A Survey on Joint Embedding of Text and Shape

Linlian jiang

School of Artificial Intelligence
Jilin University
jianglinlian.ll@gmail.com

Abstract

Joint learning of representations of 3D shapes and text is critical to our current work in progress. To my knowledge, joint embeddings of shapes and text are mostly used for i) text-to-shape retrieval and ii) text-to-shape generation. This survey focuses on the ways for joint embeddings of 3D shapes, we focus more on part-level understanding rather than object-level understanding. In brief, the ways can be roughly divided into two types, i) define the text-shape similarity matrix and ii) a view-based model, to learn cross-modal representations by joint reconstruction and prediction of view and word sequences. The former focuses on object-level, the latter on part-level. Although the level is not the same, the two methods still have many similarities and reference values. However, these methods are used for cross-modal retrieval, which means that they are not yet fully applicable to our project. This requires us to design a more creative approach when shape-text joint embedding is applied to our work. Thus, to implement modifications at the part level, we need to focus on structure awareness and how to explicitly represent the structural information of each model.

1. Introduction

Language allows people to communicate thoughts, feelings, and ideas. With the rapid development of artificial intelligence, the combination of language and other modes is also more extensive, among them, the joint embedding of natural language and 3D shape has attracted much attention. However, most joint embedding research revolves around 3D shape detection and generation. For example, according to describing a "round glass coffee table with four wooden legs", and having a matching colored 3D shape be retrieved or generated. On the other hand, learn the joint embedding of 3D shapes and text by matching parts from shapes to words from sentences. Joint embedding of text and shapes can facilitate 3D design. Most intuitively, the generation of initial shapes can be described by natural language, which means more information is provided for the handling of the 3D shape. According to the above research, we want to apply the method to text-driven point cloud completion by learning joint embedding of text and shapes on retrieval and generation.

The traditional point cloud completion method is based on certain prior information of the basic structure of the object, such as symmetry information or semantic information, etc. And repairs the missing point cloud through certain prior information. This kind of method can only deal with some missing point clouds with very low missing rates and obvious structural features. In recent years, some works have also tried to use deep learning to achieve point cloud completion, such as LGAN-AE [1], PCN [12], 3D-Capsule [2], etc. These works take incomplete point clouds as input and output complete point clouds, causing the network to pay too much attention to The overall characteristics of the object are ignored and the geometric information of the missing regions is ignored. Other paper also uses incomplete point cloud as input, only output the missing part of the point cloud, and better retain the individual characteristics of an object. These methods are all unimodal and only input shapes. We want to input the information of two modalities of text and shape to better complete the work of point cloud completion. Input natural language can be used to initialize shapes, and text-driven point cloud completion can become a new way of completion.

To achieve this goal, first of all, we have to figure out how the previous research solves the joint embedding of text and shape. According to my survey, there has been a lot of research on natural language and part-level matching, different from the previous method, the input of our project is the missing point cloud, and we need to find the keyword corresponding to the missing part in the text. In simple terms, we have to find which words in the text describe the missing part. At the same time, we need to pay attention to the part-level of the model, explicitly representing the structural information. In What follows, I will introduce the methods for joint embedding of text and shapes and structure-aware

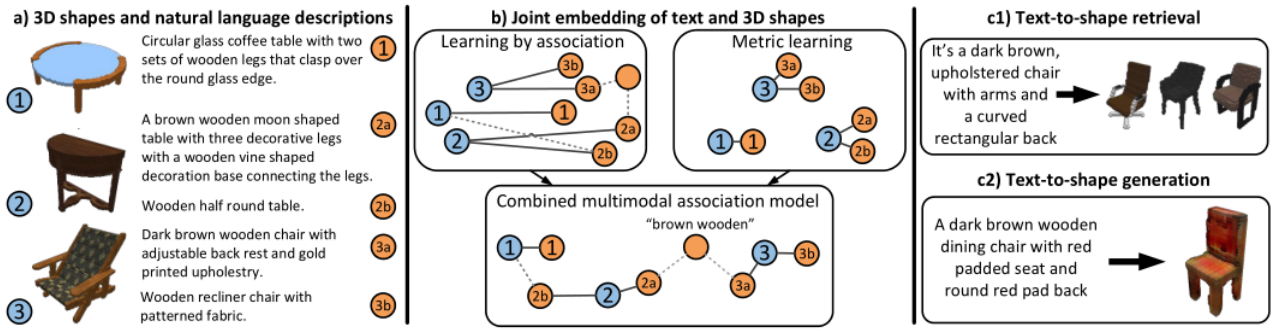

Figure 1. Two task in Text2Shape and the way of joint embedding.

point cloud generation. And the implications of these methods for our project.

