ShapeScaffolder: Structure-Aware 3D Shape Generation from Text

Xi Tian
University of Bath
Bath, UK
xt275@bath.ac.uk

Yong-Liang Yang
University of Bath
Bath, UK
y.yang@cs.bath.ac.uk

Qi Wu
University of Adelaide
Adelaide, Australia
qi.wu01@adelaide.edu.au

Abstract

We present ShapeScaffolder, a structure-based neural network for generating colored 3D shapes based on text input. The approach, similar to providing scaffolds as internal structural supports and adding more details to them, aims to capture finer text-shape connections and improve the quality of generated shapes. Traditional text-to-shape methods often generate 3D shapes as a whole. However, humans tend to understand both shape and text as being structure-based. For example, a table is interpreted as being composed of legs, a seat, and a back; similarly, texts possess inherent linguistic structures that can be analyzed as dependency graphs, depicting the relationships between entities within the text. We believe structure-aware shape generation can bring finer text-shape connections and improve shape generation quality. However, the lack of explicit shape structure and the high freedom of text structure make cross-modality learning challenging. To address these challenges, we first build the structured shape implicit fields in an unsupervised manner. We then propose the part-level attention mechanism between shape parts and textual graph nodes to align the two modalities at the structural level. Finally, we employ a shape refiner to add further detail to the predicted structure, yielding the final results. Extensive experimentation demonstrates that our approaches outperform state-of-the-art methods in terms of both shape fidelity and shape-text matching. Our methods also allow for part-level manipulation and improved part-level completeness.

1. Introduction

The advance of 3D representation learning and generative models has sparked increasing interest in 3D shape generation. However, text-guided shape generation remains a challenging task. Many current approaches generate the 3D shape as a whole when no part information is utilized, while others treat text as a simple collection of words in order to provide finer guidance. Both shape and text, however, possess inherent internal structures that can be leveraged to improve the alignment between the two modalities.

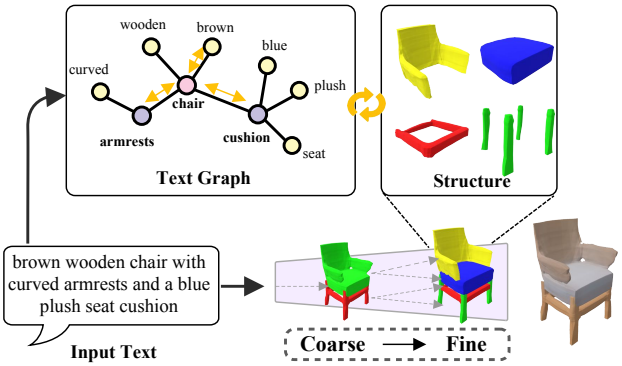

Figure 1. Text to shape generation with structure awareness. Raw description generates the initial coarse shape with structures. Further correspondence is built between shape parts and text graph nodes for fine-grained refinement.

Humans are born with a sense of physical understanding of the world [3]. Studies have shown that the human visual system perceives objects as hierarchical arrangements of parts [4], comprising high-level structural properties and low-level details. In addition, humans use language to describe objects, including their appearance, structure, and functions. It has been discovered that language, shaped by the human brain over time, has evolved to have hierarchical linguistic structures in order to convey more complex meanings [10]. This suggests that structure-based reasoning is present in both the visual and language modalities. While this is intuitive and effortless for humans, it is a challenge for intelligent algorithms to understand the structural connection between these two modalities.

In this research, we aim to augment shape generation from text with an awareness of structure in both shape and language modalities. This would allow for shape generation to be guided not only by high-level descriptions, such as a sentence but also refined with part-level guidance from semantic text entities. This process is similar to how humans create visual artworks: sketching an idea's gist and then progressively refining the details. However, there are

several challenges to overcome. One challenge is that the shape structures are unknown unless annotated segmentation labels are provided. Even with annotation, where the process is time-consuming and labor-intensive, the definition of "parts" in shapes is subjective and ambiguous at multiple levels. Another challenge is that text descriptions of shapes are diverse and complex, making it difficult to optimize learning. In addition, for colored shapes, the alignment between structure and color representation must also be taken into consideration.

To address the challenges above, we propose a two-stage approach. First, we leverage an unsupervised hierarchical shape structure given part hierarchy is a prevalent and effective approach for representing shape structures, as demonstrated in previous research [22, 28]. To achieve this, we introduce a decomposition model to progressively decompose the shape into parts at multiple levels of granularity, from coarse to fine, such that each level can reconstruct the entire shape. Additionally, we learn color reconstruction at the part level. Fig. 1 illustrates the process. While the structures and parts identified in the first stage hold significance within the shape domain, they do not have a corresponding relationship with the text. In the second step, we establish a connection between the shape structure and the text structure. We utilize linguistic dependency graphs, which can be used to represent the structural elements present in the text, as attention signals to form the structure-level connection. Through these two steps, we are able to improve the generation results at the part level by building upon the established structure.

In our experimentation on the ShapeNet dataset [7], we have demonstrated the efficacy of utilizing a structure-aware representation in improving the correspondence between text and shape. As a result of the part-aware refinement, we observe a significant improvement in the quality of generated shapes, particularly in terms of their constituent parts. Additionally, our approach attains state-of-the-art results regarding several commonly used metrics such as Intersection over Union (IOU), Accuracy, and Inception Score (IS). Furthermore, through further exploration of the latent shape structure space, it is demonstrated that our method has the capability of generating previously unseen shapes.

