and accounts for the topology, while enhancing the reconstruction quality and enriching mesh generation with diverse topologies. Another important design is that we extract a topology code (blue) and a shape code (red) from the input, to guide the learning of the two decoupled sub-tasks, respectively. By doing so, two key aspects of 3D objects, topology and shape, can be jointly learned to ensure the reconstruction plausibility, while being disentangled in the latent space, for enabling novel disentangled controls in the mesh generation process; see Figure 1 (right). Please refer to Section 3.2 for further elaborations on our framework.

Method-wise, we design an end-to-end framework with the topology-learning module to first learn to produce a topology-aware neural template composed of convexes. To decouple topology learning and shape learning, we learn a family of invertible maps [23, 73] to maintain the topology between the neural template and the final reconstructed object. Also, we propose to use a dual (implicit and explicit) representation for the neural template, so it can be trainable via the implicit functions and extractable as polygonal meshes at the inference. Importantly, our approach can directly learn the topology-aware neural template without intermediate topology annotations, while well-aligning it with an inversely-deformed version of the ground-truth mesh.

Both quantitative and qualitative results show that DT-Net enables the reconstruction of high-quality meshes with diverse topologies, performing favorably over the state of the arts. Further, our method supports various generative applications via disentangled controls, which cannot be achieved by existing reconstruction-based methods.

2. Related Work

Learning-based shape synthesis and analysis have attracted increasing research interest recently, benefiting from the availability of large shape collections [6, 41] and advances in the design of generative neural networks. In this section, we briefly review the recent advances in 3D reconstruction and generative modeling. We first focus on the mesh representation of object surfaces, learned explicitly or implicitly, and then discuss related works on shape abstraction and disentangled representation learning.

Explicit surface representation has been extensively studied for 3D voxels [20, 61, 65, 66, 71, 72, 78], octrees [24, 49, 58, 64], and point clouds [2, 16, 28, 35]. However, these representations are usually restricted to low resolutions and lack an explicit topology for detailed shape reconstruction.

In contrast, polygonal mesh is an efficient and continuous surface representation with local topological information explicitly defined by the connections between vertices. Since the learning of connection relations is challenging, most mesh-based approaches strive to learn a vertex-based deformation of an initial mesh template with graph convolutions [63], MLPs [22, 59], or neural ODEs [23]. These initial meshes are either searched from a set of CAD models [26, 32, 47, 53], customized category-based templates [31, 80], or category-agnostic meshes [22, 43, 51, 52, 56, 63], such as a genus-zero ellipsoid or 2D planer patches.

While these mesh-based methods achieve finer reconstructions, the topologies of the generated objects are constrained by the template models that they deform from. Instead of manually or explicitly defining a template, we learn to produce a topology-aware neural template adapted to the input, promoting a high-quality reconstruction with varying topologies. Particularly, the disentangled topology also enables our method to support controllable shape generation, which is not achievable by the existing methods.

Implicit surface representation models a 3D shape as a level set of discrete volume or continuous field, from which we can extract a surface mesh, *e.g.*, via iso-surfacing [36]. From an input image, these methods extract a context vector then train a neural network to predict a signed distance field [3, 21, 39, 44, 55] or occupancy probabilities [12, 38] for 3D reconstruction. Some recent works attempted to adopt extra information, e.g., camera pose [34, 69, 70] and shape skeletons [56, 57], to enhance the 3D reconstructions.

While these methods improve the reconstruction quality, they lack interpretability on the 3D structure or topology. In this work, we propose to implicitly learn a disentangled representation for the topology and shape, facilitating novel controls on the 3D mesh generation process.

Shape abstraction aims to coarsely approximate shapes with few primitives like cuboids [42, 54, 60, 79], superquadrics [46, 48, 50], and spheres [25]. Recent works [10, 14, 18, 19, 45] also leverage a structured set of implicit primitives to compose shapes. With primitives defined explicitly, these methods enable a direct extraction of 3D meshes. We draw inspiration from them to design our framework.

Disentangled representations have been widely studied in image generation, allowing manipulations separately in different aspects, *e.g.*, texture style [27, 33], facial attribute [9, 30], *etc.* For disentanglement in 3D shapes [1, 4, 77], some existing works focus on specific categories such as human faces and animal bodies. Alternatively, with additional part annotations [41], some recent works [17, 40, 67, 68, 74] tried to achieve certain part-based disentanglement by encoding

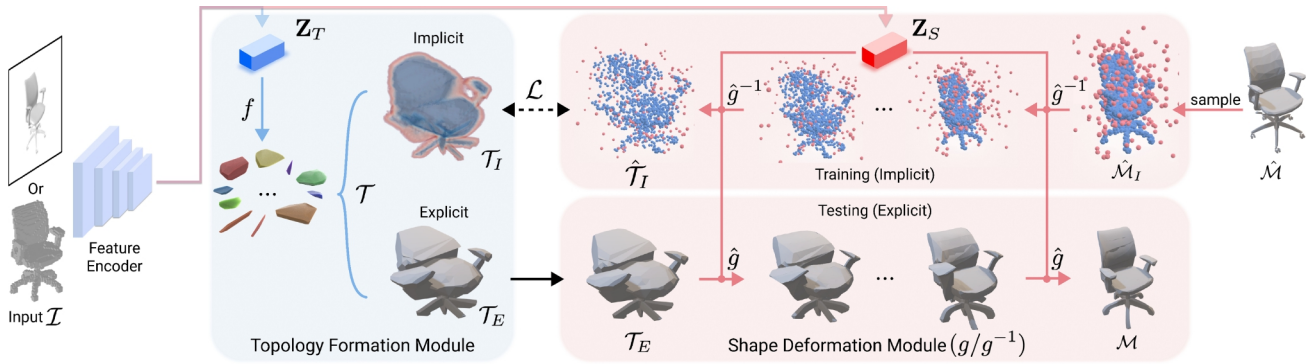

Figure 2. Overview of our DT-Net framework. Given an input, either a single-view image or 3D voxels, the encoder predicts two separate feature vectors: topology code $\mathbf{Z}_T$ and shape code $\mathbf{Z}_S$. Then, from $\mathbf{Z}_T$, we produce neural template $\mathcal{T}$ with an implicit representation $\mathcal{T}_I$ and an explicit representation $\mathcal{T}_E$ through f in the topology formation module. During the inference, we progressively deform $\mathcal{T}_E$ by function g in the shape deformation module conditioned on $\mathbf{Z}_S$ to obtain the final reconstructed shape $\mathcal{M}$. We supervise the training by using the occupancy pairs $\hat{\mathcal{M}}_I$ sampled in the shape space from the ground-truth mesh $\hat{\mathcal{M}}$. Also, we inversely map $\hat{\mathcal{M}}_I$ to the topology space by inverse function g^{-1} to produce $\hat{\mathcal{T}}_I$ for evaluating the corresponding occupancy on the implicit template representation $\mathcal{T}_I$, promoting a correct alignment between the learned topology $\mathcal{T}_I$ and the inversely-deformed shape $\hat{\mathcal{T}}_I$ by the loss $\mathcal{L}$.

parts separately and composing parts into objects via decoding. Yet, the above works depend greatly on the availability and quality of the parts or structures annotations. In contrast, our new approach decouples the reconstruction process into topology formation and shape deformation, promoting topology and shape disentanglement automatically, *without* requiring these annotations as supervision.