2. Related Work

We review work in related areas such as object-level and part-level multi-model representation learning of shapes and text, deep learning of 3D point clouds, text-related matching tasks and explicitly modeling part semantics and shape structures.

2.1. Joint embedding of shapes and text on object-level

In the previous work, Chen et al.[4] introduces a novel 3D-Text cross-modal dataset by annotating each 3D shape from ShapeNet [3] with neural language description. In order to understand the inherent connections between text and 3D shapes, they employ CNN+RNN and 3D-CNN to extract features from freeform text and 3D voxelized shapes respectively. It uses full multi-modal loss to learn the joint embedding and calculate the similarity between both modal features. This kind of joint embedding does not need to segment the shape, it focuses on the object-level.

2.2. Joint embedding of shapes and text on part-level

However, due to the complexity of computational 3D convolutions, it is difficult to generalize this model to high resolution by focusing too much on the object level. In response to this situation, more and more research focuses on part-level, Han et al.[5] propose Y^2Seq2Seq, which is a view-based method, to learn cross-modal representations by joint reconstruction and prediction of view and word sequences. Although this method can extract texture information from multiple rendered views by CNN and acquire global shape representation by RNN, it ignores local information aggregation such as part-level features of 3D shapes, which proves to be useful for the 3D-Text task. In order to preserve the inherent 3D attributes better, and get more discriminative features, Tang et al.[11] proposes Part2Word, which method directly learns from point clouds sampled from shapes, introduce a cross-attention mechanism to learn the joint embedding of 3D shapes and text by matching parts from shapes to words from sentences. The focus of both papers is at the part-level. The task is to retrieve.

2.3. Structured Geometry Interpolation

We need to focus on 3D shapes along with associated part semantics and structure. We need to make the part editable, such as discrete changes in the composition. Mo et al. [6] propose StructureNet, this is a hierarchical graph network that produces a unified latent space to encode structured models with both continuous geometric and discrete structural variations. The parts of the chair, such as armrests, the back, and legs, besides, the height of the chair, these are editable. Shape parts can naturally be organized into n-ary hierarchies, each node is a part or part assembly. In PartNet [9] dataset, Most chairs decompose into backrest, seat, and base at the top hierarchy level. The n-ary tree in the shape representation can directly capture this hierarchy.

2.4. Part-based and Structure-aware Point Cloud Generation

Recent works such as StructureNet [6] and StructEdit [7] learn to generate and edit shapes explicitly following the pre-defined part hierarchy. And their work introduces structure-aware. We also need to explicitly model part semantics and shape structures. Mo et al.[8] proposes PT2PC, which formulated a new task of generating 3D point cloud shapes with geometric variations conditioned on structural shape descriptions. He constructed the "part tree" that conditionally generated multiple different point clouds that satisfies the structure. The symbolic part tree contains only part semantics and their relationships.

3. Methodologies

We will summarize the specific details of the above work in detail.

3.1. Joint Embedding

We focus on object-level and part-level joint embedding of text and shape. Although the processing methods at different levels are different, each method is worth learning from.

3.1.1 Object-level Joint Embedding

Chen et al.[4] does two tasks in text2shape: i) text-to-shape retrieval and ii) text-to-shape generation. As shown in Figure 1(a), firstly, generate several natural language descriptions for each 3D shape, in Figure 1(b), in addition to explicit text-shape links in solid lines, for models of the same category, it is very likely that the descriptors of one model can roughly match another one. For example, shape 1 and shape 2 belong to the same category. In addition to having a strong correspondence with text1, shape 1 also has a weak correspondence with the description text2b corresponding to shape 2. Because there are "wooden", "round" and "table" in the descriptor of text2b. We call this roughly consistent relationship an implicit semantic connection (dashed lines). This process extends learning by association and metric learning to jointly learn a text and 3D shape embedding that clusters similar shapes and descriptions, which strengthens links between similar instances in each modality. In Figure 1(c), the above approach will be applied to two tasks: text-to-shape retrieval (c1), which retrieves 3D shapes matching descriptions from a data set, and the challenging new task text-to-shape generation (c2), which generates new shapes from the text. As for cross-modal associ-