The main contributions of this work are: (1) An unsupervised hierarchical decoder for building shape structural implicit field. (2) A structure-aware text-to-shape generation method that improves part-level correspondence and refinement. (3) Comprehensive experiments that demonstrate the effectiveness of our method.

2. Related Work

Text Guided 3D Shape Generation. Deep learning has led to a surge in works exploring text-guided 3D shape or scene generation [6, 8, 1, 25, 27, 13]. Recently, to achieve faithful results, some works tend to adopt pre-trained CLIP models [35, 17, 26] or diffusion-based methods [33, 23] for large-scale learning on open datasets. Our work, however, focuses on the classic ShapeNet dataset [7], examining the generation of structure-aware, colored shapes from text and comparing our results to previous baselines.

Generating 3D shapes directly from text remains a challenge, particularly when attempting to concurrently generate colors. [8] proposes a method that learns a joint text-shape embedding space and then uses a generative adversarial network (GAN) [14] to generate colored voxels from text. However, this approach has not been successful in terms of shape resolution, color quality, and cross-modality consistency. To address these issues, [25] proposes a method that uses an implicit 3D representation with word-level attention to improve control over generated shape structures and colors. [27] and [13] both adopt a discrete autoencoder to capture patch-based shape priors and subsequently employ a transformer for autoregressive generation. However, it is worth noting that these methods do not take into account color information, which is an essential attribute of object appearance. Also, all these methods have treated the shape as a whole, ignoring the internal structure within the shape. Additionally, text modeling has been limited to language-level features, such as sentence and word features, rather than considering the structure of the text for structure-level matching with the shape domain.

3D Structure-Aware Representation and Parsing. Although structure-based shape generation with text guidance has not been previously explored, there is a significant amount of research on learning structure-aware 3D shape representation or parsing 3D shapes into parts. For supervised methods, [22] proposes the use of a symmetry hierarchy [39] for hierarchical shape structure representation, while StructureNet [28] represents shapes using a graph neural network that considers both primitives and hierarchies. Similarly, [16] utilizes a binary parsing tree to represent the 3D structure of a cable-stayed bridge. Recently, semantic-based shape decomposition or parsing methods have been proposed in [19, 18], which use learned operations to obtain grammar-level shape parts. Unsupervised methods such as [37, 31] use convolutional neural networks (CNNs) to generate primitive shapes, *e.g.*, cuboids or superquadrics, which abstract the input shape. BAE-NET [9] attempts to co-segment shapes into parts using a branched autoencoder architecture that disentangles common features for segmentation. However, these methods only predict the parts without considering their dependencies in the shape structure. [30] recovers the 3D objects into a hierarchy of parts directly from an RGB image using CNN, while the parts are primitive and lack details. RIM-NET [29] infers

hierarchical structures of 3D shapes using recursive implicit fields. It decomposes an input shape into two parts at each level, resulting in a binary tree hierarchy, where each level of the tree corresponds to an assembly of shape parts represented as implicit functions. However, the method does not consider color information and only processes one shape at a time during training. Our work uses a hierarchical tree structure for shape representation, but also includes color information for part-level implicit representation. In addition, our model is able to learn a shared latent space for a set of shapes.

Text as Textual Graph Guidance. Text-to-image (T2I) generation has seen significant progress in recent years, with a focus on both generating realistic images and increasing consistency between texts and images. Some approaches use attention-based methods [40] for fine-grained alignment, while others leverage shared structures in images and texts. Scene graphs, which are derived from texts and contain expressive structural relationships between entities, have been widely used to reduce the cross-modality gap between texts and images for tasks such as text-image matching [24], fine-grained image generation [2], and image manipulation [12]. Given that both shapes and texts contain structural elements, we opt to employ textual graphs as additional guidance in conjunction with the generation of structural parts.

3. Method

3.1. Overview

The task is to generate 3D shapes with colors based on the given text description T. The predicted object is denoted as $O \in \mathbb{R}^{N\times 4}$, comprising the shape occupancy values $S \in \mathbb{R}^{N\times 1}$ and the color RGB values $C \in \mathbb{R}^{N\times 3}$, where N is the number of sample points.

The overview of our method is illustrated in Fig. 2. Our framework consists of a text encoder E_T, textual graph encoder E_G, hierarchical structure-aware shape decoder D_T, and shape encoder E_S. In contrast to general methods that generate the whole shape at once, our approach, called ShapeScaffolder, aims to generate the 3D shape by considering its internal structures, enabling refinement at the part level and improvement of text-shape correspondence. To achieve this, we propose to capture the part-level information from both shape and text. For the shape, we introduce the concept of *structural implicit field* in the shape decoder to capture the underlying structure and relationships of shape parts and colors. As shown in Fig. 2 (middle right), the structure latent fields capture the semantic shape parts for a chair, including arm, cushion, legs, *etc*. For the text, in addition to encoding it as a global feature such as a sentence vector, we also propose using a linguistic textual graph parsed from the text that includes explicit entities with relations. This enables the structure latent fields to interact with textual entities at a finer level, resulting in more direct and explicit correspondence between the text description, such as "curved armrests" and the corresponding arm part of the generated shape. However, it is non-trivial to train a hierarchical structure-aware generator conditioned on text. We approach this task by dividing the training process into two stages: (A) for learning the structural implicit field using an autoencoder, and (B) for generating shapes guided by text, as summarized in Fig. 2. Stage B can be further divided into text encoding (B-1) and shape generation (B-2). The following describes each stage in detail.

