ShapeScaffolder: Structure-Aware 3D Shape Generation from Text (Supplementary Material)

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

1. Introduction

We include the following in the supplementary material:

- Graph modeling (Sec. 2)
- Additional results (Sec. 3)

2. Graph modeling

Raw Textual Dependency. The raw textual dependency is obtained utilizing the SpaCy library [3]. The initial text is initially parsed into sentences, in the event that there are multiple sentences in a description. For each sentence, the dependency graph is parsed. An illustration of this result can be found in Figure 1. However, the raw dependency has limitations for direct usage, as the parsed entities may exist across multiple sentences, and there are linguistic terms that are not essential. To address these limitations, we employ an improved shape-oriented graph representation, which allows for the creation of a concise yet informative structure representation.

Shape-Oriented Graph Representation In previous work, such as [6], parsing general text into entities and relations has been explored. However, for the specific task of parsing 3D shapes, these methods were found to be inadequate due to a lack of shape-related glossaries and the desired final structure. To address this issue, we have designed a new parser that employs a multi-step approach. Firstly, we conduct an analysis of all text words to construct glossary lists. Next, we identify the relevant *nodes* present in the lists and establish rules to identify their *modifiers* as attributes. By analyzing the linguistic relationships between words based on the raw text dependency, we further develop rules to determine the relationships between nodes. Finally, we construct the final graph by merging or connecting the identified nodes based on their established relations. It should be noted that a single node may occur multiple times, and in the case of a single shape, these repeated nodes typically refer to a single entity. Additionally, we also consider lemmas as nodes, rather than the original words, for example, "leg" as the lemma for "legs".

In the initial step of our method, we prepare glossary lists for various categories of words, including *shape roots*, *shape parts*, *relations*, *adjectives* and *common words*. These lists are constructed by considering words with a frequency greater than 10. It should be noted that there may be shared words across the different lists, as some words can have multiple functions, such as serving as both a noun and an adjective. Specifically, the *roots* category represents the central node in a graph, which is the main class of the shape, such as "chair", "table", "stool", *etc*. The *shape parts* category includes words such as "arm", "back", "top", *etc*. The *relations* typically includes verbs, prepositions, or phrases, such as "have", "made up of", "with", *etc*. The *common words* category includes words that are typically used to describe the functions or applications of the shape, such as "family" or "kitchen". The second step is non-trivial, as the linguistic relationship is not direct in some cases. For example, for the text "the table is round and has 3 legs. the table is rotating", the "rotating" word is a Verb gerund but should be considered as an attribute.

3. Additional results

Metric Details and Comparison In the ablation study, we employ the Frechet Inception Distance (FID) and CLIP metrics to evaluate the performance of our method. To calculate the FID, we first train a classification task using features encoded by a 3D encoder [7], and we follow the computation process established in 2D generative networks [2]. For the CLIP metric, we first collect 18 rendered images for every 20° rotation for each generated 3D shape. We then use a pre-trained CLIP model [5] to compute the similarity between the text and each rendered 2D image. The final score is obtained by averaging the results of all the rendered images, with a further normalization as described in [1]. Based on the results of the ablation study, our method achieved a higher CLIP score of 54.89 compared to 37.58

