

ISS: IMAGE AS STEPPING STONE FOR TEXT-GUIDED 3D SHAPE GENERATION: SUPPLEMENTARY MATERIAL

Zhengzhe Liu¹ Peng Dai² Ruihui Li³ Xiaojuan Qi^{2*} Chi-Wing Fu^{1*}

¹The Chinese University of Hong Kong ²The University of Hong Kong ³Hunan University

In this supplementary material, we first introduce the background augmentation in text-guided shape stylization (Section 1). Next, we present the implementation details and evaluation metrics, and provide the details of human perceptual evaluation results (Section 2). Then, we introduce the setup and analysis of our ablation studies (Section 3). After that, we discuss text-guided stylization and provide more results (Section 4). Then we show additional generative results of our approach (Section 5). Further, we discuss two alternative training strategies (Section 6). Then, we show how feature is mapped in the latent space (Section 7) and also review some related literatures about SVR and differentiable rendering (Section 8). Afterwards we compare the number of parameters in our model and existing works (Section 9). We also discuss failure cases (Section 10) and limitations of this work (Section 11). Finally, we provide our text sets (Section 12) and summarize the notations in the paper (Section 13).

1 BACKGROUND AUGMENTATION IN TEXT-GUIDED SHAPE STYLIZATION

One important thing in text-guided shape stylization is that the generated texture should align with the given shape. However, it cannot be ensured with a simple white or black background during training since the generated textures can be affected by the background color. As shown in Figure 1 (a), the shape in white color may confuse with the white background; thus, the model would struggle to capture the boundary of objects, hence cannot generate textures that well-align with the table. In Figure 1 (c), the generated texture is severely affected by the black background color, causing low-quality stylization results.

To address the above issue, we propose a background augmentation strategy to improve the alignment between texture and shape. Specifically, the background color is replaced with random RGB values in each training iteration. By this means, the foreground shape may be more easily captured in training as shown in Figure 1 (b,d), improving the texture-shape consistency and the stylization quality.

Discussion on L_{bg} and background augmentation. In two-stage feature space alignment (Section 3.3 in the main paper), we introduce a background loss L_{bg} to encourage the color prediction on the background region to be white. A natural question is whether we can use background augmentation as a replacement, and our answer is no. As shown in Figure 2 (a), the background color can affect the cosine similarity of CLIP features between the image and input text; thus, using different background color in each iteration makes stage-2 alignment unstable and affects shape generation. As a result, the two-stage alignment can only benefit from L_{bg} , but not from the background augmentation, for producing a plausible shape. Besides, we empirically found that the two-stage feature space alignment with L_{bg} performs well, even if a white shape is being considered, see the bottom row in Figure 5 (i).

2 IMPLEMENTATION DETAILS, METRICS, AND HUMAN PERCEPTUAL EVALUATION DETAILS

Implementation details. Our framework is implemented using PyTorch (Paszke et al., 2019). We first train the stage-1 CLIP-image-to-shape mapping for 400 epochs with learning rate $1e^{-4}$, and then train the stage-2 text-to-shape module at test time for 20 iterations which takes only 85 seconds on average on a single GeForce RTX 3090 Ti. Optionally, we can further train text-guided stylization with the same learning rate. We empirically set hyperparameters λ_M , λ_{bg} , t , m , τ_1 , τ_2 to be 0.5, 10, 0.5, 10, 0.2, 0.95, respectively, according to a small validation set.

Details on camera poses. In Stage 1, we follow Niemeyer et al. (2020) to set the camera poses to encourage the background to be white. Specifically, we randomly sample the distance of the camera and the viewpoint on the northern hemisphere.

In Stage 2, compared with Niemeyer et al. (2020), we sample the camera distance to be 1.5 times further compared with Niemeyer et al. (2020). It helps to encourage sampling more global views instead of only local ones, so that the CLIP image encoder can capture the whole shape and yield a better CLIP feature.

In Stage 3, we also sample the camera distance to be 1.5 times. Since this stage aims to generate textures instead of searching for a target shape like Stages 1 and 2, only sampling view points on the northern hemisphere of the view space cannot ensure good generation quality in the bottom regions. Thus, we further randomly sample viewpoints on the southern hemisphere for random 10% training iterations to encourage the stylized results to be consistent with the text in various viewpoints.

Decoder duplication Except for the output layer, we simply duplicate them to be D_o and D_c . For the output layer, it takes d -dimensional features as input and outputs one value for occupancy and three values for RGB. We then copy $d \times 1$ weights to D_o and copy $d \times 3$ weights to D_c . See Figure 4 for more details.

Metric: Shape generation quality. To measure the shape generation quality, we employ Fréchet Inception Distance (FID) (Heusel et al., 2017) between five rendered images of the generated shape with different camera poses and a set of ground-truth ShapeNet or CO3D images. We adopt the official model with Inception Net trained on ImageNet, which is widely used to evaluate generative quality and realism. We do not train a model on ShapeNet, since it is too small to train a better network for evaluating the FID than models trained on ImageNet. In addition, we randomly sample 2600 images in the ShapeNet dataset as ground truths for FID evaluation, instead of using images from ImageNet. It helps to measure the similarity between the generated shapes and ground truths of ShapeNet.