3.2. Shape-Specific Pre-training

To capture the internal semantic structures of shapes and facilitate shape decoding conditioned on text, we employ a pre-training technique using a shape-specific hierarchical decoder D_S and a shape encoder E_S, forming an autoencoder (see Fig. 2 (A)). The encoder E_S, which is a convolutional neural network (CNN), takes the volume of an object O as input and produces latent shape and color features to be used by the decoder. The decoder then creates an implicit field by partitioning the shape ($\bar{f}_s$) and color ($\bar{f}_c$) latents at each hierarchical level, reconstructing the latent features into parts and assembling them into the final object. An example of a successfully learned structural implicit field is shown in Fig. 2 (right), where semantic shape parts can be reconstructed from their corresponding latent positions.

Hierarchical Shape Decoder The Hierarchical Decoder consists of two modules at each level: the Structure Divider (SD) and the Part Generator (PG). Given a pair of shape/color latent features ($f_s^{l,i} \in \mathbb{R}^d$, $f_c^{l,i} \in \mathbb{R}^d$) at level l and position $i (1 \le i \le 2^l)$, the SD first divides the shape latent into two child shape features and then generates the color latent for each child conditioned on the corresponding child shape latent. The SD constructs the latent space for both shape and color, as depicted by the grey and yellow squares in Fig. 2. The PG then generates shape parts one by one from its shape/color latents at a given set of point coordinates $\mathbf{p} = \{(x, y, z) \mid x, y, z \in \mathbb{R}\}$. For each point p, the input is the concatenation of p with the latent value, and SD outputs one occupancy probability for the shape channel and three values (RGB) for the color channel. The resulting generated shape O_l at level l is a composite of generated parts, where a point is occupied by a part with the highest occupancy probability, provided that this probability exceeds a pre-defined threshold. The color of each point is determined based on the corresponding color values of the occupying part. Both SD and PG are implemented using multi-layer perception (MLPs). The SD is implemented using a two-layer MLP for partitioning the shape latent space

