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

"a red hatchback car"   "a gray armless chair"   "a wooden office table"   "a black motorbike"
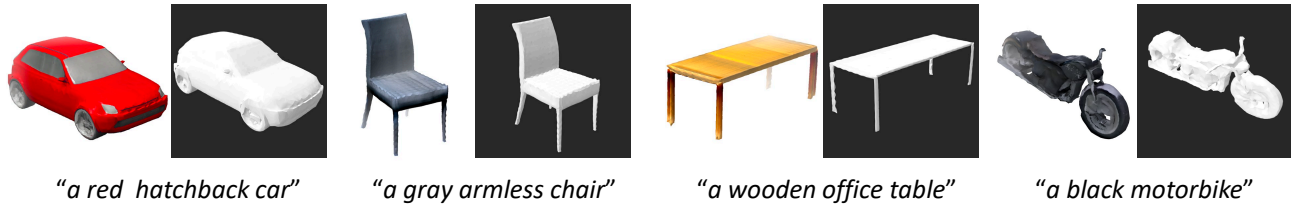
Figure 1. Our proposed model is capable of generating detailed textured shapes based on given text prompts. We demonstrate the effectiveness of our approach by showcasing generated results for four distinct object classes: *Car*, *Chair*, *Table*, and *Motorbike*. Both the textured meshes and geometries are presented, with visualizations rendered using ChimeraX [10].

Abstract

In this paper, we investigate an open research task of generating controllable 3D textured shapes from the given textual descriptions. Previous works either require ground truth caption labeling or extensive optimization time. To resolve these issues, we present a novel framework, TAPS3D, to train a text-guided 3D shape generator with pseudo captions. Specifically, based on rendered 2D images, we retrieve relevant words from the CLIP vocabulary and construct pseudo captions using templates. Our constructed captions provide high-level semantic supervision for generated 3D shapes. Further, in order to produce fine-grained textures and increase geometry diversity, we propose to adopt low-level image regularization to enable fake-rendered images to align with the real ones. During the inference phase, our proposed model can generate 3D textured shapes from the given text without any additional optimization. We conduct extensive experiments to analyze each of our proposed components and show the efficacy of our framework in generating high-fidelity 3D textured and text-relevant shapes. Code is available at https://github.com/plusmultiply/TAPS3D

1. Introduction

3D objects are essential in various applications [21–24], such as video games, film special effects, and virtual reality. However, realistic and detailed 3D object models are usually hand-crafted by well-trained artists and engineers slowly and tediously. To expedite this process, many research works [3,9,13,34,48,49] use deep generative models to achieve automatic 3D object generation. However, these models are primarily unconditioned, which can hardly generate objects as humans will.

In order to control the generated 3D objects from text, prior text-to-3D generation works [43,44] leverage the pretrained vision-language alignment model CLIP [37], such that they can only use 3D shape data to achieve zero-shot learning. For example, Dream Fields [13] combines the advantages of CLIP and NeRF [27], which can produce both 3D representations and renderings. However, Dream Fields costs about 70 minutes on 8 TPU cores to produce a single result. This means the optimization time during the inference phase is too slow to use in practice. Later on, GET3D [9] is proposed with faster inference time, which incorporates StyleGAN [15] and Deep Marching Tetrahedral (DMTet) [46] as the texture and geometry generators respectively. Since GET3D adopts a pretrained model to do text-guided synthesis, they can finish optimization in less time than Dream Fields. But the requirement of test-time optimization still limits its application scenarios. CLIP-NeRF [50] utilizes conditional radiance fields [45] to avoid test-time optimization, but it requires ground truth text data for the training purpose. Therefore, CLIP-NeRF is only applicable to a few object classes that have labeled text data for training, and its generation quality is restricted by the NeRF capacity.

To address the aforementioned limitations, we propose

---

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

to generate pseudo captions for 3D shape data based on their rendered 2D images and construct a large amount of ⟨3D shape, pseudo captions⟩ as training data, such that the text-guided 3D generation model can be trained over them. To this end, we propose a novel framework for **T**ext-guided 3D textured sh**A**pe generation from **P**seudo **S**upervision (TAPS3D), in which we can generate high-quality 3D shapes without requiring annotated text training data or test-time optimization.

Specifically, our proposed framework is composed of two modules, where the first generates pseudo captions for 3D shapes and feeds them into a 3D generator to conduct text-guided training within the second module. In the pseudo caption generation module, we follow the language-free text-to-image learning scheme [20, 54]. We first adopt the CLIP model to retrieve relevant words from given rendered images. Then we construct multiple candidate sentences based on the retrieved words and pick sentences having the highest CLIP similarity scores with the given images. The selected sentences are used as our pseudo captions for each 3D shape sample.

Following the notable progress of text-to-image generation models [29, 38, 39, 42, 52], we use text-conditioned GAN architecture in the text-guided 3D generator training part. We adopt the pretrained GET3D [9] model as our backbone network since it has been demonstrated to generate high-fidelity 3D textured shapes across various object classes. We input the pseudo captions as the generator conditions and supervise the training process with high-level CLIP supervision in an attempt to control the generated 3D shapes. Moreover, we introduce a low-level image regularization loss to produce fine-grained textures and increase geometry diversity. We empirically train the mapping networks only of a pretrained GET3D model so that the training is stable and fast, and also, the generation quality of the pretrained model can be preserved.