Besides adopting FID, we further utilize the metric Fréchet Point Distance (FPD) proposed in (Liu et al., 2022) to measure the shape generation quality without texture. We first convert the generated shapes to 3D point clouds without color (see Figure 3) and then evaluate FPD. Note that Dream Field (Jain et al., 2022) does not produce 3D shapes directly, so that we cannot evaluate this work in this regard.

To further assess the text-shape consistency, we conduct a human perceptual evaluation which is detailed as follows.

Human perceptual evaluation Setup. First, we prepare generated results for human evaluation. For each input text, we produce nine results from state-of-the-art methods, including CLIP-Forge (Sanghi et al., 2022) and Dream Fields (Jain et al., 2022), six baseline methods, and our full method; see Section 4.2 in the main paper and Section 3.1 in this supplementary material for details of each baseline. Second, we invite 10 volunteers (3 females and 7 males; aged from 19 to 58; all with normal vision) to evaluate the results. We show these results to the participants in random order without revealing how each result is produced. Then, they are asked to give a score from $\{1, 0.5, 0\}$ (1: perfect match, 0.5: partial match; and 0: don't match) on the degree of match between the generated

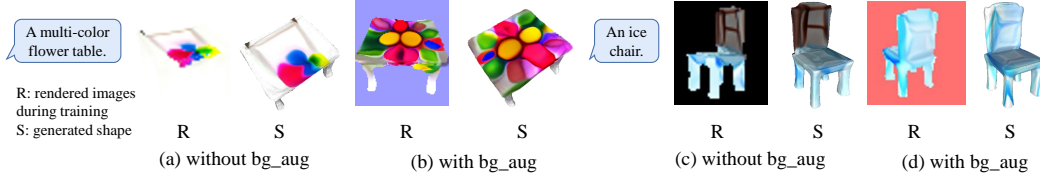


Figure 1: Effective of text-guided shape stylization with/without background augmentation.

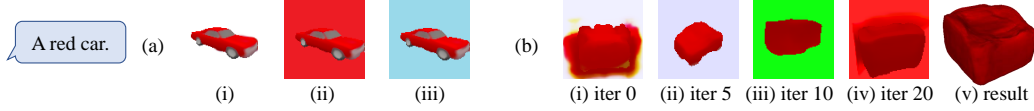


Figure 2: An investigation on background loss and background augmentation. (a) Background color affects the cosine similarity of the CLIP features between the image and the text “a red car”, *i.e.*, (i) 0.292 (ii) 0.303 and (iii) 0.285. (b) Effect of generating shapes with background augmentation, but without background loss L_{bg} . Comparing with Figure 4 (b) in the main paper, the two-stage feature space alignment works well with L_{bg} , but fails with the background augmentation.

shapes and input text. Then, for each method, we gather the evaluation scores from all participants and obtain the “Consistency Score” as s/n , where s is the total score and n is the number of samples.

Details of human perceptual evaluation results. We show Consistency Score and the preferences in the A/B/C test from each volunteer in Tables 1 and 2. As shown in Table 1, all ten volunteers consistently give the highest Consistency Score to our approach, and in the A/B/C test (see Table 2), all ten volunteers prefer results from our approach. The above further manifest the superiority of our model.

3 ABLATION STUDIES

3.1 BASELINE SETUPS

We create the following baselines in our ablation study. The first six baselines aim to assess the effectiveness of key modules in our approach, whereas the last two adopt state-of-the-art text-to-image generation approaches to first create images then adopt DVR (Niemeyer et al., 2020) to generate shapes for a fair comparison with our approach. Note that we do not adopt the most recent SVR model SS3D (Alwala et al., 2022) (which aims to work with in-the-wild images), due to its inferior generative quality and lack of texture generation.

- $E_1 + D$: As the first empirical study in Section 3.2 in the main paper, E_1 is adopted to extract the image feature f_1 and D is trained for 3D shape generation from f_1 without the two-stage feature-space alignment.
- w/o Stage 1: Stage-2 alignment is optimized from randomly initialized M , without stage 1.
- w/o Stage 2: Generate with M after stage 1, without the test-time optimization of stage 2.
- w/o L_{bg_1} : Remove L_{bg} in stage-1 alignment.
- w/o L_{bg_2} : Remove L_{bg} in stage-2 alignment.
- w/o L_{bg} : Remove L_{bg} in both stage-1 and stage-2 alignment.

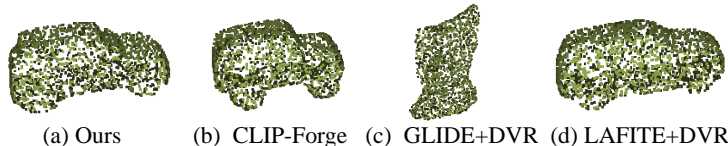
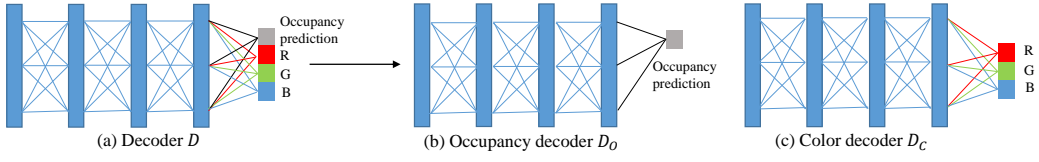


Figure 3: Visualization of point clouds of different methods for FPD evaluation.