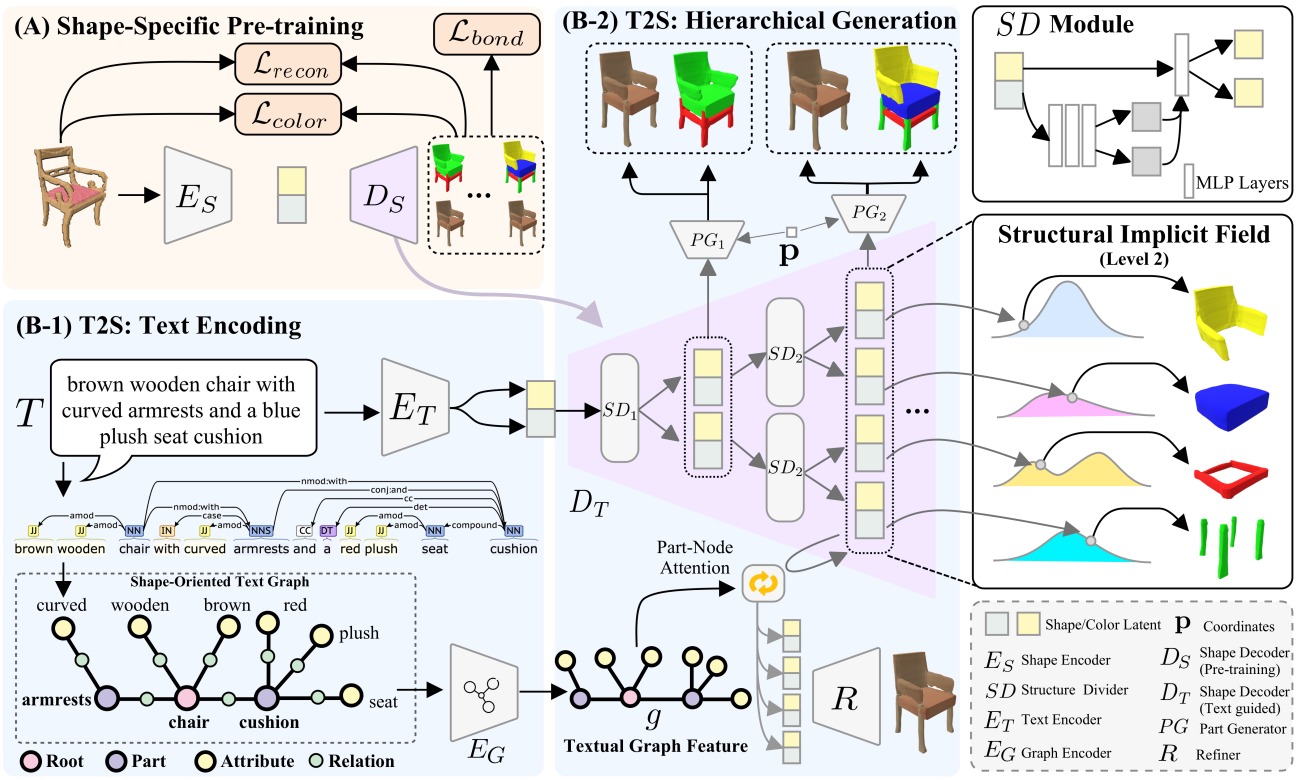

Figure 2. Method overview. The proposed method includes two phases: shape autoencoder pre-training (A) and text-to-shape (T2S) phase including (B-1) and (B-2). Phase (A) trains the shape-specific decoder using shape/color reconstruction loss and hierarchical bond loss. Phase (B-1) utilizes the text encoder E_T to convert text into global-level shape ($\bar{t}_s$) and color ($\bar{t}_c$) representations, as well as local-level features in the form of a textual graph (g). In Phase (B-2), the shape decoder generates the object hierarchically using the global input from Phase (B-1), with further use of a part-node attention module and refiner (R) to connect part latents and graph node features for more precise guidance.

and a one-layer MLP for partitioning the color latent space. The PG has three layers for the shape latents and two layers for the color latents. Both SD and PG use LeakyReLU as intermediate activation layers and Sigmoid as the final one. Note SD and PG have shared weights within each level of the hierarchy tree, but different weights between the levels.

Reconstruction Loss The total reconstruction loss of shape and color is calculated by summing the individual reconstruction losses at each level using Mean Squared Error (MSE), as shown below:

$$L_{recon} = \lambda_{recon} \sum_{l=1}^{L} \left(S_{gt} - S_l\right)^2, \tag{1}$$

$$L_{color} = \lambda_{color} \sum_{l=1}^{L} \left(C_{gt} - C_l\right)^2, \tag{2}$$

where S_{gt} and C_{gt} represent the ground truth for shape and color, respectively. S_l and C_l are the assembled part's occupancy and color value at level l.

Hierarchical Bond Loss The generated shape parts, following the hierarchical tree structure, need to be constrained by the relationships between parent and child parts in terms of geometric composition. To ensure this, a bond loss is calculated between level l and the next level $l+1$. The bond loss is defined as the error between the occupancy value of each parent part and the maximum of the occupancy values of its two derived parts:

$$L^l_{bond} = \sum_{i=1}^{2^l} (S^l_i - \max(S^{l+1}_{2i-1}, S^{l+1}_{2i}))^2, \tag{3}$$

where S^l_i is the occupancy value of the parent part at position i and level l, and S^{l+1}_{2i-1} and S^{l+1}_{2i} are the occupancy values of the two derived child parts. The total bond loss is computed by summing the bond loss across all levels when $l < L$: $L_{bond} = \lambda_{bond} \sum_{l=1}^{l<L} L^l_{bond}$.

Regularization Loss To reduce the complexity of the network and avoid overfitting, we introduce regularization loss L_{reg} at the last layer of each Structure Divider at each level.

This loss is calculated as the sum of the absolute values of all weight parameters, multiplied by a small regularization parameter λ_{reg}. The regularization loss is given by: $L_{reg} = \lambda_{reg} \sum_{i=1}^{n} |w_i|$, where w_i is the i-th weight parameter in the layer, and n is the total number of weight parameters. Our experiments show that without using regularization loss, the generated shapes may contain parts that are over-segmented or lack reasonable shape.

Comparison with Existing Method Our method differs from previous approaches that utilize (hierarchical) decoders for unsupervised shape part learning (such as in [9, 29]) in several ways. Firstly, our hierarchical decoder aims to learn semantic shape part-level implicit field rather than simply segmenting the shape into more parts. For example, when generating a chair with four legs, our approach considers the legs as a whole unit rather than segmenting them individually, as we typically describe the legs as a whole rather than as individual parts. To balance the network's fitting ability with its complexity, we utilize a regularization loss. Secondly, we also construct hierarchical color latent fields, which are often overlooked in other methods, to represent colored shapes.

3.3. Text-Guided Shape Generation

Fig. 2 (B) shows the text-guided shape generation network, comprising three modules: text encoder E_T, textual graph encoder E_G, and hierarchical shape decoder D_T. The D_T module is similar to the pre-trained shape-specific decoder D_S (as described in Section 3.2), with the addition of a new part-node attention module and a refiner. This attention module allows the network to learn the local connections between shape parts/colors and textual graph entities, followed by a refiner module R for final results generation. D_T module is initialized from the pre-trained D_S to leverage its predefined structural implicit fields and facilitate the generation process in a hierarchical manner. We provide a detailed description of each part in the following sections.

Text Global Encoding We utilize a pre-trained BERT model [11] as the text encoder base. The encoded [CLS] token feature is used to extract global-level latent features $\bar{t}_s$ and $\bar{t}_c$ for shape and color, respectively, which serve as the input for the hierarchical shape decoder D. Most existing approaches employ word-level features w as fine-grained guidance in text-conditioned image or 3D shape generation tasks [40, 25]. However, we argue that the use of structured textual representation, such as the textual graph, would be more appropriate in aligning with shape structures. As a result, we opt to use the word features to initialize the graph node representation, as described below.