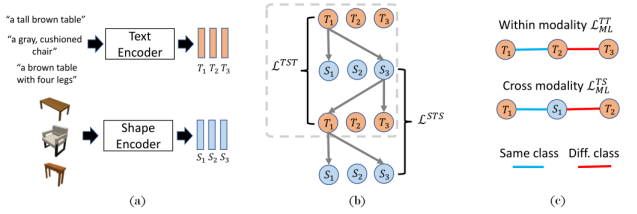

Figure 2. Overview of our joint representation learning approach.

ations, as shown in Figure 2(a), each shape and text description are passed through a shape encoder or a text encoder, producing shape embedding S and text embedding T. The text encoder adopts a CNN+RNN (GRU) structure and the shape encoder adopts a 3D-CNN, which similar to Reed et al.[10].

The core part of jointing embedding is to define the text-shape similarity matrix $M_{ij} = T_i \cdot S_j$, where T_i and S_j are the ith and jth rows of the corresponding matrices. Then convert this into the probability that description i is associated with shape j. After that, pass these through a softmax layer:

$$P_{ij}^{TS} = e^{M_{ij}} / \sum_{j'} e^{M_{ij'}}$$

As shown in Figure 2(b), first define the concept of text-shape-text round trips: in Figure 1(a), text 1 belongs to shape 1, it is an explicit text-shape link, "round table" appearing in text 2b is weakly related to text 1. The process of combining text1-shape1-text2b is text-shape-text round trips, and by using text-shape-text round (TST) trips, it establishes the implicit connections between separate text and shape instances that are semantically similar (e.g."round table" pulling shapes 1 and 2 together in Figure 1). Moreover, it extends the association learning approach with additional round trips and greatly improves the performance of our model for the multimodal problem. The round-trip probability for description i and shape j is $P_{ij}^{TST} = \left(P^{TS}P^{ST}\right)_{ij}$. For a given description i, the goal is to have P_{ij}^{TST} be uniform over the descriptions j which are similar to description i.

3.1.2 Part-level Joint Embedding

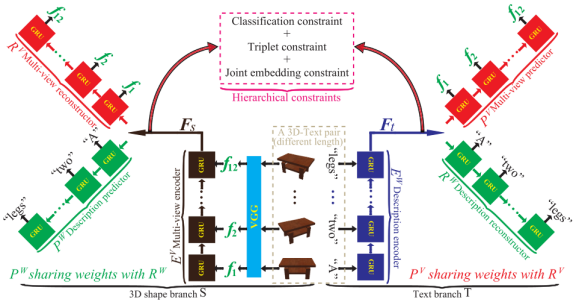

Figure 3. The framework of Y^{2}Seq2Seq consists of a 3D shape branch S and text branch T.

Han et al.[5] proposes a new method named Y^{2}Seq2Seq for the part-level joint embedding of shape and text. The framework of Y^{2}Seq2Seq is shown in Figure 3. It is a view-based model, to learn cross-modal representations by joint reconstruction and prediction of view and word sequences. The framework consists of a 3D shape branch S and text branch T. Each branch jointly reconstructs within the modality and predicts across modalities. The two branches are independent, and their work is in a sense mutually inverse. Briefly, 3D shape branch S is formed by a multi-view encoder E^V, a multi-view reconstructor R^V, and a description predictor P^W. E^V learns the feature F_s of shape s by aggregating all the N view features, where F_s is the hidden state at the N-th step of the RNN, then jointly reconstruct the view features in the sequence v and predict the word sequence w. This process predicts the description from the part, matching the word from the 3D shape.