Other related works. Our work shares some conceptual similarities with neural cages [75], as both predict an input-adaptive mesh (template (our) or cage [75]) for further deformation. Yet, our objectives and applications are very different. Also, we noticed few recent works [15, 76] learn a shared implicit-field-based template per category for modeling dense correspondence among shapes. Different from them, we learn a topology template adapted to each input for enhancing 3D reconstructions with diverse topologies.

3. Method

3.1. The DT-Net Framework

Figure 2 shows an overview of our DT-Net framework, which consists of two modules, the *topology formulation module* and *shape deformation module*. Given input $\mathcal{I}$, which can be a 2D image or a 3D voxelized data, DT-Net first encodes it to produce two separate feature vectors, the *topology code* $\mathbf{Z}_T$ and *shape code* $\mathbf{Z}_S$. To match the given input, the topology formulation module takes $\mathbf{Z}_T$ to generate topology-aware neural template $\mathcal{T}$, whereas the shape deformation module takes $\mathbf{Z}_S$ as a guidance to refine $\mathcal{T}$ to produce the final output $\mathcal{M}$ with geometric details. In the topology formation module, we learn function f to compose $\mathcal{T} = f(\mathbf{Z}_T)$ using a set of learned convexes. Then in the shape deformation module, we learn an *invertible* homeomorphic flow function g to progressively deform $\mathcal{T}$ towards $\mathcal{M} = g(\mathbf{Z}_S, \mathcal{T})$. Note that both f and g are implemented as neural networks; see the details in Section 3.3.

Very importantly, we design $\mathcal{T}$ to have a *dual representation*; see again Figure 2. The *explicit representation* $\mathcal{T}_E$ is in the form of 3D meshes (*i.e.*, vertices and faces on object surface), while the *implicit representation* $\mathcal{T}_I$ is in the form of an implicit field (*i.e.*, an occupancy function that indicates whether any given query point is inside/outside the object). By this design, the training of DT-Net (essentially f and g) can be differentiable via the implicit representations (see the top branch in Figure 2); see more details later in this section. On the other hand, during the inference (see the bottom branch), $\mathcal{T}_E$ and $\mathcal{M}$ can be directly extracted as explicit meshes using the trained f and g.

Further, we refer to the 3D space of $\mathcal{T}$ as the *topology space* and the 3D space of $\mathcal{M}$ as the *shape space*. To obtain a continuous gradient between the two spaces, we learn inverse function g^{-1} from the shape space to topology space, *i.e.*, $\hat{\mathcal{T}}_I = g^{-1}(\mathbf{Z}_S, \hat{\mathcal{M}}_I)$. As shown in Figure 2 (top right), during the training, we sample occupancy field $\hat{\mathcal{M}}_I$ (*i.e.*, point coordinates and occupancy values) from the ground-truth mesh $\hat{\mathcal{M}}$ in the shape space. Using $\hat{\mathcal{M}}_I$, we can then construct $\hat{\mathcal{T}}_I$ using g^{-1} and formulate a regularization in the topology space as $\mathcal{L}(\hat{\mathcal{T}}_I, \mathcal{T}_I)$:

$$\min_{f,g} \mathcal{L}(g^{-1}(\mathbf{Z}_S, \hat{\mathcal{M}}_I), f(\mathbf{Z}_T)). \quad (1)$$

This optimization function defines how well the inversely-transformed implicit shape $\hat{\mathcal{T}}_I$ (from g^{-1}) aligns with the composed implicit neural template $\mathcal{T}_I$ (from f).

Implicit representation $\mathcal{T}_I$ can be derived by the bijective mapping g between the topology and shape spaces. Since $g : R^3 \leftrightarrow R^3$ is a point-wise continuous function, $\mathcal{T}$, or more specifically $\mathcal{T}_I$, can be derived by using

$$O(\mathcal{T}_I, \hat{\mathcal{T}}_I) = O(\mathcal{T}_I, g^{-1}(\mathbf{Z}_S, p)) \quad (2)$$

where $\{p\}$ are sample points in $\hat{\mathcal{M}}_I$. To evaluate point p relative to $\hat{\mathcal{M}}$, it suffices to find whether the transformed point $g^{-1}(\mathbf{Z}_S, p)$ is inside or outside the surface of $\mathcal{T}$, calculated via an occupancy function $O(\cdot)$. In other words, points that are originally inside (outside) $\hat{\mathcal{M}}$, after an inverse transformation, should also be inside (outside) the $\mathcal{T}$ as well.

Explicit representations $\mathcal{T}_E$ and $\mathcal{M}$ are 3D meshes that have the same face set F but different vertex sets $V_{\mathcal{T}}$ and $V_{\mathcal{M}} = g(\mathbf{Z}_S, V_{\mathcal{T}})$, respectively. Here, functions g and g^{-1} map between corresponding vertices in $V_{\mathcal{T}}$ and $V_{\mathcal{M}}$. So, the final reconstructed object $\mathcal{M}$ can be obtained through a deformation (function g) from the template mesh $\mathcal{T}_E$. We can extract $\mathcal{T}_E$ by grouping a set of learned primitives, such that we can flexibly represent 3D objects of various topologies. More details will be given in Section 3.3.

3.2. Framework Design

Before elaborating on the details of the DT-Net framework, we first discuss the key ideas in framework design.

(i) *Topology-aware learning*. We learn to produce neural template $\mathcal{T}$ whose topology specifically follows the input $\mathcal{I}$, instead of manually defining a template as previous works. To adapt $\mathcal{T}$ for varying topologies, we produce it by composing geometric primitives, which are based on well-defined implicit and explicit representations [10, 14]. Thus, $\mathcal{T}$ is *trainable* via implicit functions and *extractable* directly as explicit meshes.