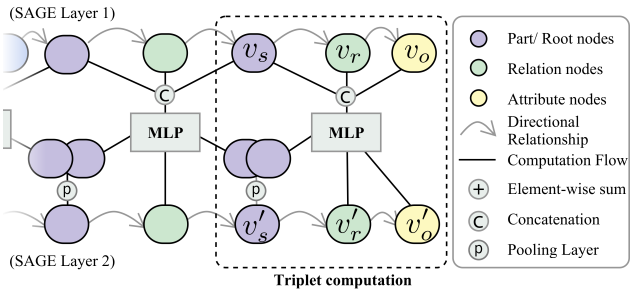

Figure 3. The dashed box shows a triplet (v_s, v_r, v_o) in the graph, denoting the subjective, relation, and objective nodes. The three nodes are first concatenated and then jointly processed using MLP, which outputs a new value for each node. The final updated state v'_s of v_s is the result of pooling the corresponding values from all triplets in which the node v_s is involved.

Shape-Oriented Text Graph We use a unique text graph representation designed specifically for shapes. The graph nodes are classified into *root*, *part*, and *attribute* categories, which represent the class of the shape, the parts, and the attribute for root/part nodes, respectively. We use a special type node *relation* to connect these nodes, forming the edges. For example, the text "brown wooden chair with curved armrests" would be parsed as (chair: brown, wooden), (armrests: curved), and their relation triplet (chair, with, armrests) in the (subjective, relation, objective) convention. To construct graphs from the original text, we first use the spaCy library [15] to parse the text's semantic dependencies. These raw dependencies contain many linguistic terms and do not correspond well with real-world structures. To obtain more clear and shape-oriented graph formation, we introduce a custom parser targeting 3D shapes based on [36]. To ensure that the graph remains connected, we connect all *part* nodes with the *root* node using the *with* relation if no linguistic relation is found. The lower left of Fig. 2 illustrates these two forms of text. Finally, we use the resulting relation triplets for training. Further details can be found in the supplementary materials.

Spatial-Aware Graph Encoding One challenge in using a shape-oriented graph, where entity or attribute nodes are connected through relation nodes, is the lack of a canonical grid for embedding the graph since nodes in such a graph exist in a non-Euclidean space. Existing methods based on Graph Convolution Networks (GCN) [21] typically rely on an adjacency matrix to represent the relations between nodes, which reduces their ability to represent spatial relations. To address the issue, we designed a novel Spatial-Aware Graph Encoder (SAGE) that differentiably encodes relation nodes in order to capture the structural relations between nodes. Specifically, we represent each graph triplet g in the form (v_s, v_r, v_o), where v_s, v_o, and v_r represent

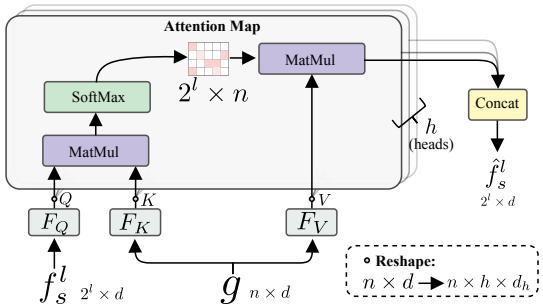

Figure 4. The Part-Node attention. The attention map is obtained by the dot-product between the features from the intermediate part latent feature (f_s^l) at level l and the graph node features g.

the values of subjective, objective, and relation nodes in a relationship, respectively. This makes the graph directional through the order of the nodes between the relation node, like *chair has arms*. As shown in Fig. 3, the computational layer in SAGE jointly encodes the nodes and their relation in each triplet, followed by a pooling layer that fuses the features of each node involved in more than one triplet.

Part-Node Attention We introduce the Part-Node Attention modules for learning the relationship between intermediate parts/colors with textual graph nodes. These modules are based on the multi-head attention mechanism presented in [38]. For level l, given the intermediate shape latents $f_s^l \in \mathbb{R}^{2^l \times d}$ as *query*, and the encoded graph node features, $g \in \mathbb{R}^{n \times d}$ (n is node number), as *key* and *value*, the calculation inside an attention head is:

$$\hat{f}_s^l = \text{softmax}\left(\frac{QK}{\sqrt{d_h}}\right)V, \tag{4}$$

where $Q = F_Q(f_s^l)$, $K = F_K(g)$, $V = F_V(g)$, and F_Q, F_K, and F_V are functions implemented using MLPs. d is the feature dimension for g and f_s^l, while d_h represents the dimension used within an attention head. The SoftMax operation is applied across the graph nodes, resulting in a weighted aggregation of the graph node features as the new intermediate part latents, $\hat{f}_s^l$. This process is visualized in Fig. 4. Also, attention is applied between the paired color features f_c^l with the graph nodes using a separate module. In practice, we use the attention module at level 2 of the hierarchical decoder due to the generated semantically detailed shape components aligned with graph nodes.

Joint Refiner The hierarchical bond loss, introduced in Section 3.2, serves as the foundation for structure-aware generation by constraining the composition of shape parts to follow a hierarchical tree structure. However, we posit that this also negatively impacts the generated results, as demonstrated in the perception-distortion tradeoff [5]. To mitigate this issue, we introduce a refiner R in one level that uses the level latents jointly to generate the final shape. The refiner is also an implicit shape decoder with six layers of MLPs. The final shape output is accompanied by an additional refiner reconstruction loss L_R but without the need to consider the hierarchical bond, which allows for the incorporation of additional detail in the final shape.

