Supplementary Material for TAPS3D: Text-Guided 3D Textured Shape Generation from Pseudo Supervision

Jiacheng Wei[1*] Hao Wang[1*] Jiashi Feng[2] Guosheng Lin[1†] Kim-Hui Yap[1]
[1]Nanyang Technological University, Singapore [2]ByteDance
{jiacheng002@e., hao005@e., gslin@, ekhyap@}ntu.edu.sg, jshfeng@bytedance.com

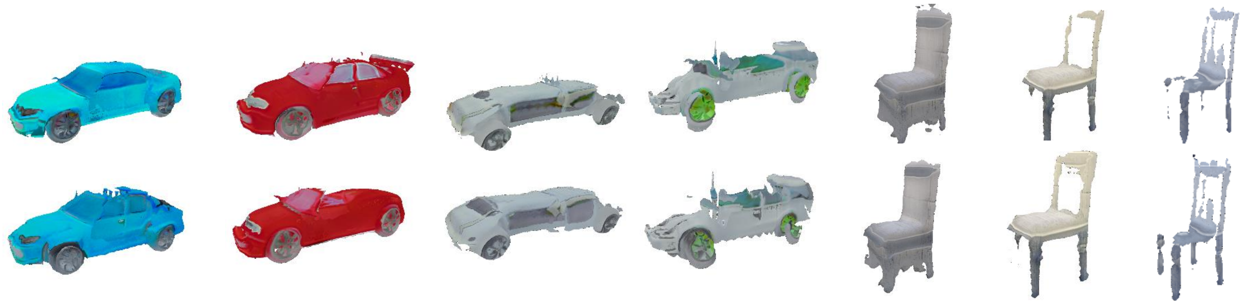

Figure A. We show the generated results of the model trained from scratch, where we update all model parameters including the mapping networks, generator, and discriminators. We observe that the model easily collapses during the training.

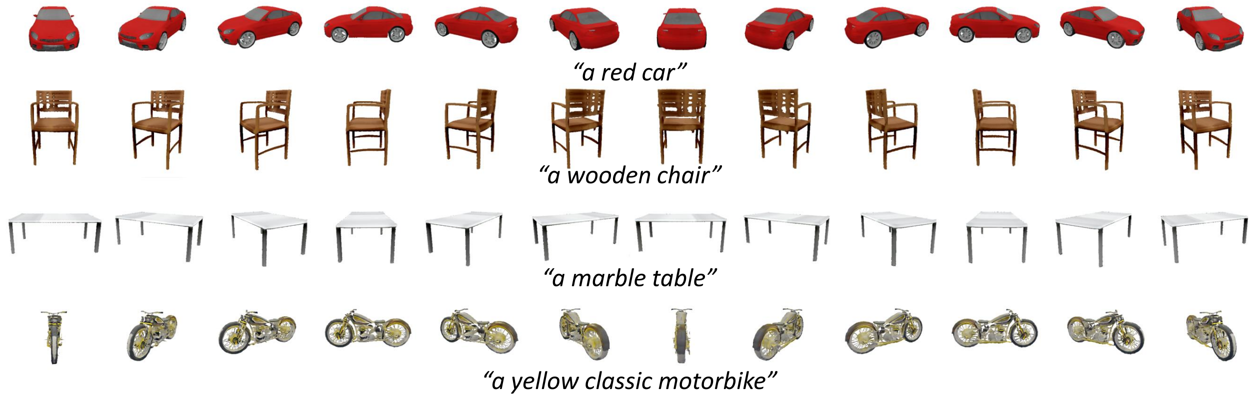

Figure B. We show the multi-view results of generated objects via neural rendering. Our method produces multi-view consistent results related to the input texts.

1. Comparison of different training strategies

In Fig. A, we show the results of training the entire model from scratch with both the CLIP loss and the GAN loss. We observe that the model is easy to collapse when we update the whole network parameters. More loss functions involved during the training phase makes it harder to converge for the generator and discriminators, resulting in coarse and incomplete generation outputs. While updating the mapping network only retains the pretrained network capability, which is learned during the unconditional training.

2. Details of the mapping networks

We follow [1, 3] to implement the text-conditioned mapping networks, in which we take the random vectors $\mathbf{z} \in \mathbb{R}^{512}$ and CLIP text features $E_t(t) \in \mathbb{R}^{512}$ as input. We first adopt 2-layer MLPs f_t to map $E_t(t)$ to another space so that we can concatenate $E_t(t)$ with the random vectors $\mathbf{z}$.

[*] Equal contribution. Work done during an internship at Bytedance.
[†] Corresponding author.