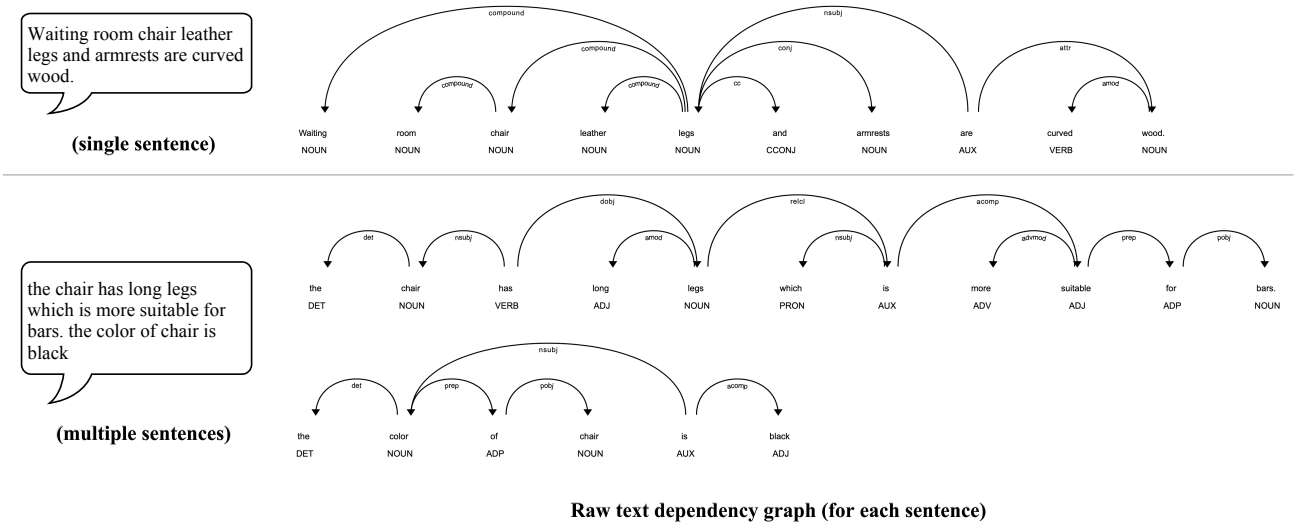

Figure 1. The raw text dependency graph examples. A piece of text will be parsed to more than one graph if multiple sentences are found.

obtained by T2S-Implicit [4]. This demonstrates the superior performance of our approach in the given evaluation metrics.

High-Res Results More high-resolution results are shown in Fig. 2.

Hierarchical latent field visualization. Fig. 3 shows the hierarchical latent field visualization which depicts the composition relationships between parts at different hierarchical levels.

Ablation Study Qualitative Visualization Fig. 4 gives the qualitative results of the ablation study. We show three aspects including the (A) structural decoder types, (B) local attention types, and (C) the use of refiner.

Latent Space Exploration Upon deeper examination of the structural latent space through the randomized composition of parts, we discovered that the learned structure latent space is capable of creating objects that were not present in the original dataset which only contained chairs and tables (see Fig. 5). Although there are no corresponding texts, we believe this has potential for future applications such as user-guided generation where a user can select and combine parts to create novel shapes.

Limitations In some cases, the generation process is limited due to the conflicting priorities of parts. As shown in Fig. 6, errors can happen at the junctions that result in unrealistic artifacts.

Failure Results Through the examination of the generated shapes, we discovered that our methods may encounter challenges in certain scenarios. These include: (1) instances where the text is overly complex, and (2) instances where the shape being generated is present in low frequency within the dataset (such as "a table with x-shaped legs"). These types of failures have been observed in previous methods as well, as reported in [4]. Examples of these failures are depicted in Fig. 7.