Figure 4: Visualization to duplicate the decoder D to D_o , D_c .Table 1: Total scores s from the ten volunteers out of 52 generated shapes.

Method	1	2	3	4	5	6	7	8	9	10	mean \pm std	Consistency Score (%) \uparrow
CLIP-Forge	21	35	34	16.5	18.5	29	21.5	4	21	17	21.75 ± 9.16	41.83 ± 17.62
Dearm Fields	13	17.5	7	19	6.5	22	22	9	10	6	13.2 ± 6.41	25.38 ± 12.33
$E_I + D$	14	9.5	5	18	4.5	25	15.5	4.05	4.5	9	10.91 ± 7.06	20.97 ± 13.59
w/o stage-1	1.5	1	0	1.5	0	3.5	2	0	0	0.5	1.00 ± 1.15	1.92 ± 2.22
w/o stage-2	20	14.5	8.5	23.5	7.5	27	19	8.5	4.5	20.5	15.35 ± 7.73	29.52 ± 14.86
w/o L_{bg}	17.5	15.5	8.5	21	8	27	26.5	9.5	5	22.5	16.1 ± 8.06	30.96 ± 15.49
GLIDE+DVR	5	3.5	1.5	10	2	13	6.5	0.5	1	3	4.60 ± 4.13	8.85 ± 7.94
LAFITE+DVR	23	33.5	31	23	27	31.5	27.5	14.5	32.5	27.5	27.10 ± 5.75	52.12 ± 11.05
ISS (ours)	32	37	35.5	29.5	29	37	29	17.5	32.5	33	31.20 ± 5.69	60.00 ± 10.94

- GLIDE+DVR: Use a recent zero-shot text-guided image generation approach GLIDE (Nichol et al., 2021) to generate image I from T , then use DVR (Niemeyer et al., 2020) to generate S from I .
- LAFITE+DVR: Train a recent text-guided image generation approach LAFITE (Zhou et al., 2022) with ShapeNet images, then generate image I from T . Further generate S from I with DVR (Niemeyer et al., 2020).

3.2 QUANTITATIVE AND QUALITATIVE COMPARISONS

In this section, we analyze the results of the above baselines one by one.

- $E_I + D$: column (c) shows that the results generated from CLIP space Ω_I have inferior fidelity in terms of the texture (top row in Figure 5) and shape structure (bottom two rows in Figure 5) due to the limited capability of E_I to capture details of the image.
- w/o Stage 1: column (d) of Figure 5 indicates that without stage-1 alignment, the generated shapes are almost the same whatever text T is adopted as input, since M tends to map text feature f_T to almost the same feature even though stage-2 alignment is enabled. It shows that a good initialization provided by stage-1 alignment is necessary for the test-time optimization of stage 2.
- w/o Stage 2: as shown in column (e) of Figure 5, without Stage 2, may not align well with f_S due to the semantic gap between f_I and f_T . Now, we use Figure 6 (a) to illustrate their associated results: the model in “w/o stage 2” can generate reasonable shapes from a single image (see “SVR” in Figure 6 (a)) but fails with text as input (see “stage 1” in Figure 6 (a)); further with the stage-2 optimization, a plausible phone can be generated (see “stage 2 (ours)” in Figure 6 (a)). w/o L_{bg_1} , w/o L_{bg_2} , w/o L_{bg} : stage-2 alignment cannot work properly without L_{bg} in either stage-1 or stage-2 alignment or both (see column (f, g, h) of Figure 5) due to the lack of foreground awareness. Even though stage-1 alignment has encouraged the background to be white, we still need this loss in stage 2 to get satisfying results.

Table 2: A/B/C Test results of the ten volunteers. The numbers in the table indicate the number of shapes from the corresponding method he/she likes most out of the three candidates. Volunteers can optionally select “pass” instead of “A/B/C” if he cannot decide which one is the best.

Category	Method	1	2	3	4	5	6	7	8	9	10	mean \pm std \uparrow
Existing works	CLIP-Forge	9	17	12	13	6	9	3	6	6	8	8.9 ± 4.12
SOTA Text2Image+SVR	LAFITE+DVR	9	16	9	12	7	13	9	20	8	14	11.7 ± 4.11
Ours	ISS	27	19	21	20	17	25	19	26	13	30	21.7 ± 5.19

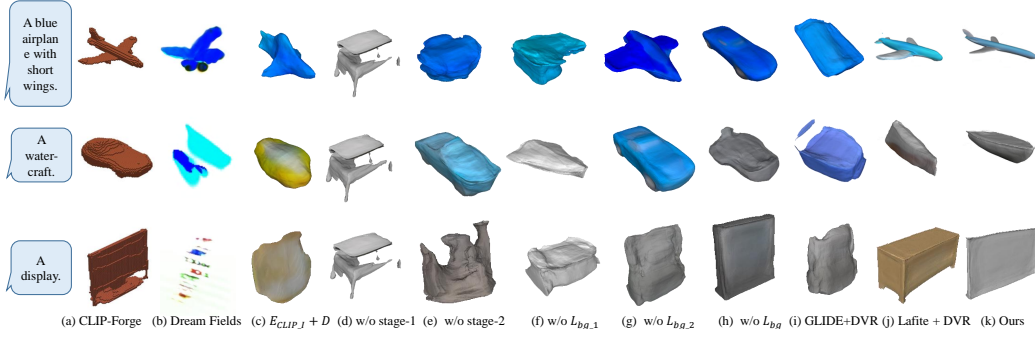


Figure 5: Additional qualitative results compared with existing works and baselines.