Our proposed model TAPS3D can produce high-quality 3D textured shapes with strong text control as shown in Fig. 1, without any per prompt test-time optimization. Our contribution can be summarized as:

- We introduce a new 3D textured shape generative framework, which can generate high-quality and fidelity 3D shapes without requiring paired text and 3D shape training data.

- We propose a simple pseudo caption generation method that enables text-conditioned 3D generator training, such that the model can generate text-controlled 3D textured shapes without test time optimization, and significantly reduce the time cost.

- We introduce a low-level image regularization loss on top of the high-level CLIP loss in an attempt to produce fine-grained textures and increase geometry diversity.

2. Related Work

2.1. Text-Guided 3D Shape Generation

Text-guided 3D shape generation aims to generate 3D shapes from the textual descriptions so that the generation process can be controlled. There are mainly two categories of methods, i.e., fully-supervised and optimization-based methods. The fully-supervised method [6, 8, 25] uses ground truth text and the paired 3D objects with explicit 3D representations as training data. Specifically, CLIP-Forge [43] uses a two-stage training scheme, which consists of shape autoencoder training, and conditional normalizing flow training. VQ-VAE [44] performs zero-shot training with 3D voxel data by utilizing the pretrained CLIP model [37].

Regarding the optimization-based methods [13, 18, 26, 34], Neural Radiance Fields (NeRF) are usually adopted as the 3D generator. To generate 3D shapes for each input text prompt, they use the CLIP model to supervise the semantic alignment between rendered images and text prompts. Since NeRF suffers from extensive generation time, 3D-aware image synthesis [3, 3, 4, 11, 30, 31] has become popular, which generates multi-view consistent images by integrating neural rendering in the Generative Adversarial Networks (GANs). Specifically, there are no explicit 3D shapes generated during the process, while the 3D shapes can be extracted from the implicit representations, such as the occupancy field or signed distance function (SDF), using the marching cube algorithm. These optimization-based methods provide a solution to generate 3D shapes, but their generation speed was compromised. Although [50] is free from test-time optimization, it requires text data for training, which limits its applicable 3D object classes. Our proposed method attempts to alleviate both the paired text data shortage issues and the long optimization time in the previous work. We finally produce high-quality 3D shapes to bring our method to practical applications.

2.2. Text-to-Image Synthesis

We may draw inspiration from text-to-image generation methods. Typically, many research works [36, 40, 53] adopt the conditional GAN architecture, where they directly take the text features and concatenate them with the random noise as the input. Recently, autoregressive models [7, 39] and diffusion models [29, 38, 41, 42] made great improvement on text to image synthesis while demanding huge computational resources and massive training data.

With the introduction of StyleGAN [14–16] mapping networks, the input random noise can be first mapped to another latent space that has disentangled semantics, then the model can generate images with better quality. Further, exploring the latent space of StyleGAN has been proved useful by several works [1, 17, 33] in text-driven image synthe-
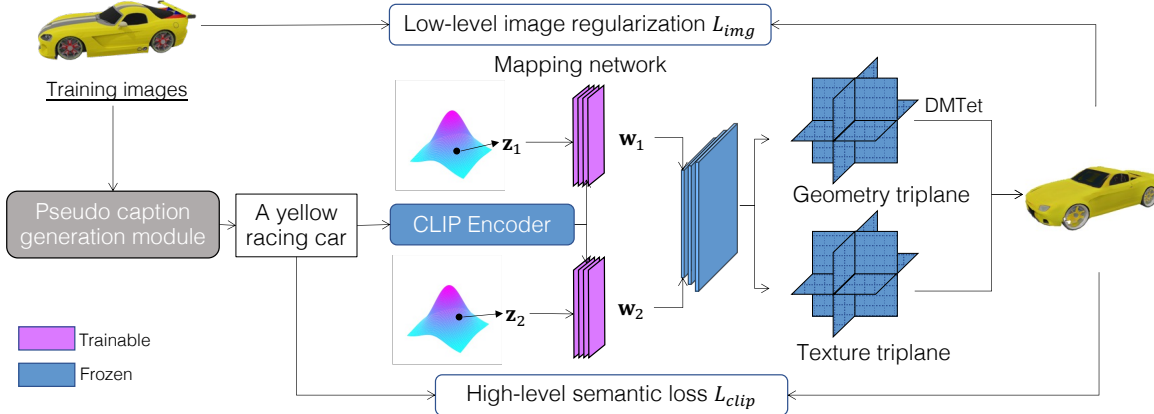
Figure 2. Our proposed framework supports text-guided 3D shape generation training without paired text labeling. To generate captions for training, we first adopt a pseudo caption generation module to produce textual descriptions, given rendered 2D images. Then we feed the pseudo caption CLIP embeddings into the mapping networks to train the controllable text-guided 3D shape synthesis network. Our training pipeline is supervised by high-level semantic loss $\mathcal{L}_{clip}$ and low-level image regularization $\mathcal{L}_{img}$. It is notable that during the training phase, only the mapping network weights are updated, and the rest components are fixed.

sis and manipulation tasks, where they utilize the pretrained vision-language model CLIP [37] model to manipulate pre-trained unconditional StyleGAN networks. In order to relieve the need for paired text data during the training phase, Lafite [55] proposes to adopt the image CLIP embeddings as training input while using text CLIP embedding during the inference phase. However, as discussed in the recent works [20, 54], there exists a modality gap between text and image embeddings in the CLIP latent space. Therefore, language-free methods that are solely trained on CLIP image embedding could introduce distribution mismatch when encountering the CLIP text embedding input during inference. In this paper, we further explore language-free training for text-guided 3D textured shape generation by proposing a pseudo caption generation module.