(ii) *Topology-preserved deformation*. To decouple topology learning and shape learning, we preserve the topology of the neural template $\mathcal{T}$ while deforming it to form the output mesh. Particularly, we learn a family of invertible maps [23, 73] between the topology space and shape space, such that we can impose various constraints on $\mathcal{T}$ from $\hat{\mathcal{M}}$ for efficiently computing its implicit and explicit representations.

(iii) *Without topology annotations.* DT-Net learns to produce the topology-aware neural template $\mathcal{T}$ directly from input $\mathcal{I}$ and ground-truth mesh $\hat{\mathcal{M}}$ *without* requiring topology annotations as the intermediate supervision. We achieve so by inversely mapping $\hat{\mathcal{M}}$ from the shape space into the topology space, *i.e.*, by inversely deforming sample points $\hat{\mathcal{M}}_I$ into $\hat{\mathcal{T}}_I$. So, DT-Net can learn to produce $\mathcal{T}_I$ in an unsupervised manner by precisely aligning $\hat{\mathcal{T}}_I$ with the learned $\mathcal{T}$, as Eq. (1).

(iv) *Topology & shape disentanglement*. Also, we provide controllability in the generation by injecting the topology code $\mathbf{Z}_T$ and shape code $\mathbf{Z}_S$ into the training of f and g, respectively. By this means, topology and shape are jointly learned to ensure plausible reconstructions, while being as disentangled as possible in the latent space. This design provides a family of novel high-level controls, such as manipulating a mesh by modifying its shape code while preserving its topology code; examples will be presented in Section 3.5.

3.3. Network Architecture

Topology formation module. We learn function f to map topology code $\mathbf{Z}_T$ to neural template $\mathcal{T} = f(\mathbf{Z}_T)$. Inspired by [10, 14, 45, 46, 60], we propose to compose the topology-aware neural template via a union of geometric primitives. As referred to the key idea (i) in Section 3.2, we adopt the formulation in [10] to group a collection of convex polyhedra to assemble the implicit field of the neural template.

Specifically, given $\mathbf{Z}_T$, we implement f using multi-layer perceptrons that first predict the parameters to define the various hyperplanes $\mathcal{H} \in R^{N_h * 4}$ (*i.e.*, $ax+by+cz+d = 0$) then group these planes to form a set of convexes $\mathcal{C}$ via a learnable binary matrix $\mathbf{B} \in R^{N_h * N_c}$ (a selective mask), where N_h and N_c denote the number of hyperplanes and convexes, respectively. Lastly, these convexes are assembled to form the neural template $\mathcal{T}$. This formulation enables an explicit representation $\mathcal{T}_E$ (*i.e.*, a union of the convexes) and also an implicit representation $\mathcal{T}_I$ (*i.e.*, a scalar function $O(\cdot)$ in Eq. (2) for indicating the occupancy: a given point is inside/outside these convexes).

Shape deformation module. We learn invertible deformation function g that preserves the topology between output object and learned neural template; see key idea (ii) above. Given $\mathbf{Z}_S$, it learns to progressively deform the neural template $\mathcal{T}$ towards the detailed surface $\mathcal{M} = g(\mathbf{Z}_S, \mathcal{T})$.

Specifically, we adopt the neural ordinary differential equation module (NODE) in [23, 73] to achieve a continuous deformation on the topologies. It defines an invertible transformation $g : R^3 \leftrightarrow R^3$ via a parameterized ODE $p_T = g(\mathbf{Z}_S, p_0) = p_0 + \int_0^T \hat{g}(\mathbf{Z}_S, p_t)\, dt$, where p_0 and p_T are input to and output from neural network $\hat{g}$ (*i.e.*, $[x, y, z]$), and T is a hyperparameter that denotes the number of deformation steps from p_0 to p_T. This integration is approximated using numerical solvers, while its gradient can be computed by using the adjoint method proposed in [8]. Due to the diffeomorphic nature of g, we can then preserve the general topology of $\mathcal{T}$ in the deformation process.

3.4. Network Training

Without topology annotations, as mentioned in Section 3.1, we propose to train DT-Net via $\mathcal{L}(\hat{\mathcal{T}}_I, \mathcal{T}_I)$. The joint-optimization function is composed of two terms:

$$\mathcal{L}(\hat{\mathcal{T}}_I, \mathcal{T}_I) = \mathcal{L}_{\text{align}} + \mathcal{L}_{\mathbf{B}}, \tag{3}$$

where $\mathcal{L}_{\text{align}}$ encourages a correct alignment between the learned topology $\mathcal{T}_I = f(\mathbf{Z}_T)$ and the inversely-deformed shape $\hat{\mathcal{T}}_I = g^{-1}(\mathbf{Z}_S, \hat{\mathcal{M}}_I)$. Also, we adopt the sparsity term $\mathcal{L}_{\mathbf{B}}$ in [10] to encourage the learned topology to be composed by a sparse set of convexes.

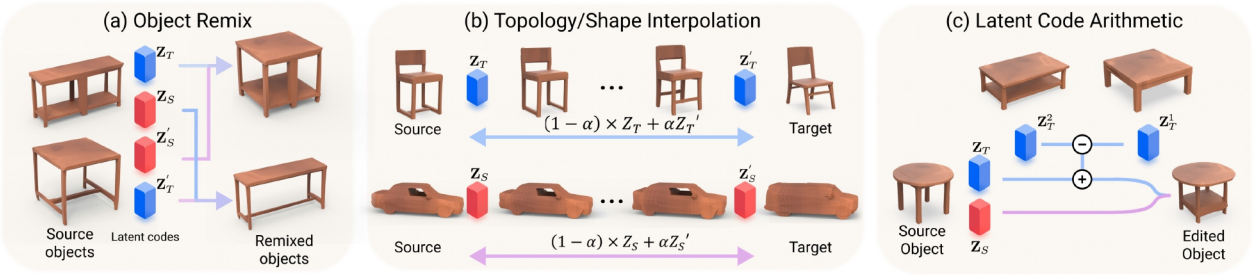

Figure 3. Our DT-Net framework learns a disentangled representation for the topology and shape, thus facilitating novel generation applications via disentangled manipulation on the topology code $\mathbf{Z}_T$ and/or the shape code $\mathbf{Z}_S$, *e.g.*, (a) remix the shape and topology of two different objects; (b) object interpolation by manipulating the topology/shape code; and (c) arithmetic operations in the latent space.