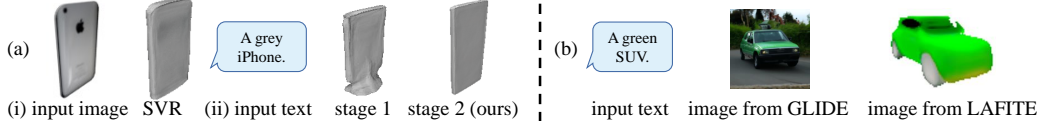


Figure 6: A further investigation on baselines “w/o stage 2”, “GLIDE+DVR”, and “LAFITE+DVR”. (a) “w/o stage 2” produces a plausible shape (“SVR”) from image but a low-quality shape (“stage 1”) from text; further fine-tuning using stage-2 alignment enables us to produce a more plausible shape from text (“stage 2 (ours)”). (b) GLIDE / LAFITE generate out-of-domain and inferior-quality images, limiting the performance of subsequent 3D generation.

- GLIDE+DVR: the images created by GLIDE (Nichol et al., 2021) have a large domain gap from the training data of DVR (see Figure 6 (b)), severely limiting the performance of GLIDE+DVR (see Figure 5 (i)).
- LAFITE+DVR: as shown in Figure 5 (j), some generative results of LAFITE+DVR can be coarse (first row of the main paper and the last row of Figure 5 in this supplementary material) due to the error accumulation of the isolated two steps, *i.e.*, LAFITE (see Figure 6 (b) “image from LAFITE”) and DVR, and some do not match the input text due to the semantic gap between f_I and f_T (two bottom rows of Figure 5 in the main paper and the last row of Figure 5 in this supplementary material). Despite the above, subsequently generating images then shapes is still a strong baseline that is a valuable research direction in the future.
- ours (ISS): column (k) of Figure 5 in the main paper and Figure 5 in this supplementary material show that our approach can generate shapes and textures with good text-shape consistency (see “small wings” in the top row, “water craft” in the middle row of Figure 5) and fidelity, beyond all the above baselines and the existing works CLIP-Forge (Sanghi et al., 2022) and Dream Field (Jain et al., 2022).

3.3 A/B/C TEST

To further compare our approach with the strongest baselines CLIP-Forge (Sanghi et al., 2022) and “LAFITE+DVR”, we perform an A/B/C test with 10 volunteers to compare these two baselines with ours. Specifically, the results from the three approaches (per input text) in random order for all the 52 texts. Then, they were instructed to choose a most preferred one. The results in Table 1 “A/B/C Test” in the main paper show that our results are more preferred than others, outperforming (Sanghi et al., 2022) by 143.8% $((21.70 - 8.90)/8.90)$ and “LAFITE+DVR” by 85.5% $((21.70 - 11.70)/11.70)$.

3.4 DIVERSIFIED GENERATION

In addition, we evaluate the diversified generation results discussed in Section 3.3 in the main paper. Specifically, we generate additional two samples per input text, and then adopt FID (Heusel et al., 2017), FPD (Liu et al., 2022) (the lower, the better) for the fidelity and diversity evaluation. The results are: **FID: 113.98**, **FPD: 35.37**, which is even better than our one-text-one-shape generative results (FID: 124.42 ± 5.11 , FPD: 35.67 ± 1.0), manifesting the superior performance of diversified generation capability of ISS.

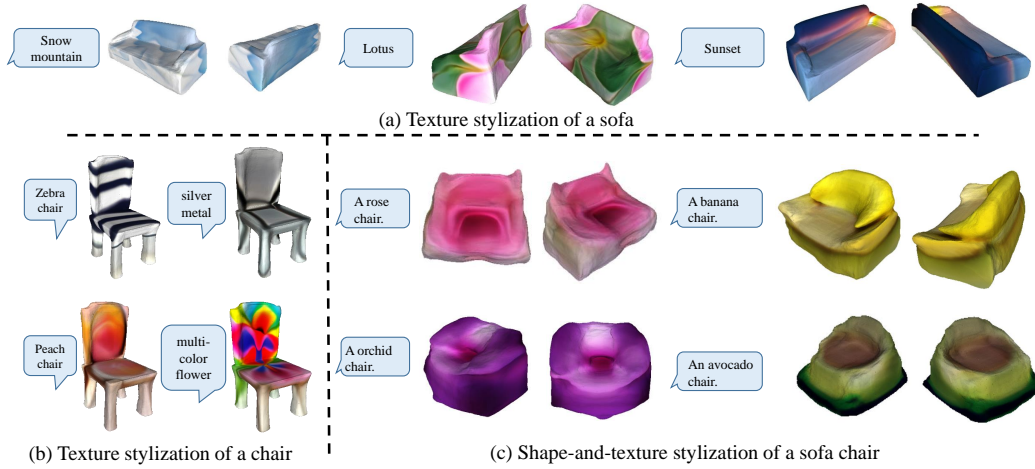
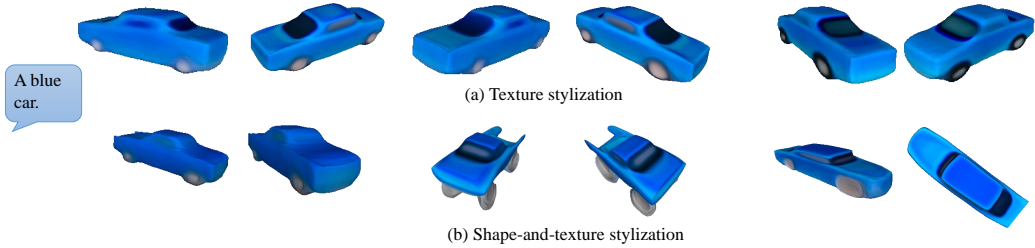