3. Method

3.1. Preliminary

GET3D [9] is a newly proposed 3D generator that can generate high-fidelity textured 3D shapes. Specifically, GET3D maps noise vectors $\mathbf{z} \in \mathcal{N}(0, \mathbf{I})$ to a textured mesh. In order to disentangle the geometry and texture information, GET3D takes two random vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ as inputs for the geometry and texture generation branches, respectively. Following the design of StyleGAN [15], and EG3D [3], they map $\mathbf{z}_1$ and $\mathbf{z}_2$ to the intermediate latent codes $\mathbf{w}_1$ and $\mathbf{w}_2$ with two mapping networks since the semantics would be better represented in the mapped space. The geometry and texture generators are built conditioned on $\mathbf{w}_1$ and $\mathbf{w}_2$.

3.2. Overview

Since the GET3D [9] model only supports unconditional 3D generation, if one need to do text-guided 3D shape syn-

thesis, additional optimization is required with the original GET3D model. In order to control the semantics of the generated results and achieve text-based textured 3D generation in one shot, we propose to add text conditions to GET3D models. In Fig. 2, we present our training framework.

To this end, we address two main challenges: one is the lack of training textual descriptions, and another is color and geometry distortion. Regarding the first challenge, we propose a pseudo caption generation module, which gives large amounts of captions for training purposes. These captions are used as conditions and fed into the mapping networks to control the semantic meanings of the output textured 3D shapes. We apply the high-level CLIP [37] semantic loss on the paired input text and rendered 2D images to achieve text-3D alignment. To resolve the second challenge, we propose to regularize the color and geometry training with low-level image regularization. This method provides complimentary image-level guidance to our learning objective, which gives fine-grained textures during the generation phase.

Notably, we empirically adopt the pretrained GET3D model and train its mapping networks only since we observe this practice not only improves the training efficiency but also stabilizes the training process.

3.3. Pseudo Caption Generation

Most recent 3D shape generation models are data-driven. With large-scale training data, their performance improvement is remarkable. However, when we aim to train a text-guided 3D shape generation model, we can only obtain a few textual descriptions for limited samples. To resolve this issue, we follow the method of [54] and propose to generate pseudo captions for each 3D object.

Specifically, we observe the textual descriptions for a single 3D object are highly structured, e.g., *"A yellow*
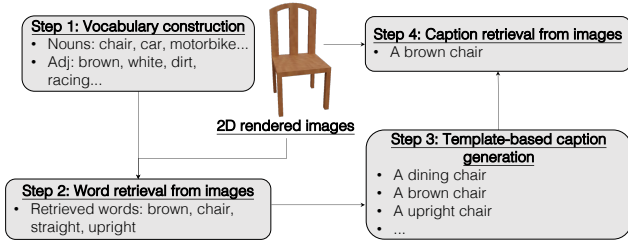
Figure 3. Our proposed pseudo caption generation module can be split into four steps. Firstly, we construct vocabulary from ShapeNet-related nouns and adjectives in the CLIP vocabulary. Then we retrieve multiple words based on the 2D-rendered images. In the third step, we create candidate captions with retrieved words. Finally, we select one caption based on the computed text-image similarities with the CLIP model.

sports car", which typically consists of adjectives and a noun. Therefore, it is critical to obtain suitable adjectives and nouns for given 3D objects to construct pseudo captions automatically. To this end, we adopt the pretrained text-image similarity model of CLIP [37] to retrieve useful nouns and adjectives and further build complete sentences.

The pseudo caption generation module can be split into four steps, shown in Fig. 3. In the first step, we construct a vocabulary to cover the related adjectives and nouns. Since we aim to generate ShapeNet [5] object classes, we collect all the class names as nouns in the vocabulary, which has 169 tokens. Then we extract all the adjectives from CLIP vocabulary with the Natural Language Toolkit [2]. The size of our collected adjectives is 1463.

In the second and third steps, we aim to retrieve relevant words from 2D-rendered images and construct candidate captions. Technically, we first extract all the vocabulary word embeddings and rendered image embeddings, where we adopt the pretrained CLIP text and image encoders, respectively. Then for each object, we compute the cosine similarities between the given image embedding and all the text embeddings. It is notable that we retrieve the nouns and adjectives separately, where we select the top-K_1 retrieved nouns and top-K_2 retrieved adjectives. With the selected words, we utilize the templates to construct candidate sentences. Since these sentences are only supposed to describe a static object, we use the template "a {Adjective} {Noun}", where we put 1 or 2 adjectives randomly to increase the caption diversity. Due to having K_1 nouns and K_2 adjectives, we are able to combine them and produce multiple candidate pseudo captions.