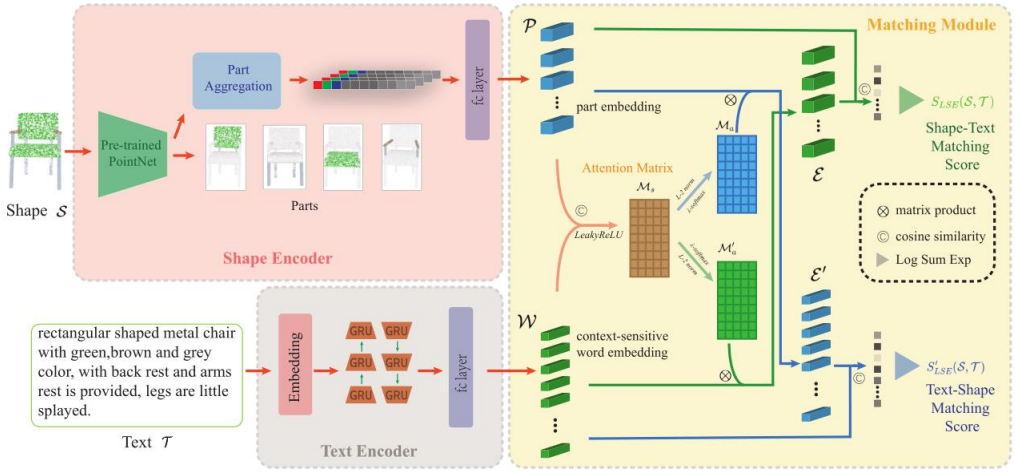

Figure 4. The network of Part2Word.

We will focus on text branch T. Similarly, text branch T is formed by a description encoder E^W, a description reconstructor R^W, and a multi-view predictor P^V. E^W learns the feature F_t of the description t of shape s by aggregating the embedding e_j of words w_j in sequence w. The feature F_t is represented by the hidden state at the M-th step of E^W. R^W and P^V jointly reconstruct the word sequence w and predict the view features in sequence v. This process predicts the part by sequence w. The loss of branch T is formed by the loss L_{W2W} from word to word and the loss L_{W2V} from word to view, as defined below:

$$
\begin{aligned}
&L_{\mathrm{T}} = \gamma L_{W2W} + \delta L_{W2V} \\
&L_{W2W} = -\sum_{j\in[1,M]} \mathrm{og}\, p\left(w_j \mid w_{<j}, \boldsymbol{w}\right), \\
&L_{W2V} = \frac{1}{N}\sum_{i\in[1,N]} \left\| \boldsymbol{f}_i'' - \boldsymbol{f}_i \right\|_2^2
\end{aligned}
$$

Another article introduces another approach to part-level joint embedding. Tang et al.[11] proposes Part2Word, which is a point-based multi-modal alignment network to solve the problem of joint embedding of shape and text. Introduce a cross-attention mechanism to learn the joint embedding of 3D shapes and text, the network is trained to match parts on point clouds to words in sentences.

The network structure is shown in Figure 4, the proposed network includes three modules: shape encoder S, text encoder T, and matching module. The role of the shape encoder S is to segment and extract the shape, and get $\mathcal{P} \in \{p_1, p_2, \ldots, p_k\}$ of the input shape S. The role of the text encoder T is to learn context sensitive embedding $\mathcal{W} \in \{w_1, w_2, \ldots, w_m\}$ by Bi-directional Gate Recurrent Unit (GRU). Shape-Text matching module matches the input 3D shape S and text T by the part embedding P and context-sensitive word embedding W. The key to the matching between P and W is to use an alignment-based matching module that uses cross attention to align parts with words to obtain similarity scores. The similarity score here is somewhat similar to the text-shape similarity matrix mentioned in text2shape. According to computing cosine similarity between P and W, we obtain the attention matrix M^s, use LeakyReLU to weaken the impact of negative values:

$$
\mathcal{M}_{ij}^s = \mathrm{LeakyReLU}\left(\frac{p_i^T w_j}{\|p_i\|\,\|w_j\|}, 0.1\right), i \in [1,k], j \in [1,m]
$$

Then normalize by part-wise L-2 normalization:

$$
\mathcal{M}_{i,j}^a = \frac{\mathcal{M}_{ij}^s}{\sqrt{\sum_{i=1}^k \left(\mathcal{M}_{ij}^s\right)^2}}
$$