Figure 7: Additional stylization results.

Figure 8: Results of shape-and-texture stylization with the same text “A blue car” and different 3D prior loss L_P .

4 DISCUSSIONS ON TEXT-GUIDED SHAPE STYLIZATION

Additional stylization results. As shown in Figure 7 (a), our text-guided shape stylization is able to generate vivid landscape and flower textures on the sofa shape. Note that the sofa is generated from the text “A sofa with black backrest” (see Figure 9 top right) with totally different initial color from the stylization results. Further, in Figure 7 (b), we show four additional texture stylization results in addition to the Figure 9 in the main paper. In addition, as shown in Figure 7 (c), our shape-and-texture stylization is able to generate novel shapes and textures beyond the dataset, and create imaginary shapes with diversified structures. Note that our results achieve a good balance on the stylization and the functionality. For example, our result of “avocado chair” possesses both the style of “avocado” and the functionality of “chair”.

As shown in Figure 8, shape-and-texture stylization consistently produces the cars that is consistent with the input text “a blue car”, with proper variation in terms of the color and shape. In addition, the degree of the shape variations can be controlled by the loss weight λ_P of the 3D prior loss L_P .

Why does shape-and-texture stylization need an initial shape? Shape-and-texture stylization is initialized by the two-stage feature-space alignment result, since it provides the 3D prior.

Relationship of texture stylization and shape-and-texture stylization. Texture stylization and shape-and-texture stylization have their own merits. Texture stylization keeps the shape unchanged and is able to guarantee the functionality of the shape. In addition, it can take some abstract text descriptions as input like “sunset” in Figure 7 (a). Shape-and-texture stylization is able to generate novel and imaginary structures beyond the training dataset. However, there is a tradeoff between the stylization and the functionality.

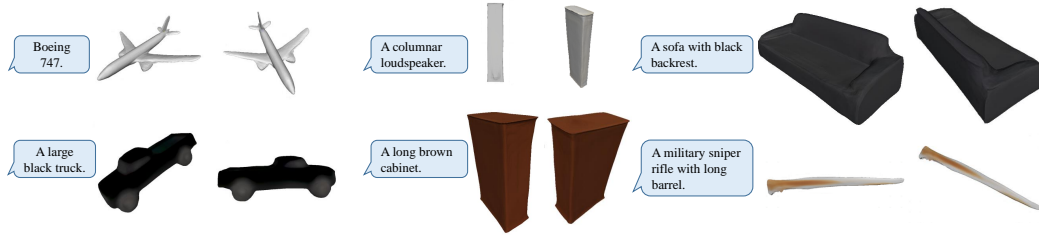


Figure 9: Additional generative results on the ShapeNet (Chang et al., 2015) dataset.

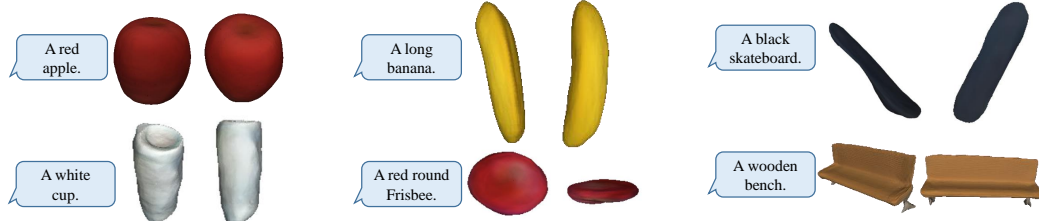


Figure 10: Additional generative results on the CO3D (Reizenstein et al., 2021) dataset.

5 ADDITIONAL RESULTS

ShapeNet. As shown in Figure 9, our approach can generate view-consistent 3D shapes on ShapeNet (Chang et al., 2015) that well match the input texts.

CO3D. Further, we show more text-guided shape generation results on the CO3D (Reizenstein et al., 2021) dataset in Figure 10. These results again manifest the capability of our approach on real-world 3D shape generation, beyond the existing works (Chen et al., 2018; Sanghi et al., 2022; Liu et al., 2022) that focus only on the synthetic shape generation on ShapeNet (Chang et al., 2015).