Specifically, $\hat{\mathcal{M}}_I = \{p_i', o_i\}_{i=1}^{N_p}$ denotes N_p occupancy pairs sampled from the shape space of ground-truth mesh $\hat{\mathcal{M}}$; p_i' is the i-th sample point coordinates, and o_i=1 (o_i=0) indicates that p_i' is inside (outside) the object. By inversely mapping p_i' into the topology space as $p_i = g^{-1}(\mathbf{Z}_S, p_i')$, we then obtain $\hat{\mathcal{T}}_I = \{p_i, o_i\}_{i=1}^{N_p}$ as the intermediate signal to optimize topology learning function f. For each query point $p_i \in \hat{\mathcal{T}}_I$ and associated ground-truth occupancy value o_i, $\mathcal{L}_{\text{align}}$ measures the difference between $O(\mathcal{T}, p_i)$ and o_i, promoting the network to predict the right occupancy value.

To ease the gradient flow, we adopt the two-stage training strategy in [10]: stage 1 (continuous) computes a relax approximation $\mathcal{L}^{\text{con}}(\hat{\mathcal{T}}_I, \mathcal{T}_I)$ from $\mathcal{T}_I$ to $\hat{\mathcal{T}}_I$, then stage 2 (discrete) promotes an accurate alignment $\mathcal{L}^{\text{dis}}(\hat{\mathcal{T}}_I, \mathcal{T}_I)$ between $\mathcal{T}_I$ and $\hat{\mathcal{T}}_I$. Specifically, $\mathcal{L}_{\text{align}}^{\text{con}}$ adopts a least-squares model to approximate the ground-truth occupancy value o_i, whereas $\mathcal{L}_{\text{align}}^{\text{dis}}$ adopts binary cross entropy to encourage the output occupancy value to be discrete as o_i:

$$\mathcal{L}_{\text{align}}^{\text{con}} = \frac{1}{N_p}\sum_{i=1}^{N_p}(O(\mathcal{T}_I, p_i) - o_i)^2$$

$$\text{and} \quad \mathcal{L}_{\text{align}}^{\text{dis}} = \frac{1}{N_p}\sum_{i=1}^{N_p}\big[o_i * \max(O(\mathcal{T}_I, p_i), 0) + (1-o_i) * (1 - \min(O(\mathcal{T}_I, p_i), 1))\big].$$

3.5. Shape Generation with Controllability

With the disentangled representation, *i.e.*, the topology code $\mathbf{Z}_T$ and shape code $\mathbf{Z}_S$, DT-Net enables novel forms of 3D object manipulations, opening up new possibilities for high-level object generation and re-synthesis:

- **Object Remix**. $\mathbf{Z}_T$ and $\mathbf{Z}_S$ jointly contribute to reconstructing a 3D object, so we can remix them between objects to manipulate the shape (topology) while preserving its original topology (shape); see Figure 3(a) for two re-synthesized coffee tables. Figure 4 shows more results produced by remixing different tables as sources of $\mathbf{Z}_T$ (leftmost column) and $\mathbf{Z}_S$ (top row).
- **Object Interpolation.** Also, we can produce a disentangled interpolation between objects, either on $\mathbf{Z}_T$ or $\mathbf{Z}_S$, as shown in Figure 3(b). From left to right, the chair (top) morphs towards the target, yet preserving its rectangular-like shape, whereas the car (bottom) becomes taller towards a truck with the same topology. Figure 5 shows more results of our disentangled interpolation on topology (top) and shape (bottom).
- **Latent Code Arithmetic**. With the learned smooth latent space, we can exploit arithmetic operations in the latent space. Figure 3(c) shows that we can subtract

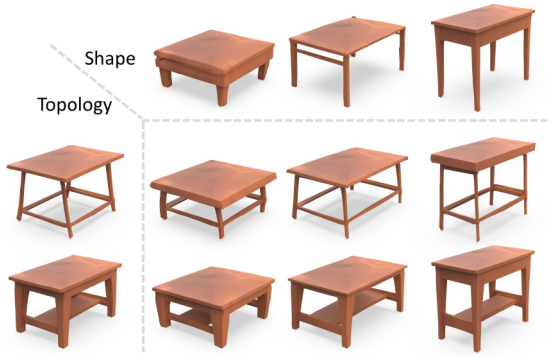

Figure 4. Remixing shape and topology. Objects on top provide the shape codes and objects on the left provide the topology codes.

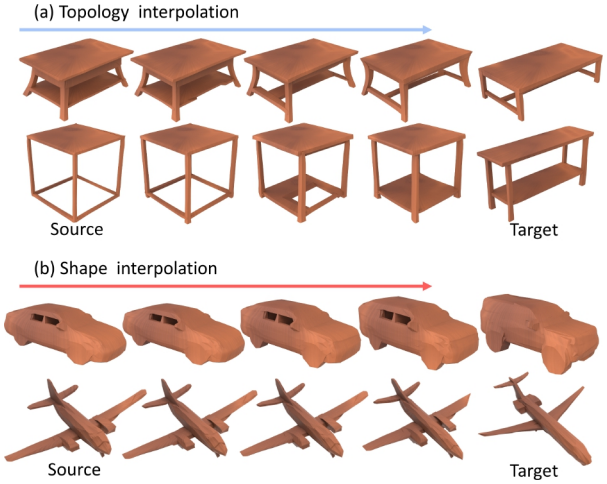

Figure 5. Object interpolation separately on topology (top) and shape (bottom). Note the smooth transitions achieved by DT-Net.

Figure 6. Galleries showcasing the results produced by our DT-Net. Each pair shows the learned topology-aware neural template (left) and the associated reconstructed object (right). The produced objects cover various shapes and diverse topologies, ranging from smooth surface (*e.g.*, car and lamp), to complex geometry (*e.g.*, chair and airplane). It is observed that the neural template visually appears like a coarse version of the final shape, even without regularizing the amplitude of the deformation module.

the topology codes of two tables, *i.e.*, with and without a storage plate, and add the difference to the topology code of another table to augment it with a plate. Figure 7 shows more results on latent code arithmetics.

Please refer to the supplement for more results.

4. Result and Evaluation

4.1. Dataset and Metric

Dataset. We employ 13 classes in ShapeNet [6] for mesh reconstruction as [10–12] and adopt input voxels directly from [24] and input images from [13]. For each reconstruction task (voxels/images), we train one model on all categories and use the same train-test split as previous works. At inference, we directly obtain the mesh of the topology-aware neural template as a union of convexes and deform it to obtain the final mesh. *Please find details on training, testing, network architecture,* etc.*, in the supplement.*

Evaluation metric. To quantitatively evaluate the predicted mesh $\mathcal{M}$ relative to ground-truth mesh $\hat{\mathcal{M}}$, we employ (i) Light field distance (LFD); inspired by human vision system [7, 10], LFD measures the visual similarity in rendered images of $\mathcal{M}$ and $\hat{\mathcal{M}}$ at different views; (ii) Point-to-surface distance (P2F) measures the minimum distance from the sampled points of $\mathcal{M}$ to the surface of $\hat{\mathcal{M}}$; and (iii) Chamfer distance (CD) measures the bidirectional shortest distance between the point samples of $\mathcal{M}$ and $\hat{\mathcal{M}}$. Importantly, LFD measures the visual quality of object surfaces, whereas P2F and CD merely account for point-wise distances. For all metrics, a lower value indicates a better performance.