In order to obtain the most suitable captions for given objects, we further conduct image-to-candidate caption retrieval, where we also use CLIP encoders to extract the image and sentence embeddings. To be specific, we use the top retrieved sentences as the final pseudo captions, which are used as the input condition for the 3D generator.
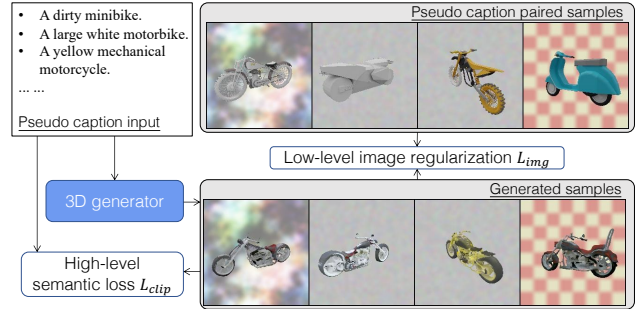

Figure 4. In the training phase, we not only conduct the high-level text-to-image similarity learning through the CLIP model but also use low-level image regularization to obtain fine-grained textures and diverse shapes, where we render real and fake images with the same camera poses and apply background augmentation to allow the model focus on object features only.

3.4. Text-3D Alignment Training

After the pseudo caption generation, we feed these captions into a pretrained GET3D [9] model in an attempt to control the semantic meanings of generated 3D objects. To be specific, we first adopt the CLIP text encoder E_t to extract the caption t features, then the caption features $E_t(t)$ are served as conditions and fed into the mapping networks of pretrained GET3D models. The input of the original mapping networks is random noise only. To inject the text information, we concatenate the caption embeddings and the random noise together as the new input. This enables us to generate diverse 3D objects that match the text's semantic meanings.

Since we adopt the pretrained 3D generator as our model backbone, which has a good ability to synthesize high-fidelity 3D textured shapes, fixing the generator weights allows our model to focus on getting semantic alignment between input textual descriptions and generated 3D objects and also alleviates the learning difficulty. Our main training objective is to maximize the image-text similarity measured by the CLIP model, which can be denoted as:

$$\mathcal{L}_{clip} = 1 - \cos(E_i(I_x), E_t(t)). \tag{1}$$

Here E_i and I_x represent the CLIP image encoder and rendered images, respectively. These 2D images are rendered from the generated 3D textured shapes with randomly sampled camera poses.

3.5. Towards Diverse and Fine-Grained Generation

It is observed using $\mathcal{L}_{clip}$ only enables the model to generate plausible 3D shapes that align with the input text. However, the diversity of geometry generation is limited, and the generated colors are unnatural. This is caused by two main reasons. One reason is the diversity of our pseudo captions is also limited, as there are only a few descriptive words for a single object class. Moreover, the CLIP model

may wrongly retrieve irrelevant words to the given objects, which also affects the generated shape quality. Another reason is the CLIP model only provides high-level guidance for our training supervision while failing to give fine-grained texture information, thus making the generated results unrealistic.

In order to resolve this issue, we further introduce a low-level image regularization loss. Specifically, we use the same camera poses as that used in the I_x rendering to produce ground truth rendered images I_x^{gt}. The optimization objective is denoted as

$$\mathcal{L}_{img} = 1 - \cos(E_i(I_x), E_i(I_x^{gt})). \qquad (2)$$

Since the rendering process returns an alpha channel, which results in a simple white or black background, if we directly take rendered images with the same backgrounds over training iterations, the background information will also be involved during optimizing Eq. (2). To alleviate this issue and allow the model to focus on the foreground objects, we follow [13] to conduct background augmentation, as shown in Fig. 4. Technically, we randomly select random Fourier textures [28], Gaussian noise, and checkerboard patterns as backgrounds. For each paired real and fake image, we apply the same augmented backgrounds.

3.6. Overall Training Objective

Our overall training objective can be depicted as

$$\mathcal{L} = \mathcal{L}_{clip} + \mathcal{L}_{img}. \qquad (3)$$

During the training phase, we empirically experiment with updating all model parameters and the mapping networks only. We observe training all GET3D backbone parameters would severely distort the generated geometry and textures, where we show the experimental results in the supplementary material. This is because the learning objective is two-pronged: improving the generation quality and matching the input captions. While borrowing a pretrained unconditional generator ensures good generation quality, expedites the training process, and also yields stable text-guided results. Therefore, we regard training the mapping networks only of a pretrained 3D generator as a useful and practical solution for the text-guided 3D shape generation task.

4. Experiments

4.1. Dataset

We train and evaluate our method on the large-scale 3D dataset ShapeNet [5]. Our experiments are conducted on four classes with complex geometry, *Car*, *Table*, *Chair*, and *Motorbike*, containing 7497, 8436, 6778, and 337 shapes, respectively. We follow GET3D [9] to split the data into training, validation, and test sets in a 7:1:2 ratio. The training image data are rendered following the GET3D setting.

a white automobile a low race car   a ural motorbike   a brown motorcycle

a stationary chair   a white ladder-back chair   a brown rectangular table   a structural table
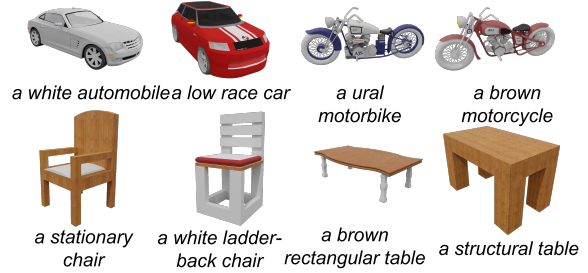
Figure 5. The demonstration of our generated pseudo captions.