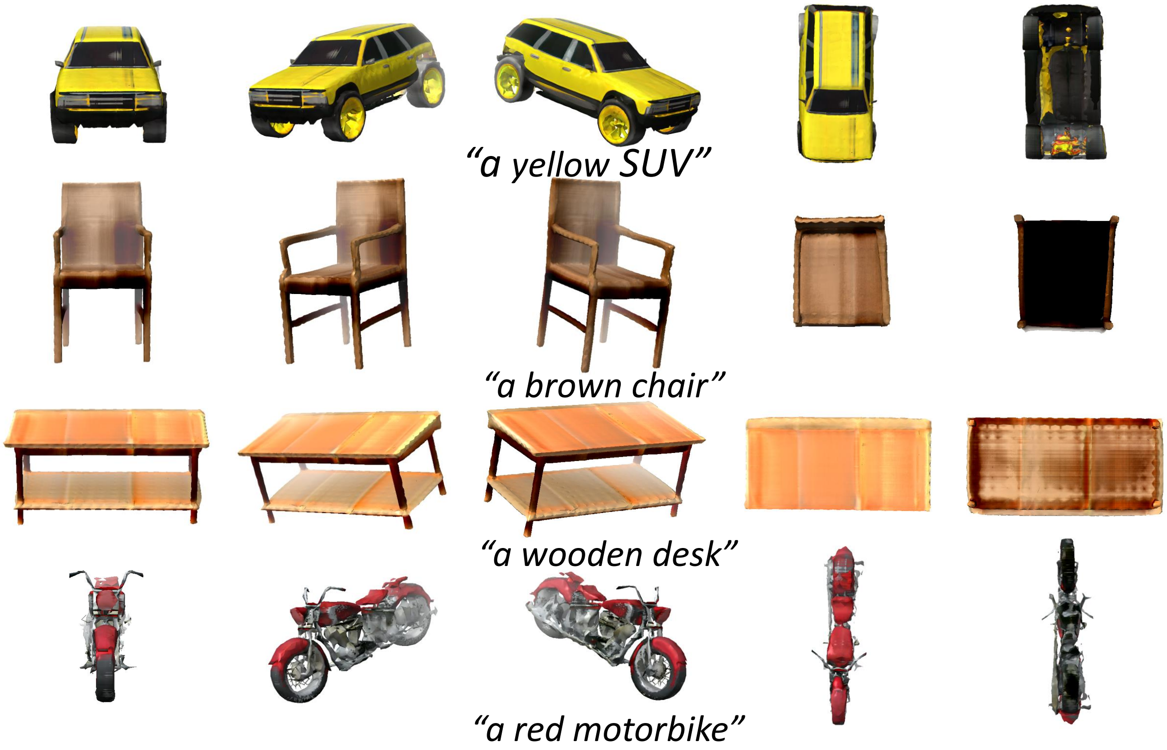

Figure C. We show the multi-view visualizations of the generated textured meshes, given the input texts. We also present the top views and bottom views of the generated meshes. These mesh results are visualized with ChimeraX [2].

Then we produce latent codes $\mathbf{w} \in \mathbb{R}^{512}$ from the concatenation of $\langle f_t(E_t(t)), \mathbf{z} \rangle$ by 8-layers MLPs f_{map}, which can be denoted as $\mathbf{w} = f_{map}(\mathbf{z}, f_t(E_t(t)))$. Please note each layer of the MLPs is a fully-connected layer having 512 hidden dimensions and a leaky-ReLU activation. We use the same architecture for the geometry and texture mapping networks.

3. More qualitative results

Multi-view consistency. In Fig. B, we show that our proposed method generates multi-view consistent images from neural rendering with respect to the input text prompts. In Fig. C, we visualize the generated textured meshes in different views. The multi-view visualizations illustrate that our model can generate 3D shapes that are consistent with the input texts on each side of the objects.

Interpolation Results. We produce interpolation results between two input texts in Fig. D. In each row, we use the same sampled random noise $\mathbf{z}$ with different input texts to generate the source and target latent codes $\mathbf{w}_1$ and $\mathbf{w}_2$. Then, we produce the interpolation results between $\mathbf{w}_1$ and $\mathbf{w}_2$. We show that our learned mapping networks generate smooth and meaningful latent codes from the text input to guide the shape generation.