4.2. Mesh Reconstruction from 3D Voxels

Gallery. Figure 6 showcases our learned neural templates (odd columns) paired with the reconstructed objects (even columns). These results manifest that our DT-Net is able to produce topology-aware templates of various connectivity and genus specific to the target objects, and the final meshes cover a wide variety of global shapes and local structures.

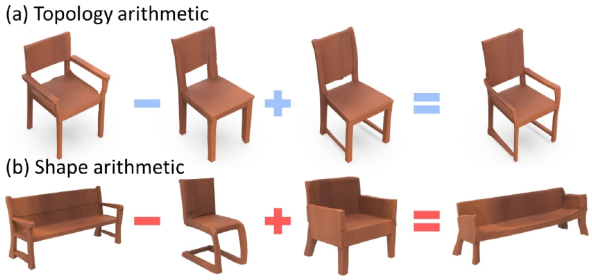

Figure 7. Arithmetic operations between different objects.

Quantitative evaluation. Beyond achieving a controllable topology-aware generation of 3D meshes, we further evaluate the quality of our generated meshes against those produced by the state-of-the-art models, IM-Net [12] and BSP-Net [10]. Using the same train-test split as [22], we directly leverage their pre-trained models provided in the original implementations. BSP-Net and our DT-Net can directly extract meshes via a union operation of primitives from input voxels of resolution 64^3. For IM-Net, we extract the final meshes via [36] from a higher resolution input (256^3). Table 1 reports the quantitative evaluation results, showing that DT-Net has good performance on most categories and its overall performance also outperforms others for all metrics. Particularly, benefited from our topology-aware neural template, DT-Net has a large improvement on object categories with high diversity in topology, *e.g.*, *chair*.

Qualitative evaluation. Figure 8 shows visual comparison results, revealing that other methods tend to produce missing parts (*e.g.*, table's beam) and less details (*e.g.*, chair's pulley). In contrast, our method can produce more complete objects that are visually the closest to the targets, and our reconstructed objects exhibit more tiny local structures (*e.g.*, airplane and gun) and manifest various object topologies. *More results are shown in the supplement.*

Table 1. Quantitative evaluations on mesh reconstruction from 3D voxels. The units of LFD, P2F, and CD are 1.0, 10^{-2}, and 10^{-3}, resp.

Metric	Method	Category													
		Mean	Plane	Bench	Cabinet	Car	Chair	Display	Lamp	Speaker	Riffle	Couch	Table	Phone	Vessel
LFD(↓)	IM-NET(256^3)	2918.9	4065.3	3452.7	1542.6	**2069.7**	2479.1	2606.2	6073.9	1763.0	5466.9	2110.7	2374.4	2109.1	**4366.5**
	BSP-NET	3026.0	4287.0	3599.	**1489.7**	2101.1	2643.1	2602.8	6384.3	1769.8	5545.1	2170.1	2471.9	2187.7	4495.2
	Ours	**2835.0**	**3955.1**	**3329.9**	1509.1	2070.4	**2368.7**	**2460.2**	**5899.3**	**1707.1**	**5333.1**	**2043.5**	**2257.6**	**2078.6**	4366.9
P2F(↓)	IM-NET(256^3)	0.820	0.597	0.739	**0.749**	**0.584**	0.876	0.821	1.543	1.045	0.794	**0.768**	0.930	**0.564**	0.864
	BSP-NET	0.899	0.677	0.826	0.755	0.654	1.016	0.889	1.859	0.985	0.830	0.793	0.946	0.632	1.062
	Ours	**0.796**	**0.542**	**0.677**	0.751	0.674	**0.847**	**0.769**	**1.422**	**0.978**	**0.651**	0.854	**0.851**	0.567	**0.843**
CD(↓)	IM-NET(256^3)	0.648	0.322	0.499	0.727	0.526	0.663	0.641	**1.351**	1.012	0.374	0.611	0.781	0.384	0.628
	BSP-NET	0.750	0.377	0.595	0.764	0.583	0.807	0.741	1.727	1.099	0.414	0.672	0.874	0.524	0.770
	Ours	**0.573**	**0.259**	**0.434**	**0.651**	**0.460**	**0.581**	**0.479**	1.442	**0.849**	**0.268**	**0.576**	**0.674**	**0.313**	**0.550**

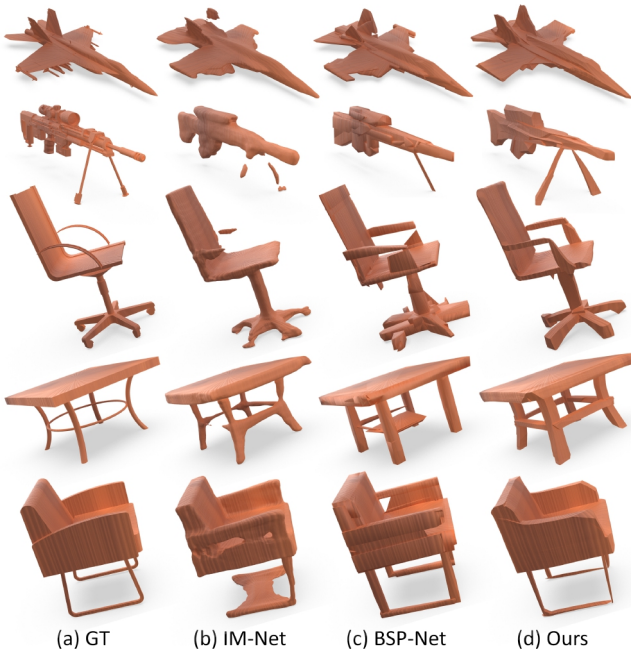

Figure 8. Visual comparison on mesh reconstruction from voxels.

4.3. Mesh Reconstruction from Single-View Images

For the single-view reconstruction task, we compare our method with two lines of works: (i) explicit methods: Pixel2Mesh [63], AtlasNet [22], and TMNet [43] that directly deform a template towards the final mesh; and (ii) implicit methods: IM-Net [12], BSP-Net [10], and DI^{2}M-Net [34] that produce implicit surfaces. For DI^{2}M-Net, their authors kindly help us generate the visual results. For other methods, we use their released implementations with the same train-test split (*i.e.*, 80%-20%) and the inputs are gray-scale images. We also noticed a very recent work [51], improved from TMNet [43] and we will make a proper comparison when the source code is available in the future.