Working with GET3D When working with GET3D (Gao et al., 2022), our approach can generate 3D shapes of good fidelity from texts, as shown in Figure 11 in this supplementary file.

Single-image categories. In Figure 12, we present more text-guided generations for more categories using single images in training without camera pose, built upon Alwala et al. (2022). The results further demonstrate the compatibility of our approach to various SVR approaches, particularly generating plausible 3D shapes from text with single images in training.

More generative results. Further, we present more generative results of our approach in Figure 13. Using ISS, we are able to effectively generate a wide variety of 3D shapes from texts.

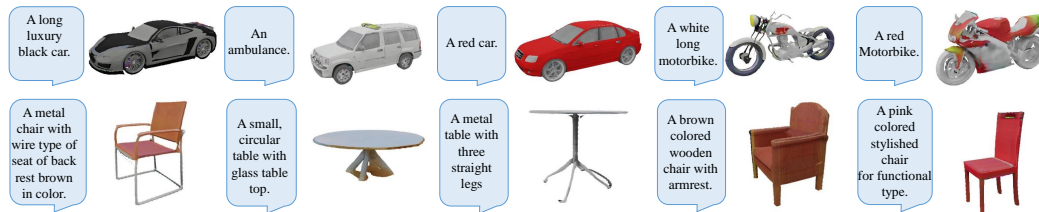


Figure 11: Additional generative results built upon GET3D (Gao et al., 2022).

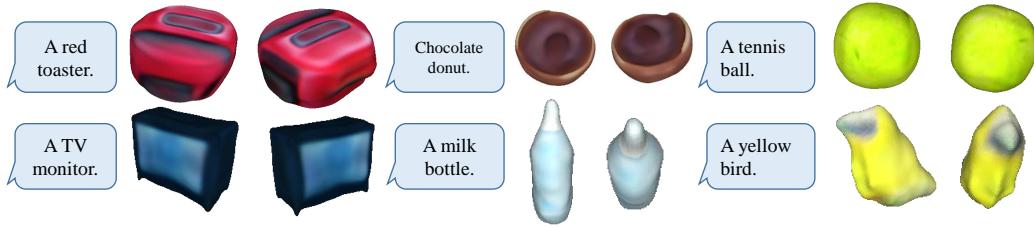


Figure 12: Additional generative results built upon SS3D (Alwala et al., 2022) using single images in training without camera pose.

6 DISCUSSIONS ON TWO ALTERNATIVE TRAINING STRATEGIES

Fine-tune decoder. In this section, we discuss an alternative training strategy to optimize the decoder directly with the CLIP consistency loss, instead of using two-stage feature space alignment.

First, we provide the results of the above training strategy. It produces unsatisfactory 3D shapes such as the one shown in Figure 14 (c) with more than 20 minutes of training. Besides, Dream Field (Jain et al., 2022) also utilizes the same idea to optimize the decoder directly; yet, the generated shape (see Figure 14 (d)) is also far from satisfactory even though Dream Fields is trained only to produce multi-view images with an NeRF-like architecture, unlike ours that is capable for 3D shape generation.

Then we try to analyze why the above strategy fails to produce desired shapes. Leveraging the pre-trained decoder that has already incorporated the 3D shape prior, our approach is able to search for the desired shape in the shape feature space Ω_S efficiently. On the contrary, fine-tuning the decoder in stage 2 without the explicit 2D/3D supervision will destroy the pre-trained shape feature space Ω_S , which is used to introduce 3D priors. Specifically, in stage 2 (“alignment stage”), the model is trained at test time with the user-provided text without any explicit 2D/3D supervision, so it is hard for the model to gain the knowledge about what the desired 3D shape should be like. In other words, it is extremely challenging for the model to learn the 3D shape prior with only CLIP consistency supervision from text.

Further, we analyze why the decoder can be fine-tuned in shape-and-texture stylization. First, note that 3D shape generation is a very different task from shape-and-texture stylization, since shape-and-texture stylization is initialized by the two-stage feature-space alignment result, which has already learned the 3D prior. Therefore, we can fine-tune the decoder in shape-and-texture stylization.

Update the feature in the shape space. In this section, we discuss another alternative training strategy to optimize the feature in shape space Ω_S , instead of optimizing the mapper M with stage-2 feature space alignment.

However, the shape feature has only 256 dimensions, which is far smaller than the number of weights in the mapping network. Hence, directly optimizing the shape feature in Ω_S is not capable for generating the desired 3D shape as shown in Figure 15, and the stage-2 feature space alignment is necessary in our approach.

7 ANALYSIS ON FEATURE SPACE MAPPING

To provide more insights on explaining how the latent space is mapped, we measure the distance of features at different stages on all the samples in our test set on ShapeNet in Table 3. The notations follow Figure 3 (c) of the main paper, M means the mapper, and d means cosine distance. Our ultimate goal is to obtain a text mapper M' (Figure 2 in the main paper) to map the text feature space f_T to shape feature space f_S using image with features f_I as a stepping stone to gradually narrow their distances using two stage mapping. Note that the image f_I and text features f_T are obtained using pre-trained CLIP models.