References

[1] 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. 1

[2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 1

[3] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. 1

[4] 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

[5] 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. 1

[6] 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*

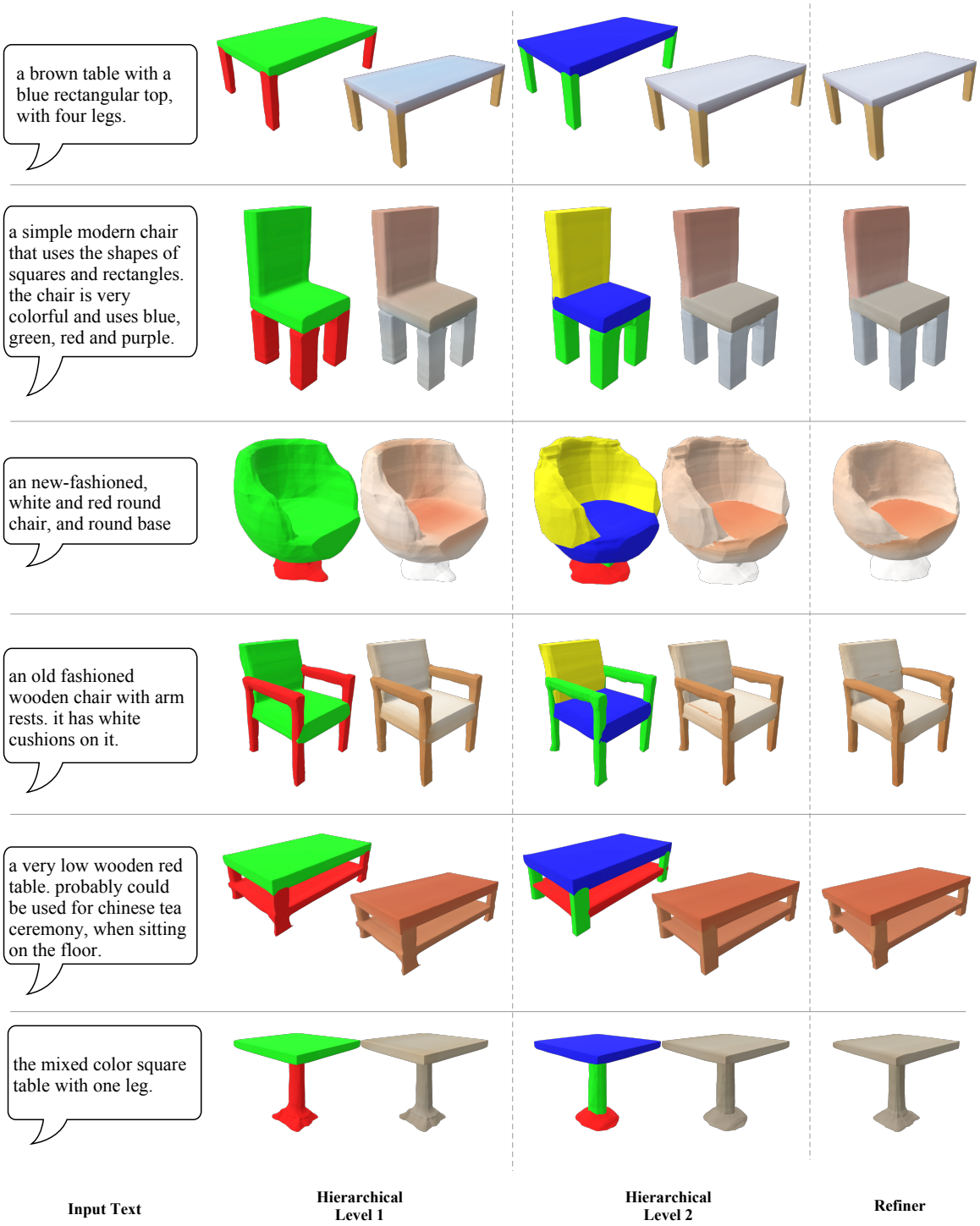

Figure 2. Additional results (structure and object) generated by our method. Best view by zooming in.

conference on machine learning (ICML-11), pages 129–136, 2011. 1

[7] Udaranga Wickramasinghe, Edoardo Remelli, Graham Knott, and Pascal Fua. Voxel2mesh: 3d mesh model generation from volumetric data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 299–308. Springer, 2020. 1