Quantitative evaluation. Table 2 lists the overall results, showing that our method consistently outperforms the other implicit methods in terms of LFD, P2F, and CD. Note that we did not include DI^{2}M-Net, since it requires an additional camera pose as input for the training. On the other hand, comparing with the explicit methods, our method is better on LFD and comparable on the distance-based metrics CD and P2F; this might be attributed to their CD-wise regularization in the training. Also, distance-based metrics may not be ideal for measuring the quality of the reconstructed meshes (see [10, 29]), as evidenced by the visual comparison results in Figure 9. *We also show the detailed results on individual categories in the supplement.*

Table 2. Quantitative results on reconstruction from 2D images. Overall, our method is better on LFD and comparable with others on distance metrics P2F and CD. Details are shown below.

Method		Metric		
		LFD(↓)	P2F(↓)	CD (↓)
Explicit	Pixel2Mesh	4056.2	1.903	1.855
	AtlasNet	3880.9	1.289	**1.041**
	TMNet	**3765.5**	**1.285**	1.149
Implicit	IM-NET(256^3)	3559.2	1.422	1.497
	BSP-NET	3426.5	1.354	1.478
	Ours	**3388.3**	**1.294**	**1.396**

Qualitative evaluation. Figure 9 shows the visual comparison results. Referring to the ground-truth meshes (a), explicit methods (b-d) are typically hard to adapt objects of various genus, therefore further confirming our motivation for topology-aware template formulation. On the other hand, implicit methods (e-g) can describe the topology flexibly, yet tend to produce over-smooth or noisy surfaces with less details, *e.g.*, chair's armrest and boat's hull. In contrast, our method (h) can produce high-quality meshes, in which the surfaces exhibit smooth and sharp features simultaneously. *More visual results are in the supplement.*

4.4. Model Analysis and Discussion

Framework analysis. We first verify the framework design of DT-Net. In Figure 10, we compare the usage of other primitives [46, 60] such as superquadrics (b) and cuboids (c), vs. our convexes (e) for composing the topology-aware neural templates. Figure 10 (d) shows results when using another invertible neural network (INN) [45] to implement

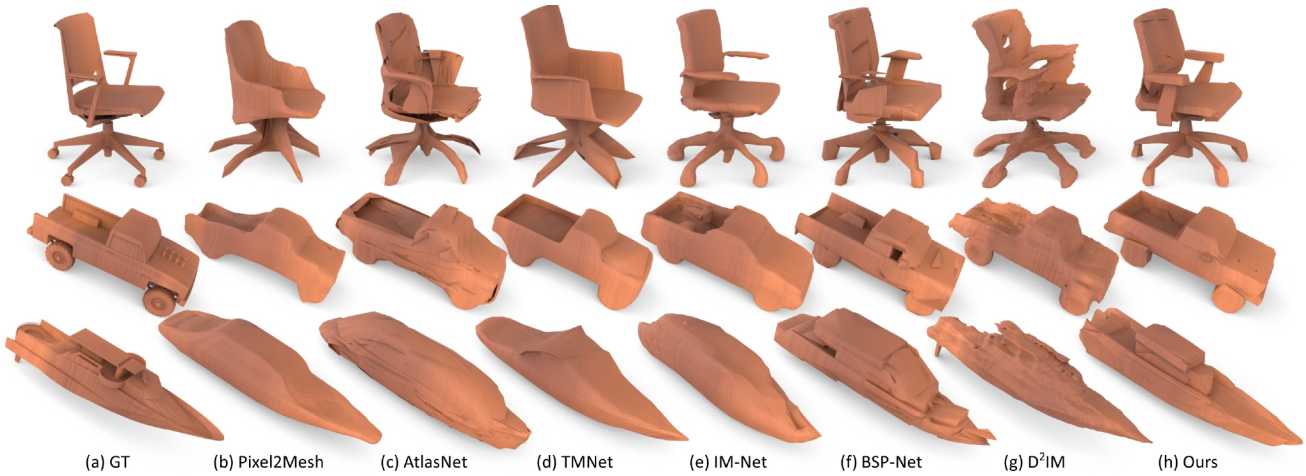

Figure 9. Visual comparison on 3D mesh reconstruction from 2D images.

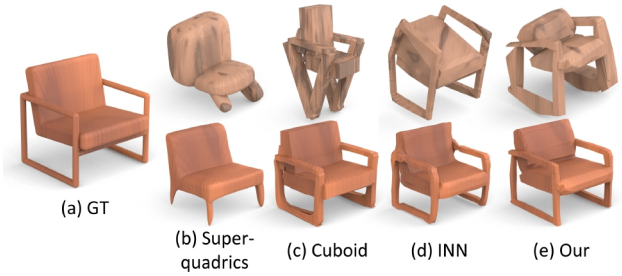

Figure 10. Given the reference mesh (a), we may use alternative representations (b-c) to compose the neural template or use INN (d) for shape deformation. Our method (e) shows better results.

the shape deformation module; *see details and evaluation results in supplement.* Generally, we take a generic design for the topology formation and shape deformation modules (see Section 3.2), meaning that we may use alternative implementations, yet our current choices provide better topological approximations and lead to better reconstructions.

Visualization of the topology space. To show the smoothness and meaningfulness of the learned topology space, we produce a visualization of the TSNE embedding for $\mathbf{Z}_T$ on chairs. From the visualization shown in *Figure 8 of the supplement*, we can see that DT-Net can learn a smooth embedding space for objects of varying topological structures and objects of similar topologies are closely clustered.

Cross-category manipulation. Since our model is trained on multiple object categories, we may conduct object remixing across different categories; see the results in Figure 11. Interestingly, we can obtain a chair-like car, which follows the car's topology and the chair's shape. *We show more cross-category results in the supplement.*

Limitation and Discussion. First, like most of the previous methods on 3D mesh generation, it is still very challenging to produce objects of extremely complex and fine structures; *see the supplement*. In the future, we aim to further formulate the topology-aware neural template in a hierarchical manner and deform it in a part-wise manner for more fine-grained reconstructions and controls. Also, since DT-Net is built on a reconstruction task, the generated new objects are still limited to the diversity of the given objects. We would like to extend it into an unsupervised generation framework and take into account voice, text, or other input modality for more intuitive object manipulations.

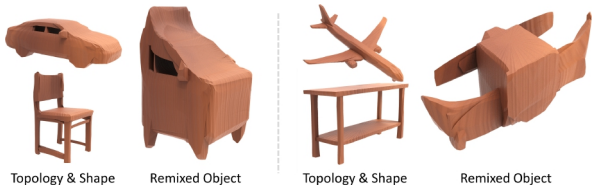

Figure 11. Cross-category object remix between different classes.