References

[1] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *arXiv preprint arXiv:2209.11163*, 2022. 1

[2] Thomas D Goddard, Conrad C Huang, Elaine C Meng, Eric F Pettersen, Gregory S Couch, John H Morris, and Thomas E Ferrin. Ucsf chimerax: Meeting modern challenges in visualization and analysis. *Protein Science*, 27(1):14–25, 2018. 2

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1

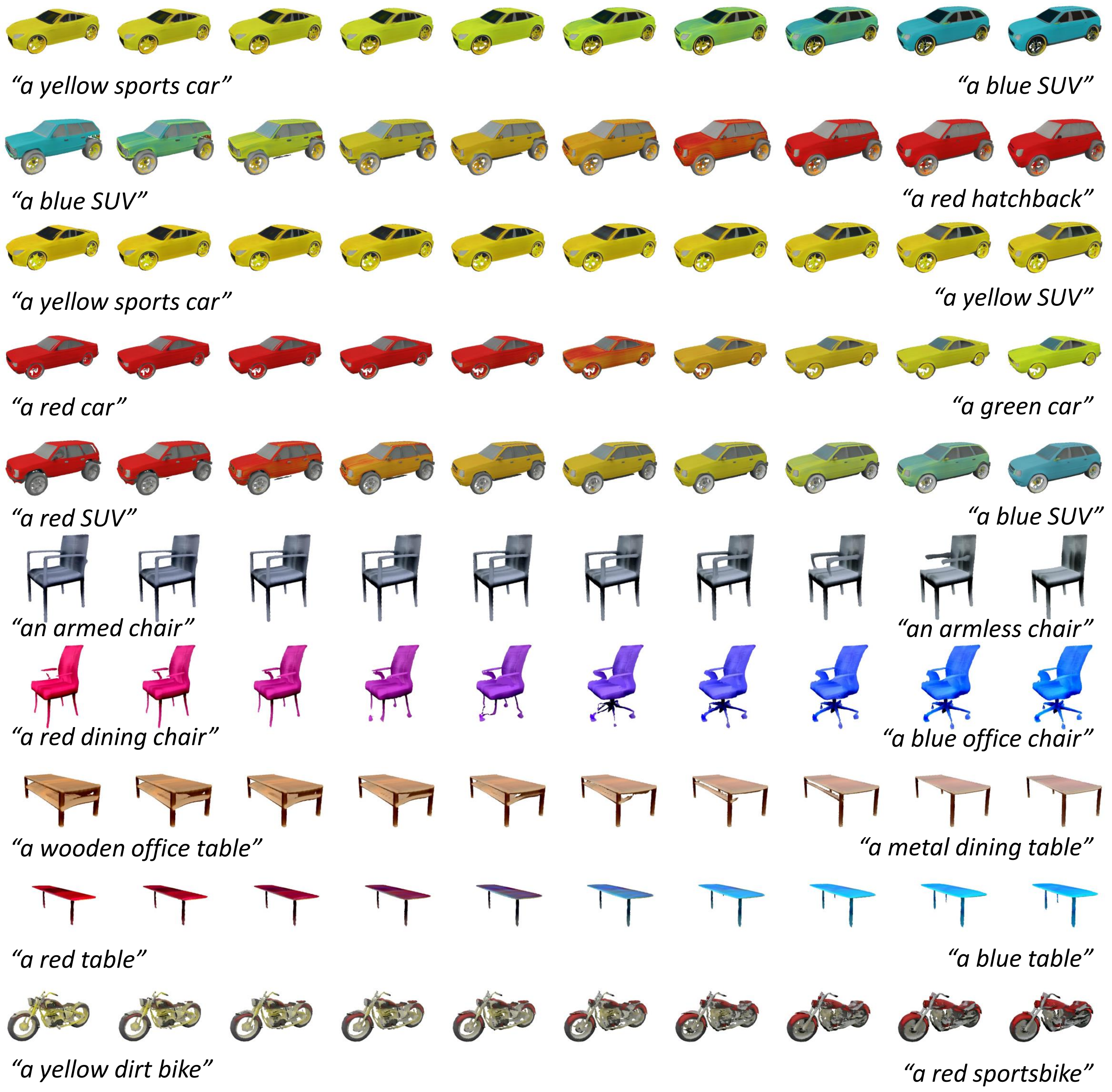

Figure D. We show the interpolation results between two text inputs. Note that each row shares the same sampled random noise. For each $\langle$source, target$\rangle$ pair, we use the same sampled random noise vector $\mathbf{z}$, and their corresponding CLIP text features to generate the latent codes $\mathbf{w}$. Then, the interpolation is performed between the source and target latent codes $\langle \mathbf{w}_1, \mathbf{w}_2 \rangle$. The interpolated latent codes are fed into the generator to synthesize the results.