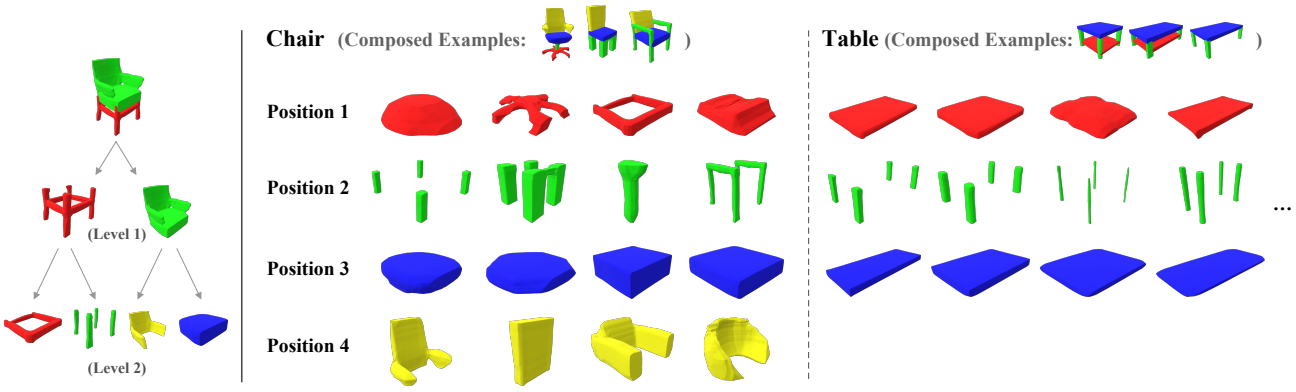

Figure 3. Hierarchical latent field visualization. (A) shows the composition relationship of parts across hierarchical levels. (B) presents a list of generated parts from different positions of level 2. The results reveal that a position can produce parts with similar features, regardless of whether the object is a chair or a table (such as legs). Position 4 for the table is typically empty as tables do not have an upper part, unlike chairs which have a back.

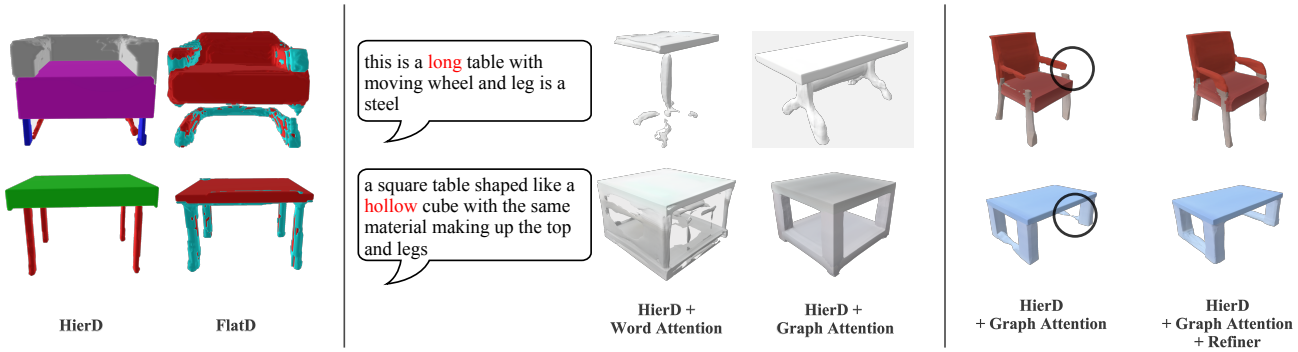

Figure 4. Qualitative visualization for ablation study. (A) compares the shape decoder using hierarchical (HierD) and flat architecture (FlatD). FlatD which directly outputs 8 parts is easy to overfit, and only 2 part positions are used to represent a shape, resulting in irregular parts. (B) compares the performance of word-based and graph-based attention modules, with the latter providing clearer and more accurate guidance. (C) shows the improvement obtained by using a Refiner to provide missing details.