In the stage-1 alignment process, we train a mapper M to map image features f_I to a space $M(f_I)$ close to the shape space f_S using image data and the regression loss L_M . Note that the text feature f_T and image feature f_I are all from the CLIP model in a shared embedding space. It’s natural that the



Figure 13: Generative results of ISS. Using our new approach, we are able to effectively produce 3D shapes for a wide variety of categories from texts.

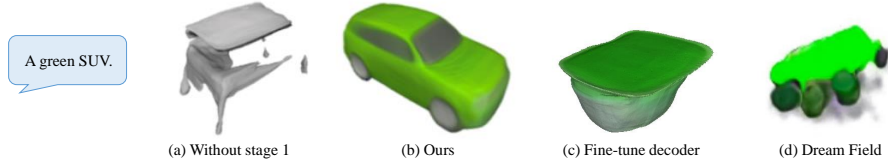


Figure 14: Results of optimizing the decoder, instead of the two-stage feature-space alignment.

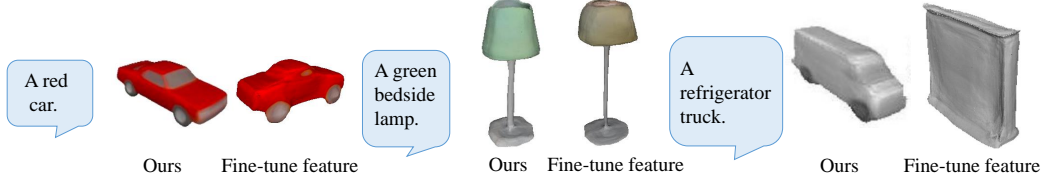


Figure 15: Results of optimizing the feature in shape space, instead of the stage-2 feature-space alignment.

trained mapper can be used to map the text feature f_T to $M(f_T)$, making text features closer to the shape space. However, when measuring the distance among $M(f_T)$, f_S , we find the average distance among all samples is $d(M(f_T), M(f_T)) = 0.58 \pm 0.23$, the average distance between $M(f_T)$ and f_S is 0.21 ± 0.10 , and the average distance between shape and text is $d(M(f_T), f_S) = 0.45 \pm 0.20$. This implies that there is a gap between CLIP image and text feature after the first step mapper and further motivates our stage-2 alignment. Note that there is no GT shape for our task on 3D generation, so we manually select a shape in the ShapeNet dataset that matches the input text as the GT.

In the stage-2 alignment, M is further updated and the final delivered mapper is called M' , which is to further narrow down the gap between mapped text features and shape features. The average distance between the mapped text feature $M'(f_T)$ and shape feature space f_S : $d(M'(f_T), f_S) = 0.17 \pm 0.08$ which is much smaller than the corresponding distance after stage-1 alignment. It shows that stage 2 alignment can significantly reduce the difference between the mapped text and the GT shape feature from 0.45 to 0.17 on average.

8 RELATED WORKS ON SINGLE-VIEW RECONSTRUCTION AND DIFFERENTIABLE RENDERING

Single-view reconstruction (SVR) This task aims to reconstruct the 3D shape from a single-view image of it. Recently, many approaches have been proposed for meshes (Agarwal & Gopi, 2020), voxels (Zubić & Liò, 2021), and 3D shapes (Niemeyer et al., 2020). Recently, to extend SVR to in-the-wild categories, Alwala et al. (2022) proposed a new approach called SS3D to learn 3D shape reconstruction using single-view images for the reconstruction of hundreds of categories.

As 3D-to-2D projections, single-view 2D images are more closely related to shapes than texts, since they reveal many attributes of 3D shapes, e.g., structure, details, appearance, etc. The strong correlation between 2D images and 3D shapes motivates us to reduce the challenging text-to-shape generation task to text-to-image and then SVR, by connecting the CLIP features from text with the shape features in SVR using images as an intermediate step to gradually bridge the gap between text and shape. Specifically, we extend the pre-trained SVR model to be compatible with text input, transforming the challenging text-to-shape task into SVR. Our framework can work with different SVR approaches to extend them for 3D shape generation from texts. So, our approach is orthogonal to the SVR approaches.

Differentiable Rendering 3D rendering is an important topic in computer vision and graphics. It takes a 3D scene as input and predicts the 2D view of it given a camera pose. Beyond 3D rendering, differentiable rendering further aims to derive the differentiations of the rendering function. With differentiable rendering, a renderer can be integrated into an optimization framework, thus a 3D shape can be reconstructed from multi-view 2D images. Neural Volume Rendering (Mildenhall et al., 2020) and its following works (Jain et al., 2021; Barron et al., 2021) aim to synthesize novel view images of a 3D scene. Besides, recent works (Niemeyer et al., 2020; Munkberg et al., 2022; Gao

Table 3: Distance changes in the feature space mapping of all the 52 samples in the test set. d means cosine distance. Almost all distances are consistently reduced after our stage-2 alignment.