Training Strategy and Losses Training text-guided hierarchical shape generation directly presents significant challenges considering the big discrepancy between text and shape modalities. To address this, we propose a three-step strategy to ensure stable and effective training. First, we collect the shape and color latents from the shape encoder E_S as guidance and train the text global feature encoder E_T alone. The loss is using MSE: $L_{latent} = \lambda^s_{latent}(\bar{t}_s - \bar{f}_s)^2 + \lambda^c_{latent}(\bar{t}_c - \bar{f}_c)^2$, where $\bar{t}_s$ and $\bar{t}_c$ are encoded by text encoder while $\bar{f}_s$ and $\bar{f}_c$ are encoded by shape encoder. Second, we train the decoder with attention modules, freezing the decoder during the first ten epochs to allow for warmed-up training. Finally, we train the entire network together. The total loss then includes the aforementioned shape/color reconstruction losses and the bond loss in shape-specific pre-training (see Sec. 3.2), and the refiner loss.

4. Experiments

This section introduces our experimental design and implementation, followed by a comprehensive analysis of the results from various aspects and an ablation study. Supplementary material provides additional results and analysis.

4.1. Experimental Settings and Implementation

We conduct experiments using the text-shape dataset [8], which is derived from the 3D shape dataset, ShapeNet [7], and augmented with textual annotations. The dataset comprises approximately 15,000 shapes and 75,000 textual descriptions, encompassing the table and chair shape classes. We also adhere to the same training and testing splits as previously established.

We implement our framework in PyTorch [32]. We set $N = 4096$ as the sampled points. We pre-train the shape-specific decoder for 1000 epochs and the text-guided model for another 500 epochs. The learning rate is constant $1e^{-4}$ using Adam [20] for all phases. The loss weights are set as follows: $\lambda_{recon} = 10.0$, $\lambda_{color} = 1.0$, $\lambda_{bond} = 1.0$, $\lambda_{reg} = 1e^{-4}$, $\lambda^s_{latent} = 10.0$ and $\lambda^c_{latent} = 1.0$. Detailed training strategy can be found in Sec. 3.3. Additionally, as our model outputs shapes at different levels, we select the results from level 2, as it is at this level that most shapes' parts have been parsed and refined. Level 3 is used as a spare level for monitoring no further parts are parsed.

	IoU (↑)	IS (↑)	EMD (↓)	Acc (↑)
Text2Shape [8]	9.64	1.96	0.4443	97.37
T2S-Implicit [25]	12.21	1.97	0.2071	97.48
ours	**13.65**	**1.98**	**0.1767**	**97.5**

Table 1. Quantitative comparison with state-of-the-art. ↑ and ↓ indicate higher or lower is better, respectively.

4.2. Text-Guided Shape Generation

Comparison with the State-of-the-art We compare our method to two existing text-conditioned colored shape generation methods, Text2Shape [8] and T2S-Implicit [25]. We employ the same quantitative metrics for evaluation, including (1) Mean intersection-over-union (IoU) for shape occupancy measurement, regardless of color; (2) Earth Mover's Distance (EMD) for color distribution evaluation; (3) Inception Score (IS) to quantify the realism of the results; and (4) Accuracy (Acc) computed by a pre-trained shape classifier to indicate whether the predicted class matches the ground truth. To ensure fairness in comparison, we convert the generated shapes to volumes of the same size as in [8, 25] before calculating the quantitative values. The comparison results are presented in Table 1, where the numbers of the existing methods are directly taken from the respective papers. Our method exhibits superior performance across all metrics. The qualitative result presented in Fig. 5 provides further evidence of the effectiveness of our approach. As can be observed, our method achieves a higher degree of correspondence between the predicted shape structure and the text description, as demonstrated in Row 1 of the figure with the example of "wood armrests". Additionally, our approach exhibits an increased level of completeness in part generation, as evidenced by the absence of incomplete parts or extra artifacts, which are clearly labeled with red circles in the results generated by the existing methods.

Part-level Refinement As T2S-Implicit [25] also employs an implicit decoder similar to our approach, we conduct a comparison with it to highlight the distinctions, particularly with regard to part-level details. The results are presented in high resolution in Fig. 6. We observed that [25] tends to struggle with smaller or complex parts, such as the legs of a sofa or the wheels of an office chair. In contrast, our method is capable of interpreting these parts explicitly, resulting in more realistic part-level generation. Another advantage of our approach is that the generation of a single part does not affect the other parts, which is a common issue in previous methods. For example, generating "silver legs" may result in the bottom of the sofa appearing slightly silver in the connection area, creating a noticeable color artifact.