5. Conclusion

We presented a novel framework called DT-Net that enables a topology-aware mesh reconstruction and promotes mesh generation with disentangled controls. A key design is to learn to form a topology-aware neural template specific to each input then deform it to reconstruct a detailed 3D object. This scheme decouples the 3D reconstruction process into two sub-tasks, effectively accommodating for the variations in topology. Importantly, our new design provides a disentangled representation of topology and shape in the latent space, enabling controllable object generations by manipulating the learned topology code and shape code, which are not achievable by the existing reconstruction methods. Extensive experiments also manifest that our method produces high-quality meshes with diverse topologies and fine details, performing favorably over the state of the arts.

Acknowledgments. We thank anonymous reviewers for the valuable comments. This work is supported by the Research Grants Council of the Hong Kong Special Administrative Region (Project No. CUHK 14206320 & 14201921).

References

[1] Victoria Fernández Abrevaya, Adnane Boukhayma, Stefanie Wuhrer, and Edmond Boyer. A decoupled 3D facial shape model by adversarial training. In *ICCV*, pages 9419–9428, 2019. 2

[2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Learning representations and generative models for 3D point clouds. In *ICML*, pages 40–49, 2018. 1, 2

[3] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *CVPR*, pages 2565–2574, 2020. 1, 2

[4] Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan Jepson, and Sven Dickinson. Geometric disentanglement for generative latent shape models. In *ICCV*, pages 8181–8190, 2019. 2

[5] Heli Ben-Hamu, Haggai Maron, Itay Kezurer, Gal Avineri, and Yaron Lipman. Multi-chart Generative Surface Modeling. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 37(6):215:1–215:15, 2018. 1

[6] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, QiXing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 6

[7] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3D model retrieval. In *Computer Graphics Forum*, volume 22, pages 223–232, 2003. 6

[8] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural Ordinary Differential Equations. In *NeurIPS*, pages 6571–6583, 2018. 4

[9] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, pages 2180–2188, 2016. 2

[10] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. BSP-Net: Generating compact meshes via binary space partitioning. In *CVPR*, pages 45–54, 2020. 1, 2, 4, 5, 6, 7

[11] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-Net: Branched autoencoder for shape co-segmentation. In *ICCV*, pages 8490–8499, 2019. 6

[12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, pages 5939–5948, 2019. 2, 6, 7

[13] Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *ECCV*, pages 628–644, 2016. 6

[14] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. CvxNet: Learnable convex decomposition. In *CVPR*, pages 31–44, 2020. 1, 2, 4

[15] Yu Deng, Jiaolong Yang, and Xin Tong. Deformed implicit field: Modeling 3D shapes with learned dense correspondence. In *CVPR*, pages 10286–10296, 2021. 3

[16] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *CVPR*, pages 605–613, 2017. 1, 2

[17] Lin Gao, Jie Yang, Tong Wu, YuJie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 38(6):243:1–243:15, 2019. 1, 2

[18] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3D shape. In *CVPR*, pages 4857–4866, 2020. 2

[19] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *ICCV*, pages 7154–7164, 2019. 2

[20] Rohit Girdhar, David F. Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, pages 484–499, 2016. 1, 2

[21] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*, pages 3569–3579, 2020. 1, 2

[22] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. AtlasNet: A papier-mâché approach to learning 3D surface generation. In *CVPR*, pages 216–224, 2018. 1, 2, 6, 7

[23] Kunal Gupta and Manmohan Chandraker. Neural Mesh Flow: 3D manifold mesh generation via diffeomorphic flows. In *NeurIPS*, pages 1747–1758, 2020. 2, 4

[24] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3D object reconstruction. In *3DV*, pages 412–420, 2017. 2, 6

[25] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge Belongie. DualSDF: Semantic shape manipulation using a two-level representation. In *CVPR*, pages 7631–7641, 2020. 1, 2

[26] Qixing Huang, Hai Wang, and Vladlen Koltun. Single-View reconstruction via joint analysis of image and shape collections. *ACM Transactions on Graphics (SIGGRAPH)*, 34(4):87:1–87:10, 2015. 2

[27] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, pages 172–189, 2018. 2

[28] Li Jiang, Shaoshuai Shi, Xiaojuan Qi, and Jiaya Jia. GAL: Geometric adversarial loss for single-view 3D-object reconstruction. In *ECCV*, pages 802–816, 2018. 1, 2

[29] Jiongchao Jin, Akshay Gadi Patil, Zhang Xiong, and Hao Zhang. DR-KFS: A differentiable visual similarity metric for 3D shape reconstruction. In *ECCV*, pages 295–311, 2020. 7

[30] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 2

[31] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image hu-

man shape reconstruction. In *CVPR*, pages 4501–4510, 2019. 2

[32] Chen Kong, Chen-Hsuan Lin, and Simon Lucey. Using locally corresponding CAD models for dense 3D reconstructions from a single image. In *CVPR*, pages 4857–4865, 2017. 2

[33] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse Image-to-Image translation via disentangled representations. In *ECCV*, pages 35–51, 2018. 2

[34] Manyi Li and Hao Zhang. D^2IM-Net: Learning detail disentangled implicit fields from single images. In *CVPR*, pages 10246–10255, 2021. 2, 7

[35] Ruihui Li, Xianzhi Li, Ka-Hei Hui, and Chi-Wing Fu. SP-GAN: Sphere-guided 3D shape generation and manipulation. *ACM Transactions on Graphics (SIGGRAPH)*, 40(4):151:1–12, 2021. 2

[36] William E. Lorensen and Harvey E. Cline. Marching Cubes: A high resolution 3D surface construction algorithm. In *Proceedings of SIGGRAPH*, volume 21, pages 163–169, 1987. 2, 6

[37] Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, Vladimir G. Kim, and Yaron Lipman. Convolutional neural networks on surfaces via seamless toric covers. *ACM Transactions on Graphics (SIGGRAPH)*, 36(4):71:1–71:10, 2017. 1

[38] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D reconstruction in function space. In *CVPR*, pages 4460–4470, 2019. 2

[39] Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *ICCV*, pages 4743–4752, 2019. 1, 2

[40] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J. Mitra, and Leonidas J. Guibas. StructureNet: Hierarchical graph networks for 3D shape generation. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 38(6):242:1–242:19, 2019. 1, 2

[41] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *CVPR*, pages 909–918, 2019. 2

[42] Chengjie Niu, Jun Li, and Kai Xu. Im2Struct: Recovering 3D shape structure from a single RGB image. In *CVPR*, pages 4521–4529, 2018. 2