We render 24 views randomly for *Car*, *Chair* and *Table* and 100 views for *Motorbike* since it has less available data. The pretrained models were trained on each class separately.

4.2. Implementation details

We use the GET3D [9] model pretrained on ShapeNet [5] dataset as our backbone 3D generator. In the pseudo caption generation module, we set K_1 and K_2 as 3 and 6, respectively. We generate 20 pseudo captions for each ShapeNet object. During the training phase, each input image is randomly paired with one of the pseudo captions generated for the object which the image was rendered from. An example of the generated pseudo can be found in Fig. 5. We only update the geometry and texture mapping networks. We set the batch size to 16 and ran all the experiments on 4 Nvidia Tesla V100-32G GPUs. We empirically observe the geometry branch is more sensitive to the weight update. Hence we set learning rates for geometry and texture mapping networks as 0.004 and 0.0005, respectively. It costs around 10 hours for model training.

4.3. Comparison with Existing Methods

4.3.1 Qualitative comparison

In Fig. 6, we first compare with the method TITG3SG [25], which is an encoder-decoder-based framework. They only experiment with object classes of *Chair* and *Table*. Here we conduct the comparison qualitatively. Specifically, we show the generated explicit 3D shapes with the corresponding text prompts. Since TITG3SG [25] generates 3D shapes in two steps, where the coarse voxel and refined mesh are produced separately. We show them both in the top row of Fig. 6. The second row presents our generated mesh in two different views, in which we generates final 3D results in one shot. We observe our method gives better 3D textured shapes than TITG3SG, which is a fully supervised method. For example, in the third generated sample of TITG3SG, the chair legs are distorted. Moreover, the high alignment between input prompts and our generated results demonstrates our text control ability.
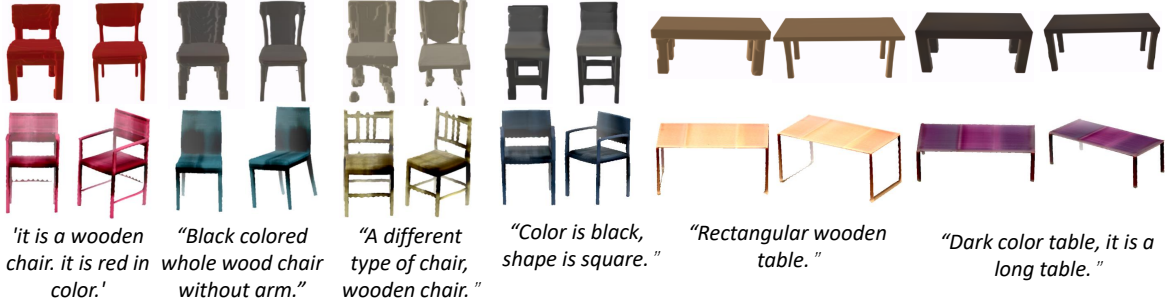
Figure 6. We compare the text-guided generation results of our methods with that of the fully supervised method TITG3SG [25] on two object classes, *Chair* and *Table*. On the top row, we present the results from [25], for each text prompt, the top left one is the raw voxel output in 64^3 resolution, and the top right one is the refined mesh based on the voxel output. The bottom row shows our mesh output with two different views. The renderings are generated with ChimeraX [10].

4.3.2 Quantitative comparison

To demonstrate our 3D generation quality, we compare existing works [25,43,50] from three aspects: 1) rendered 2D image quality using FID, 2) text-image relevance using R-Precision, and 3) 3D geometry quality using FPD.

In Tab. 1, we provide a quantitative comparison with CLIP-NeRF [50], which utilizes human-written captions for training. We employ the Fréchet Inception Distance (FID) [12] as our evaluation metric and present results for two object classes: *Car* and *Chair*. To ensure fair comparisons, we downsample our results to match the resolution of CLIP-NeRF [50]. CLIP-NeRF employs NeRF as the 3D generator, which imposes limitations on the generation resolution. In contrast, our backbone GET3D model boasts a larger capacity, supporting higher resolution. Experimental results demonstrate that our method surpasses CLIP-NeRF in text-guided generation quality.

In Tab. 2, we adopt the CLIP-R-Precision metric [32] to evaluate the consistency between input text prompts and generated shapes. Given that our model is category-specific, we perform per-category CLIP-R-Precision and compare it to the fully supervised TITG3SG method [25]. According to [32], we compute the CLIP-R-Precision using a set of unseen random captions with words sampled from existing captions, thereby enhancing correlation with human evaluations and mitigating model bias inherent compared to the conventional R-Precision metric. It is observed our method outperforms TITG3SG [25] in each category.

In Tab. 3, we assess the geometry generation quality using the Fréchet Point Distance (FPD) [19,47], which measures the feature distance of a PointNet [35] pretrained on ModelNet40 [51]. Our method outperforms existing approaches in terms of geometry quality by a large margin.