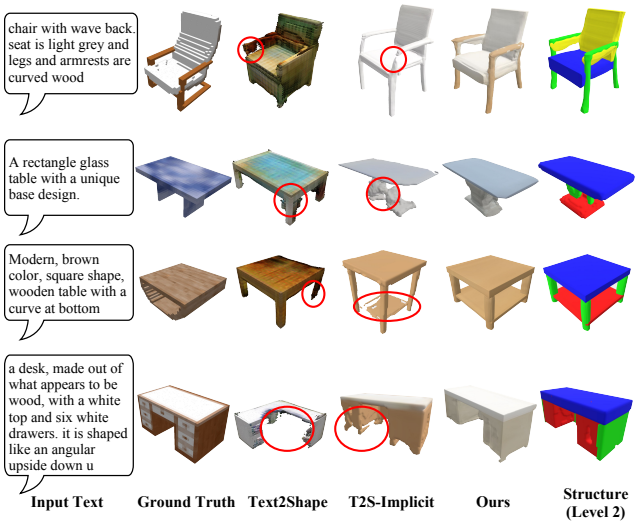

Figure 5. Visualization of the results of our method in comparison to Text2Shape [8] and T2S-Implicit [25]. Red circles indicate incomplete parts or extra artifacts.

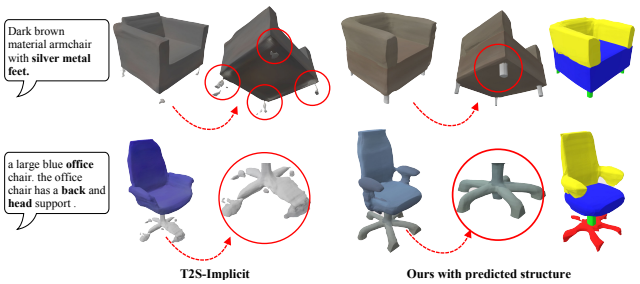

Figure 6. Comparison of part-level refinement.

Part-level Manipulation The ability to learn structure-based 3D shape representations and align them with textual descriptions enables precise and convenient manipulation of shapes. In the process of generating a shape from text, our method allows for manipulation at various levels, including modification of the part-level structure, attributes, or colors. The results of this manipulation are illustrated in Fig. 7. The changes made in the text appear to reflect the changes in part structure (a), part attribute (c), and part-related color refinement (b), suggesting a possible consistency between the constituent parts of the shape and the corresponding text.

4.3. Ablation Studies

We conduct extensive ablation studies to validate the effectiveness of the proposed hierarchical structure-based representation, graph-part attention, and joint refiner. We also use Frechet Inception Distance (FID) and CLIP [34] to evaluate the quality of the generated shapes and the similarity between the generated shapes and their corresponding text descriptions, respectively. More detailed information on the metrics can be found in the supplementary material. The re-

Method	Level 1			Level 2			Level 3		
	IOU↑	FID↓	CLIP↑	IOU↑	FID↓	CLIP↑	IOU↑	FID↓	CLIP↑
HierD [base]	11.20	17.47	45.43	11.07	17.52	45.36	10.85	17.60	43.92
FlatD	10.38	19.55	40.72	-	-	-	-	-	-
HierD + WordAttn (L2)	11.13	17.42	44.88	11.66	16.46	47.43	10.72	17.69	46.10
HierD + GraphAttn (L2)	11.29	17.36	45.01	12.63	16.31	50.64	**12.41**	16.70	50.01
HierD + GraphAttn (L1)	**11.54**	**16.86**	**52.75**	11.30	17.09	51.40	11.27	17.12	50.56
HierD + GraphAttn (L2) + Refiner(L2) [Full]	11.19	17.38	47.27	**13.65**	**15.81**	**54.89**	12.22	**16.34**	52.20

Table 2. Results of the ablation study on output levels. The base model, HierD, is the proposed hierarchical decoder using sentence-level guidance only. L1/L2 indicates component application at Level 1/Level 2. ↑/↓ denote better higher/lower results, respectively.

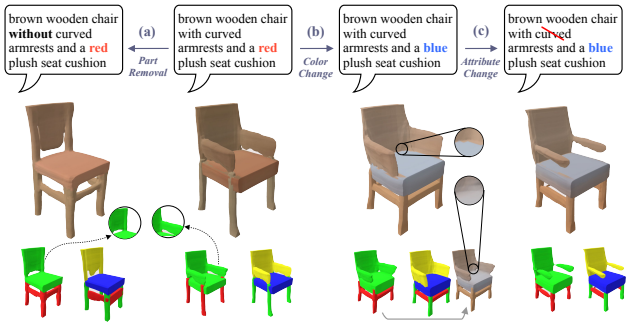

Figure 7. Manipulation of text results in corresponding changes at the part level. The second row shows the predicted structure from Level 1 to 2. (a) illustrates the removal of a part, (b) shows a change in part color, and (c) depicts a change in the part attribute.

sults of our ablation studies are presented in Table 2. We use the Hierarchical Decoder with sentence-level guidance (HierD) as a base model, and then add various modules such as word-level attention (WordAttn), our proposed graph-part attention (GraphAttn), and the joint refiner to different levels of the hierarchy (L1/L2 for Level 1 and 2). We present the results for each level for an in-depth analysis of the impact of the hierarchy level on the performance of the model.

Hierarchical Representation We compare HierD to a flattened decoder (FlatD) which directly outputs 8 parts at Level 1. Utilizing the same settings as HierD, FlatD demonstrates inferior performance. Our qualitative analysis reveals that FlatD is more prone to overfitting, resulting in the generation of fewer parts that are overly complex geometrically, instead of generating finer parts.