After that, multiple the M^s and context-sensitive word embedding W to obtain the attention sentence embedding $\mathcal{E}_i = \sum_{j=1}^m \mathcal{M}_{ij}^a \mathcal{W}_j$ corresponding to each shape part. At the end, calculate the cosine similarity between P and ε, we can get the relationship between parts and sentences:

$$
R\left(p_i, e_i\right) = \frac{p_i^T e_i}{\|p_i\|\,\|e_i\|}, i \in [1,k]
$$

And the final Shape-Text similarity score is obtained through the LogSumExp pooling:

$$
S_{LSE}(S,T) = \log\left(\sum_{i=1}^k \exp\left(\lambda_2 R\left(p_i, e_i\right)\right)\right)^{(1/\lambda_2)}
$$

What inspires me is, whether it is object-level or part-level, joint embedding can be obtained by calculating the similarity between text and shape, which seems like a general approach.

3.2. Structured Geometry Interpolation

Every object has structural information. For the models in the PartNet [9], each model is composed of several

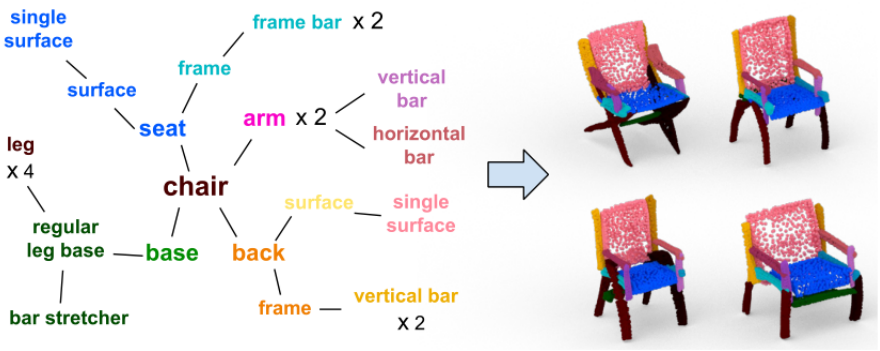

Figure 5. "part tree"-to-"point cloud"(PT2PC).

Figure 6. StructureNet.

parts. Mo proposes StructureNet [6], as shown in Figure 5, it is a hierarchical graph network that produces a unified latent space to encode structured models with both continuous geometric and discrete structural variations. It can add, remove or modify shape components and constituent structures for continuous deformation of parts, structural or discrete changes. For the two papers about part-level joint embedding I mentioned above, neither of them takes this structure-aware into account.

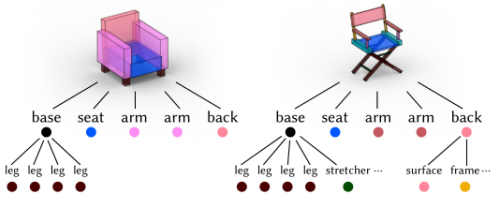

Figure 7. N-ary part hierarchies.

As shown in Figure 7, shape parts can naturally be organized into n-ary hierarchies. The top row shows oriented bounding boxes of leaf parts and the hierarchy is illustrated below. Hierarchy nodes have the same color as the corresponding part. It allows structure-aware interpolation between source and target shapes while displaying both topological and geometric variations. So that we can smartly modify a part or replace a part.

3.3. Part-based and Structure-aware Point Cloud Generation

Structure-aware is important for our work. And we need explicitly model part semantics and shape structures. Another paper by Mo: PT2PC [8] also inspires me. He proposes a conditional GAN "part tree"-to"point cloud" model that disentangles the structural and geometric factors.

As shown in Figure 5, the purpose is generating 3D point cloud shapes with geometric variations conditioned on structural shape descriptions. Obviously, as seen on the left of Figure 5, the input is only partially semantic and the part's relationships, rather than a sentence. The output generates a diverse 3D point cloud shape that meets the structural conditions, a combination of multiple parts. Briefly, generating multiple parts to fit existing structures, rather than focusing on the generation of a single part. All in all, the input and output are a little different from our current work.