4.3.3 Inference speed comparison

In Tab. 4, we compare the inference speed of our method with other works. The optimization-based methods Dream-Fields [13], DreamFusion [34], and PureCLIPNeRF [18]

Table 1. Comparison with the existing work. We evaluate the rendered 2D images using Fréchet inception distance (FID). We downsample our result to the same resolution of CLIP-NeRF [50] for fair comparisons.

	Car		Chair	
	Resolution	FID	Resolution	FID
CLIP-NeRF [50]	256^2	67.8	128^2	48.4
Ours	256^2	**20.1**	128^2	**43.7**
Ours	1024^2	**21.7**	1024^2	**44.8**

Table 2. Per category CLIP-R-Precision score using random generated captions on categories *Chair* and *Table*.

Method	Chair	Table
TITG3SG [25]	9.05 ± 1.92	6.67 ± 3.25
Ours	$\mathbf{20.19 \pm 1.89}$	$\mathbf{15.21 \pm 3.15}$

Table 3. Comparison of 3D generation quality in FPD score.

Method	Chair	Table
TITG3SG [25]	1566.76	1639.68
CLIP-Forge [43]	825.96	3051.31
Ours	**342.23**	**1468.43**

Table 4. The comparison of inference time per text prompt for different methods. The inference time of DreamFields [13], Dream-Fusion [34], and PureCLIPNeRF [18] are given in their papers. For TITG3SG [25] and ours, we generate 1000 text prompts with batch size as 1, and calculate the average inference time.

Method	Device	Output	Time
DreamFields [13]	TPU cores x8	Rendering	72 min
DreamFusion [34]	TPUv4 machine	Rendering	90 min
PureCLIPNeRF [18]	GTX 2080ti	Rendering	20 min
TITG3SG [25]	Telsa V100-32G	Voxel	2.21 sec
TITG3SG [25]	Telsa V100-32G	Mesh	24.44 sec
Ours	Telsa V100-32G	Rendering	0.05 sec
Ours	Telsa V100-32G	Mesh	7.09 sec

take tens of minutes to optimize for each input text prompt. Liu et al. [25] use an encoder-decoder architecture and thus can perform inference at 2.21 seconds. However, the raw 64^3 voxel output is very coarse, as shown in Fig. 6. The
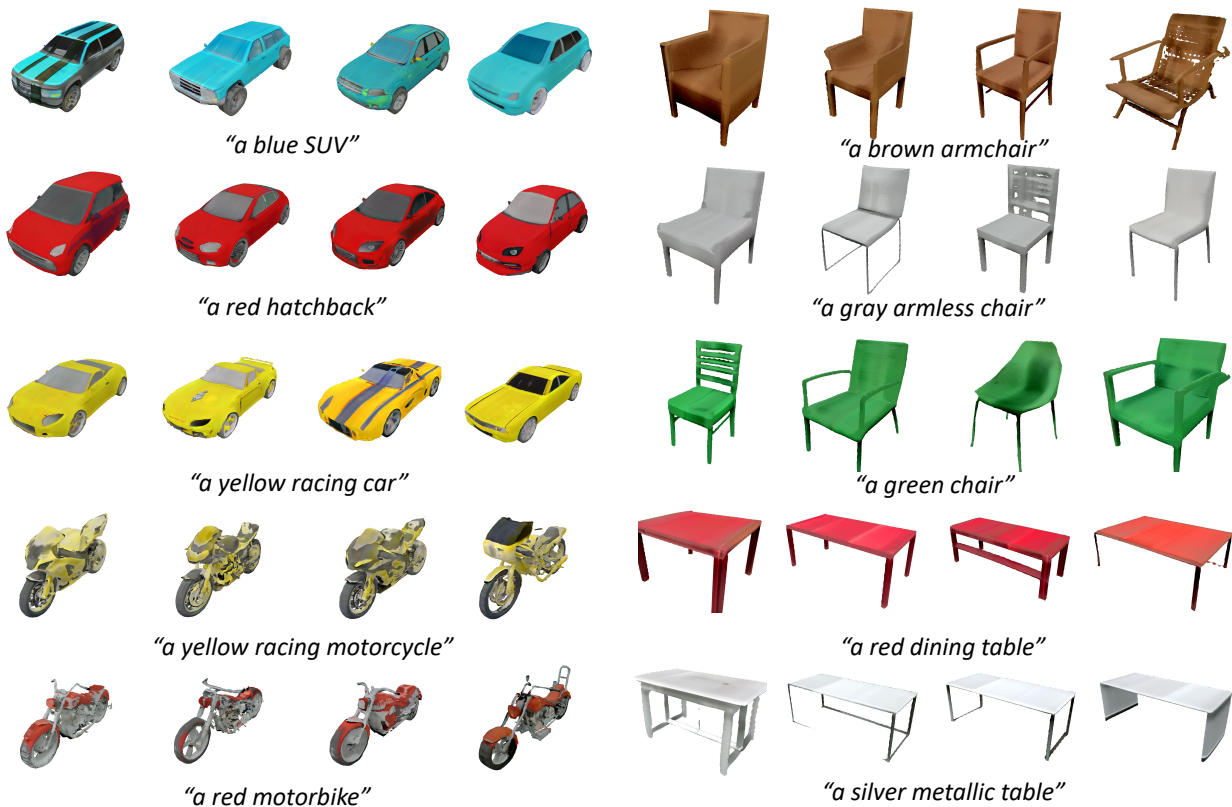
Figure 7. We show the generation diversity and text control of our method. Each row takes the same text prompt with different sampled random noise as the model input.
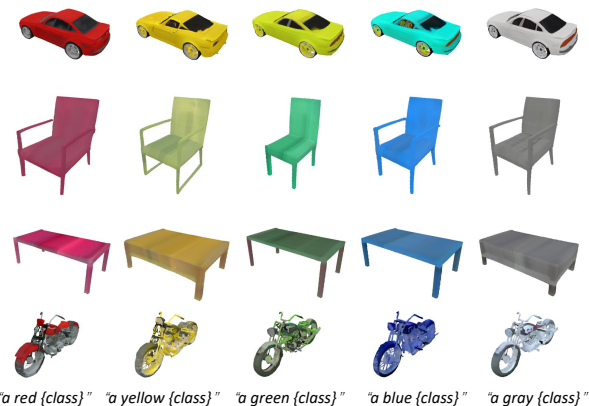