[43] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single RGB images via topology modification networks. In *ICCV*, pages 9964–9973, 2019. 1, 2, 7

[44] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019. 1, 2

[45] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural Parts: Learning expressive 3D shape abstractions with invertible neural networks. In *CVPR*, pages 3204–3215, 2021. 2, 4, 7

[46] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics Revisited: Learning 3D shape parsing beyond cuboids. In *CVPR*, pages 10344–10353, 2019. 2, 4, 7

[47] Jhony K. Pontes, Chen Kong, Sridha Sridharan, Simon Lucey, Anders Eriksson, and Clinton Fookes. Image2Mesh: A learning framework for single image 3D reconstruction. In *ACCV*, pages 365–381, 2018. 2

[48] Edoardo Remelli, Artem Lukoianov, Stephan R. Richter, Benoît Guillard, Timur Bagautdinov, Pierre Baque, and Pascal Fua. MeshSDF: Differentiable iso-surface extraction. In *NeurIPS*, pages 22468–22478, 2020. 2

[49] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. OctNet: Learning deep 3D representations at high resolutions. In *CVPR*, pages 3577–3586, 2017. 2

[50] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. OctNetFusion: Learning depth fusion from data. In *3DV*, pages 57–66, 2017. 2

[51] Yue Shi, Bingbing Ni, Jinxian Liu, Dingyi Rong, Ye Qian, and Wenjun Zhang. Geometric granularity aware pixel-to-mesh. In *ICCV*, pages 13097–13106, 2021. 2, 7

[52] Edward J. Smith, Scott Fujimoto, Adriana Romero, and David Meger. GEOMetrics: Exploiting geometric structure for graph-encoded objects. In *ICML*, pages 5866–5876, 2019. 1, 2

[53] Hao Su, Qixing Huang, Niloy J. Mitra, Yangyan Li, and Leonidas J. Guibas. Estimating image depth using shape collections. *ACM Transactions on Graphics (SIGGRAPH)*, 33(4):37:1–37:11, 2014. 2

[54] Chunyu Sun, Qianfang Zou, Xin Tong, and Yang Liu. Learning adaptive hierarchical cuboid abstractions of 3D shape collections. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 38(6):241:1–241:13, 2019. 2

[55] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural Geometric Level of Detail: Real-time rendering with implicit 3D shapes. In *CVPR*, pages 11358–11367, 2021. 1, 2

[56] Jiapeng Tang, Xiaoguang Han, Junyi Pan, Kui Jia, and Xin Tong. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single RGB images. In *CVPR*, pages 4541–4550, 2019. 1, 2

[57] Jiapeng Tang, Xiaoguang Han, Mingkui Tan, Xin Tong, and Kui Jia. SkeletonNet: A topology-preserving solution for learning mesh reconstruction of object surfaces from RGB images. *IEEE Transactions Pattern Analysis & Machine Intelligence*, 2021. to appear. 2

[58] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree Generating Networks: Efficient convolutional architectures for high-resolution 3D outputs. In *ICCV*, pages 2088–2096, 2017. 2

[59] Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta. Implicit mesh reconstruction from unannotated image collections. *arXiv preprint arXiv:2007.08504*, 2020. 2

[60] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, pages 2635–2643, 2017. 2, 4, 7

[61] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, pages 2626–2634, 2017. 1, 2

[62] Mikaela Angelina Uy, Vladimir G. Kim, Minhyuk Sung, Noam Aigerman, Siddhartha Chaudhuri, and Leonidas J. Guibas. Joint learning of 3D shape retrieval and deformation. In *CVPR*, pages 11713–11722, 2021. 1

[63] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *ECCV*, pages 52–67, 2018. 1, 2, 7

[64] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive O-CNN: A patch-based deep representation of 3D shapes. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 37(6):217:1–217:11, 2018. 1, 2

[65] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, William T. Freeman, and Joshua B. Tenenbaum. MarrNet: 3D shape reconstruction via 2.5D sketches. In *NeurIPS*, pages 540–550, 2017. 1, 2

[66] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T. Freeman, and Joshua B. Tenenbaum. Learning shape priors for single-view 3D completion and reconstruction. In *ECCV*, pages 646–662, 2018. 1, 2

[67] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. PQ-NET: A Generative Part Seq2Seq Network for 3D Shapes. In *CVPR*, pages 829–838, 2020. 1, 2

[68] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. SAGNet: Structure-aware Generative Network for 3D-shape modeling. *ACM Transactions on Graphics (SIGGRAPH)*, 38(4):91:1–91:14, 2019. 1, 2

[69] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. DISN: Deep implicit surface network for high-quality single-view 3D reconstruction. In *NeurIPS*, pages 490–500, 2019. 2

[70] Yifan Xu, Tianqi Fan, Yi Yuan, and Gurprit Singh. Ladybird: Quasi-Monte Carlo sampling for deep implicit field based 3D reconstruction with symmetry. In *ECCV*, pages 248–263, 2020. 2

[71] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective Transformer Nets: Learning single-view 3D object reconstruction without 3D supervision. In *NeurIPS*, pages 696–704, 2016. 1, 2

[72] Guandao Yang, Yin Cui, Serge Belongie, and Bharath Hariharan. Learning single-view 3D reconstruction with limited pose supervision. In *ECCV*, pages 86–101, 2018. 1, 2

[73] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. PointFlow: 3D point cloud generation with continuous normalizing flows. In *ICCV*, pages 4541–4550, 2019. 2, 4

[74] Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J. Guibas, and Lin Gao. DSM-Net: Disentangled structured mesh net for controllable generation of fine geometry. *arXiv preprint arXiv:2008.05440*, 2020. 2

[75] Wang Yifan, Noam Aigerman, Vladimir G. Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3D deformations. In *CVPR*, pages 75–83, 2020. 3

[76] Zerong Zheng, Tao Yu, Qionghai Dai, and Yebin Liu. Deep implicit templates for 3D shape representation. In *CVPR*, pages 1429–1439, 2021. 3

[77] Keyang Zhou, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Unsupervised shape and pose disentanglement for 3D meshes. In *ECCV*, pages 341–357, 2020. 2

[78] Rui Zhu, Hamed Kiani Galoogahi, Chaoyang Wang, and Simon Lucey. Rethinking Reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *ICCV*, pages 57–65, 2017. 2

[79] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3D-PRNN: Generating shape primitives with recurrent neural networks. In *ICCV*, pages 900–909, 2017. 2

[80] Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black. Lions and Tigers and Bears: Capturing non-rigid, 3D, articulated shape from images. In *CVPR*, pages 3955–3963, 2018. 2