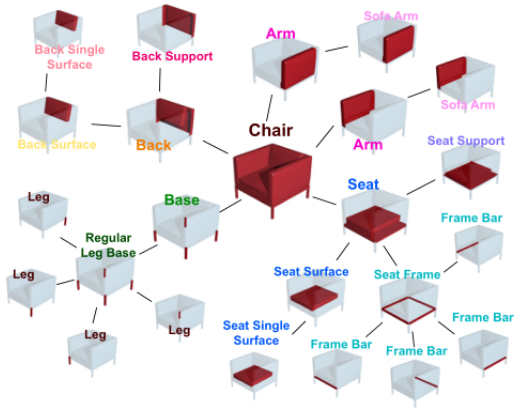

Figure 8. An example PartNet hierarchical part segmentation.

PT2PC is a conditional GAN that learns a mapping from a given symbolic part tree T and a random noise vector z to a 3D shape point cloud X composed of part point clouds for the leaf nodes of the conditional part tree.). Figure 8 shows the ground-truth part hierarchy of the chair. A symbolic part tree T is defined as $T = (T_V, T_E)$, where $\mathcal{T}_V = \left\{P^j \mid P^j = \left(\mathbf{s}^j, \mathbf{d}^j\right)\right\}_j$ represents a set of part instances and T_E represents an directed edge set of the part parent-children relationships $\mathcal{T}_E = \{(j,k)\}$. In T_V, each part instance P^j is composed a semantic label s^j and a part instance identifier d^j.

Part-tree conditioned generator is composed of three network modules: a symbolic part tree encoder G_{enc}, a part tree feature decoder G_{dec} and a part point cloud decoder G_{pc}. First, the symbolic part tree encoder G_{enc} embeds the nodes of T into compact features t^j hierarchically from the leaf nodes to the root node for every node P^j. Then, taking in both the random variable z and the hierarchy of symbolic part features $\left\{\mathbf{t}^j\right\}_j$, G_{dec} hierarchically decodes the part features in the top-down manner, produces part feature f^j for every leaf node $P^j \in \mathcal{T}_{\text{leaf}}$. Finally, G_{pc} transforms the leaf node features into 3D part point clouds in the shape space. At each step of the part feature decoding, the parent node needs to know the global structural context in order to propagate coherent signals to all of its children.

4. Conclusions

This survey revolves around ongoing work on our work. Firstly, introduce the joint embedding of text and shapes at the object-level and part-level. Although different in level, it seems that the way of joint embedding is somewhat similar in some places, and this is worth learning from. Then we want to explicitly represent the structural information of each model. Our work hopes to subtly modify or replace parts, which means allowing structure-aware interpolation between source and target shapes, and displaying topological and geometric variations. Based on this, PT2PC provides a more specific method to us, shapes that conform to the structure can be generated through the part semantics and relationships provided by part-tree, and provides ideas to deal with individual parts.

In summary, we can refer to the methods mentioned in the above papers to design the text encoder and shape encoder we need. On the other hand, add structure-aware to the shape, so as to implement modifications to the parts.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 1

[2] Ayesha Ahmad, Burak Kakillioglu, and Senem Velipasalar. 3d capsule networks for object classification from 3d model data. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 2225–2229. IEEE, 2018. 1

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[4] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Asian conference on computer vision*, pages 100–116. Springer, 2018. 2, 3

[5] Zhizhong Han, Mingyang Shang, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Y2seq2seq: Cross-modal representation learning for 3d shape and text by joint reconstruction and prediction of view and word sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 126–133, 2019. 2, 3

[6] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019. 2, 5

[7] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J Mitra, and Leonidas J Guibas. Structedit: Learning structural shape variations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8859–8868, 2020. 2

[8] Kaichun Mo, He Wang, Xinchen Yan, and Leonidas Guibas. Pt2pc: Learning to generate 3d point cloud shapes from part tree conditions. In *European Conference on Computer Vision*, pages 683–701. Springer, 2020. 2, 5

[9] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 2, 4

[10] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016. 3

[11] Chuan Tang, Xi Yang, Bojian Wu, Zhizhong Han, and Yi Chang. Part2word: Learning joint embedding of point clouds and text by matching parts to words. *arXiv preprint arXiv:2107.01872*, 2021. 2, 4

[12] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 1