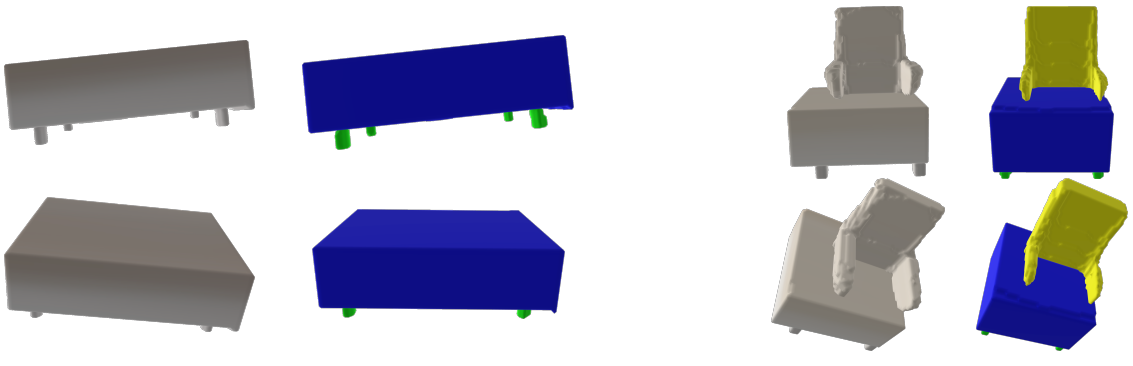

Figure 5. Exploring the latent space of structures through the randomized combination of part latents at the second hierarchical level. This provides the potential to generate novel, unseen shapes such as beds, that extend beyond the original classes of chairs and tables.

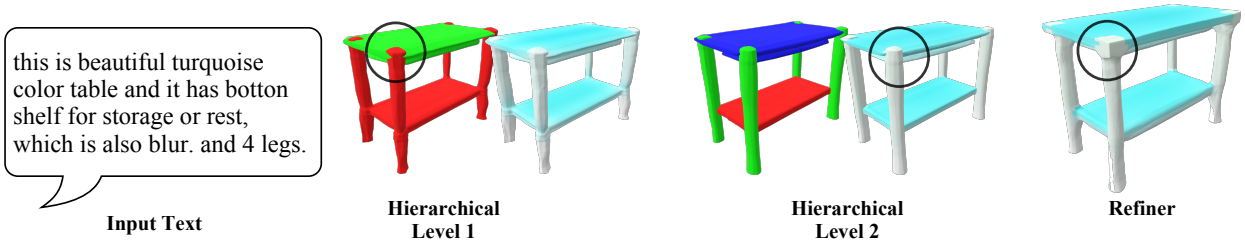

Figure 6. Limitations in generation due to conflicting priorities of parts. Errors at the junction of a tabletop and legs can lead to unrealistic artifacts, such as legs protruding slightly from the tabletop.

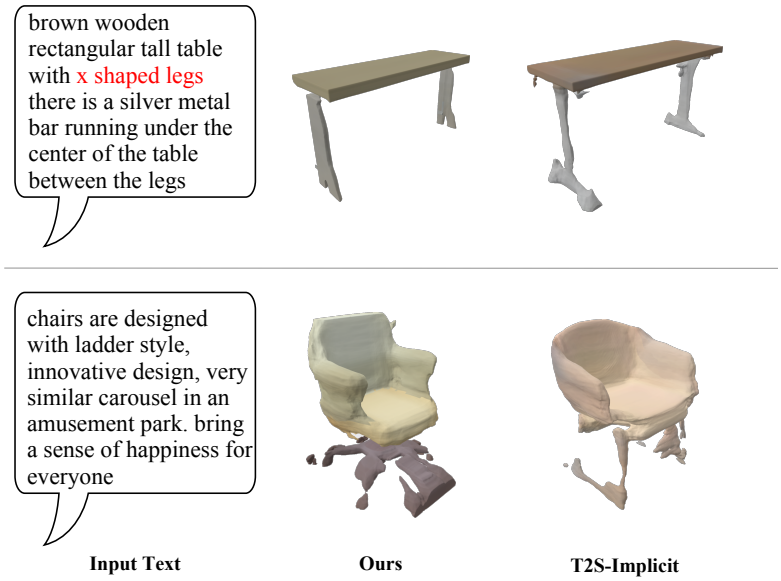

Figure 7. Failure results compared with T2S-Implicit [4].