Graph-based Attention We have observed the clear advantages of utilizing graph-based attention over word-based attention. Specifically, when applied to Level 2, the use of GraphAttn results in superior performance across all metrics. Furthermore, the results from Level 3 also demonstrate improved performance. This can be attributed to the semantic alignment of the entity nodes in the graph with the intermediate parts. However, when applied to Level 1, the use of GraphAttn does not yield similar results as in Level 2. We speculate that this discrepancy may be due to the coarser nature of the parts at Level 1, which hinders the ability to establish effective connections between shape parts and textual graph entities.

Joint Refiner The utilization of a refiner as the final shape output in our method is driven by the observation of the perception-distortion tradeoff [5]. As the level increases, we have noted a slight decline in performance as compared to previous levels. For instance, the IOU in Level 2 is consistently lower than that in Level 1. We posit that the hierarchical bond loss employed in our approach results in a stronger structural connection, but may impede the generation of detailed features on top of the structure. The incorporation of a refiner, on the other hand, has resulted in a notable improvement in all metrics.

5. Conclusion, limitation, and future work

The ShapeScaffolder is a method for generating shapes guided by text, with the understanding that both shape and text possess internal structures that can be aligned to improve the outcome. We represent shapes as hierarchical structures and model text as scene graphs. Our part-node attention mechanism allows us to learn correspondences between shape and text at various hierarchical levels, which has been shown to effectively establish connections between shape parts/colors and linguistic entities. The experiments have demonstrated the superiority of our approach in both shape generation and text-shape/color consistency.

Several limitations should be acknowledged. Firstly, the current text-to-shape dataset is limited to chairs and tables, so further research is needed to explore the generalization of our method to other shape classes. Secondly, the results may be influenced by the parts priority problem, where incorrect occupancy state of parts junctions may result in unreasonable visualization. Therefore, further incorporation of prior interpretations of shapes should be considered in future studies. Lastly, it would be beneficial to explore scene generation that takes structure into account, with a focus on the composition of different shapes as an outer structure.

Acknowledgements. This work is supported by RCUK grant CAMERA (EP/M023281/1, EP/T022523/1), Centre for Augmented Reasoning (CAR) at the Australian Institute for Machine Learning, and a gift from Adobe.

References

[1] Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. Shapeglot: Learning language for shape differentiation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8938–8947, 2019. 2

[2] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4561–4569, 2019. 3

[3] Renée Baillargeon. Infants' physical world. *Current directions in psychological science*, 13(3):89–94, 2004. 1

[4] Irving Biederman. Human image understanding: Recent research and a theory. *Computer vision, graphics, and image processing*, 32(1):29–73, 1985. 1

[5] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6228–6237, 2018. 6, 8

[6] Angel Chang, Manolis Savva, and Christopher D Manning. Learning spatial knowledge for text to 3d scene generation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 2028–2038, 2014. 2

[7] 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, 6

[8] 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, 6, 7

[9] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. Bae-net: Branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8490–8499, 2019. 2, 5

[10] Morten H Christiansen and Nick Chater. Language as shaped by the brain. *Behavioral and brain sciences*, 31(5):489–509, 2008. 1

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5

[12] Helisa Dhamo, Azade Farshad, Iro Laina, Nassir Navab, Gregory D Hager, Federico Tombari, and Christian Rupprecht. Semantic image manipulation using scene graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5213–5222, 2020. 3

[13] Rao Fu, Xiao Zhan, Yiwen Chen, Daniel Ritchie, and Srinath Sridhar. Shapecrafter: A recursive text-conditioned 3d shape generation model. *arXiv preprint arXiv:2207.09446*, 2022. 2

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, 2014. 2

[15] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. 5

[16] Fangqiao Hu, Jin Zhao, Yong Huang, and Hui Li. Learning structural graph layouts and 3d shapes for long span bridges 3d reconstruction. *arXiv preprint arXiv:1907.03387*, 2019. 2

[17] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022. 2

[18] R. Kenny Jones, Aalia Habib, Rana Hanocka, and Daniel Ritchie. The neurally-guided shape parser: Grammar-based labeling of 3d shape regions with approximate inference. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[19] R Kenny Jones, Aalia Habib, and Daniel Ritchie. Shred: 3d shape region decomposition with learned local operations. *arXiv preprint arXiv:2206.03480*, 2022. 2

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 6

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 5

[22] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 2

[23] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 2

[24] Chunxiao Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang. Graph structured network for image-text matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10921–10930, 2020. 3

[25] Zhengzhe Liu, Yi Wang, Xiaojuan Qi, and Chi-Wing Fu. Towards implicit text-guided 3d shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17896–17906, 2022. 2, 5, 7

[26] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 13492–13502, 2022. 2

[27] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315, 2022. 2

[28] 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

[29] Chengjie Niu, Manyi Li, Kai Xu, and Hao Zhang. Rim-net: Recursive implicit fields for unsupervised learning of hierarchical shape structures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11779–11788, 2022. 2, 5

[30] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1060–1070, 2020. 2

[31] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10344–10353, 2019. 2

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[33] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 7

[35] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18603–18613, 2022. 2

[36] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136, 2011. 5

[37] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017. 2

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 6

[39] Yanzhen Wang, Kai Xu, Jun Li, Hao Zhang, Ariel Shamir, Ligang Liu, Zhiquan Cheng, and Yueshan Xiong. Symmetry hierarchy of man-made objects. In *Computer graphics forum*, volume 30, pages 287–296. Wiley Online Library, 2011. 2

[40] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018. 3, 5