Figure 8. Text control of textures on four different categories. Each row shares the same sampled random noise input.

coarse output is later refined with interpolations and marching cubes to generate mesh data. This process takes ten times longer than the network inference time and costs 24.44 seconds per sample. Our method generates high-resolution renderings at 1024^2 resolution with only 0.05 seconds and generates the complete explicit mesh representation in 7.09 seconds.

4.4. Qualitative Results

In Fig. 7, we present the generation diversity and text control strength of our method in four object classes. In each row, we sample different random noises as well as the given text prompt as input to generate diversified and text-controlled 3D shapes, where the input text features are simultaneously fed into both the geometry and texture mapping network. Generally, we observe the semantic meanings of the rendered 2D images, and the given prompts are consistent. Moreover, there exists large diversity in the texture and geometry of our generated 3D shapes, even from the same text input. Here we present generated results of 4 classes, where *Car* has a relatively simple geometry, while *Table* and *Chair* are more complex. But we still have good control over the textures and geometries for the challenging classes. For example, when we input *"a gray armless chair"*, the generated samples exactly match the given instructions. However, we also observe failure cases of *Table* generation. For instance, the generation of *"a wooden office desk"* may have distorted geometry due to its complex structure. In Fig. 8, we show the text control ability of our method on different classes. The results in each row are generated from the same random noise input but with dif-

Table 5. We compare the FID score of using image CLIP features (Lafite [55]) and the pseudo text CLIP features (Ours) as training input.

Method	Car	Chair	Table	Motorbike
Lafite [55]	68.7	81.1	94.3	146.5
Ours	**21.7**	**44.8**	**43.2**	**57.2**

Table 6. We compare the FID score for our method without $\mathcal{L}_{clip}$, $\mathcal{L}_{img}$ and background augmentation (BGaug).

Method	Car	Chair	Table	Motorbike
Ours	**21.7**	**44.8**	**43.2**	**57.2**
-w/o $\mathcal{L}_{clip}$	83.7	148.9	67.1	134.9
-w/o $\mathcal{L}_{img}$	59.8	144.2	48.4	101.8
-w/o BGaug	55.1	65.1	58.1	110.0

ferent text prompts. To summarize, the results show that our generation results are highly consistent with the input text prompts across different object classes.

4.5. Ablation Study

Pseudo caption generation module. In Tab. 5, we compare our method with Lafite [55], which is a language-free training method for the text-to-image generation task. Lafite claims due to the alignment between CLIP text and image embeddings. The CLIP image features can be directly used as text input during training. To implement Lafite [55] onto our framework, we feed the rendered real image CLIP features into our model and train the model by minimizing the cosine similarity between the generated and input image CLIP features. The quantitative results show that our proposed pseudo caption generation module significantly improves the generation quality compared to the baseline. We also give a qualitative comparison in Fig. 9 (b), we observe that the generator trained with Lafite [55] fails to generate correct textures. This problem might come from the modality gap between the CLIP image and text embeddings, as discussed in [20, 54].

Effects of our loss functions. We conduct an ablation study on our used two losses $\mathcal{L}_{clip}$ and $\mathcal{L}_{img}$. In Tab. 6, we quantitatively validate the usefulness of both two losses. Specifically, $\mathcal{L}_{clip}$ supervises the text-image alignment learning, which is more critical than $\mathcal{L}_{img}$, and removing $\mathcal{L}_{clip}$ gives a more considerable performance drop. From the qualitative results in Fig. 9 (c), we observe that simply using $\mathcal{L}_{img}$ leads to the model generating distorted geometries, which may indicate the significance of high-level semantic supervision. We also observe that solely using $\mathcal{L}_{clip}$ (Fig. 9 (d)) may cause the model generation to lose fine-grained texture details, as well as the geometry information.

Effects of background augmentation. Our used GET3D [9] backbone network generates RGBA images with dark backgrounds, which may bring some noise when

Figure 9. Ablation study on our methods with text prompt *"a red car"*. We show results of (a) our full model, (b) Lafite [55], (c) without high-level semantic regularization $\mathcal{L}_{clip}$, (d) without low-level image regularization $\mathcal{L}_{img}$, (e) without background augmentation. All the visualizations are generated from models trained for 250k iterations.

"a chair with yellow legs, red seat and blue back."

Figure 10. Examples of failure cases of our method. Our model often fails to handle text input with composed descriptions for different parts of the object.

we conduct low-level image regularization. As illustrated from the quantitative results in Tab. 6 and the visualizations in Fig. 9 (e), training with the augmented background can help remove the texture flaws.

5. Conclusion

We introduce TAPS3D, a text-guided 3D textured shape generation method. We first generate pseudo captions for model training. During the inference phase, TAPS3D does not require additional optimization. Our fast generation speed enables average users to generate high-quality 3D objects with acceptable processing time.

Limitation. The limitation of our method mainly lies in the capacity of our model is still bound to the training image data, and we cannot produce different fine-grained details for different object parts, as shown in Fig. 10. Although we do not require paired text data, we still need diversified training images to handle complex text input.

Acknowledgment

References

[1] Rameen Abdal, Peihao Zhu, John Femiani, Niloy Mitra, and Peter Wonka. Clip2stylegan: Unsupervised extraction of stylegan edit directions. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 2

[2] Edward Loper Bird, Steven and Ewan Klein. Natural language processing with python. *O'Reilly Media Inc.*, 2009. 4

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 1, 2, 3

[4] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021. 2

[5] 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. 4, 5

[6] 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

[7] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 2

[8] 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

[9] 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, 3, 4, 5, 8

[10] 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. 1, 6

[11] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 2

[12] 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. 6

[13] 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. 1, 2, 5, 6

[14] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 2

[15] 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, 2, 3

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 2

[17] Umut Kocasari, Alara Dirik, Mert Tiftikci, and Pinar Yanardag. Stylemc: Multi-channel based fast text-guided image generation and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 895–904, 2022. 2

[18] Han-Hung Lee and Angel X Chang. Understanding pure clip guidance for voxel grid nerf models. *arXiv preprint arXiv:2209.15172*, 2022. 2, 6

[19] Ruihui Li, Xianzhi Li, Ka-Hei Hui, and Chi-Wing Fu. Spgan: Sphere-guided 3d shape generation and manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021. 6

[20] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *arXiv preprint arXiv:2203.02053*, 2022. 2, 3, 8

[21] Kangcheng Liu. Rm3d: Robust data-efficient 3d scene parsing via traditional and learnt 3d descriptors-based semantic region merging. *International Journal of Computer Vision*, pages 1–30, 2022. 1

[22] Kangcheng Liu, Zhi Gao, Feng Lin, and Ben M Chen. Fgnet: A fast and accurate framework for large-scale lidar point cloud understanding. *IEEE Transactions on Cybernetics*, 53(1):553–564, 2022. 1

[23] Kangcheng Liu, Aoran Xiao, Xiaoqin Zhang, Shijian Lu, and Ling Shao. Fac: 3d representation learning via foreground aware feature contrast. *arXiv preprint arXiv:2303.06388*, 2023. 1

[24] Kangcheng Liu, Yuzhi Zhao, Qiang Nie, Zhi Gao, and Ben M Chen. Weakly supervised 3d scene segmentation with region-level boundary awareness and instance discrimination. In *European Conference on Computer Vision 2022 (ECCV 2022)*, pages 37–55. Springer, Cham, 2022. 1

[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, 6

[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] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[28] Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 3(7):e12, 2018. 5

[29] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[30] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 2

[31] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513, 2022. 2

[32] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 6

[33] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 2

[34] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2, 6

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 6

[36] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1505–1514, 2019. 2

[37] 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, 2, 3, 4

[38] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[39] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2

[40] 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. 2

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2

[42] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 2

[43] 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. 1, 2, 6

[44] Aditya Sanghi, Rao Fu, Vivian Liu, Karl Willis, Hooman Shayani, Amir Hosein Khasahmadi, Srinath Sridhar, and Daniel Ritchie. Textcraft: Zero-shot generation of high-fidelity and diverse shapes from text. *arXiv preprint arXiv:2211.01427*, 2022. 1, 2

[45] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 1

[46] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021. 1

[47] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3859–3868, 2019. 6

[48] Chaoyue Song, Jiacheng Wei, Ruibo Li, Fayao Liu, and Guosheng Lin. 3d pose transfer with correspondence learning and mesh refinement. *Advances in Neural Information Processing Systems*, 34:3108–3120, 2021. 1

[49] Chaoyue Song, Jiacheng Wei, Ruibo Li, Fayao Liu, and Guosheng Lin. Unsupervised 3d pose transfer with cross consistency and dual reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[50] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3835–3844, 2022. 1, 2, 6

[51] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 6

[52] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 2

[53] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017. 2

[54] Yufan Zhou, Chunyuan Li, Changyou Chen, Jianfeng Gao, and Jinhui Xu. Lafite2: Few-shot text-to-image generation. *arXiv preprint arXiv:2210.14124*, 2022. 2, 3, 8

[55] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Lafite: Towards language-free training for text-to-image generation. *arXiv preprint arXiv:2111.13792*, 2021. 3, 8