Text	$d(M(f_I), M(f_T))$	$d(M(f_I), f_S)$	$d(M(f_T), f_S)$	$d(M'(f_T), f_S)$	$d(M(f_T), M'(f_T))$
a glass single leg circular table	0.63	0.31	0.58	0.14	0.34
a wooden double layers table	0.72	0.10	0.64	0.14	0.65
a square metal table	0.76	0.32	0.44	0.21	0.30
a round shaped single legged wooden table	0.47	0.24	0.21	0.21	0.30
this is a bar stool with metal arches as a design feature	0.43	0.33	0.35	0.18	0.17
a children chair with little legs	0.79	0.20	0.43	0.12	0.35
a swivel chair with wheels	0.61	0.30	0.38	0.22	0.11
a red recliner seems comfortable	0.33	0.20	0.23	0.24	0.05
a red car	1.02	0.19	0.78	0.10	0.69
a green SUV	0.49	0.12	0.24	0.19	0.25
a large black truck	0.74	0.19	0.50	0.15	0.53
a long luxury black car	0.67	0.22	0.54	0.09	0.54
army fighter jet	0.43	0.54	0.43	0.25	0.28
a black airplane with long white wings	0.62	0.12	0.32	0.16	0.10
a blue airplane with short wings	0.51	0.14	0.55	0.33	0.40
boeing 747	0.35	0.13	0.23	0.06	0.20
a big ship for transportation	0.95	0.17	0.72	0.35	0.21
a boat with sail	0.39	0.19	0.46	0.12	0.45
a watercraft	0.12	0.14	0.27	0.22	0.06
a wooden boat	0.76	0.29	0.52	0.12	0.49
a blue sofa	0.46	0.20	0.34	0.12	0.27
sofa with legs	0.67	0.13	0.55	0.10	0.35
a sofa with black backrest	0.59	0.14	0.33	0.08	0.16
a small sofa	0.49	0.20	0.33	0.10	0.28
a long brown bench	0.57	0.29	0.34	0.18	0.27
a marble bench	0.61	0.16	0.46	0.12	0.29
a metal bench	0.18	0.29	0.20	0.20	0.09
concrete bench	0.48	0.07	0.46	0.11	0.36
a military sniper rifle with long barrel	0.47	0.14	0.29	0.07	0.21
a rifle with magazines	0.71	0.48	0.51	0.17	0.43
a short rifle	0.75	0.15	0.81	0.35	0.52
rifle shotgun	0.38	0.13	0.25	0.10	0.09
a computer monitor	0.28	0.07	0.22	0.07	0.20
a display	0.22	0.18	0.15	0.13	0.02
a monitor with square base	0.56	0.17	0.75	0.33	0.34
a TV monitor	0.70	0.16	0.41	0.11	0.27
a cabinet with cylindrical legs	0.53	0.24	0.27	0.18	0.11
a cupboard	0.54	0.37	0.42	0.19	0.45
a long brown cabinet	0.57	0.29	0.34	0.18	0.27
a wardrobe	0.41	0.13	0.32	0.16	0.26
a desk lamp	0.92	0.35	0.51	0.16	0.45
bedside lamp	0.71	0.48	0.55	0.17	0.43
lamp supported by a long pillar	0.19	0.12	0.18	0.05	0.14
mushroom-like lamp	1.20	0.17	1.06	0.32	0.46
a mobile phone	0.75	0.09	0.85	0.28	0.55
a small cell phone	0.49	0.20	0.33	0.10	0.28
a mobile phone with black screen	0.94	0.39	0.61	0.14	0.68
an iphone	0.75	0.25	0.54	0.22	0.57
a columnar loudspeaker	0.66	0.25	0.54	0.15	0.40
a loudspeaker with metal surface	0.34	0.17	0.44	0.16	0.30
a wooden loudspeaker	0.72	0.10	0.64	0.14	0.65
a cylindrical loudspeaker	1.06	0.19	0.82	0.31	0.27
mean \pm std	0.58 \pm 0.234	0.21 \pm 0.10	0.45 \pm 0.20	0.17 \pm 0.08	0.32 \pm 0.17

Table 4: Number of parameters and performance of existing works and our method.

Method	Number of parameters (M)	FID	FPD	Consistency Score
CLIP-Forge (Sanghi et al., 2022)	Normalized flow network: 18.37	162.87	37.43	41.83±17.62
Dream Fields (Jain et al., 2022)	0.61	181.25	N.A.	25.38±12.33
Ours	Mapper: 2.43	124.42±5.11	35.67±1.09	60.0±10.94
Ours with a lightweight mapper	Mapper: 0.46	129.01	34.26	67.21±10.64

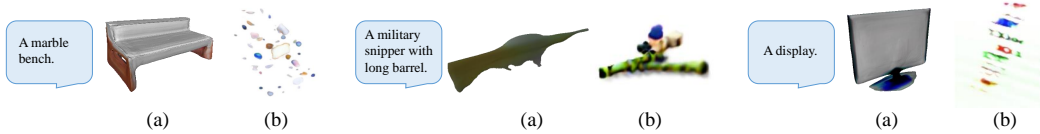


Figure 16: (a) Our generative result with lightweight mapper vs. (b) Dream Fields (Jain et al., 2022).

et al., 2022) leverage differentiable rendering for 3D shape generation using 2D images. In this work, we derive 2D images of the generated 3D shape using differentiable rendering and use a pre-trained large-scale image-language model CLIP to encourage the 2D images to be consistent with the input text. Thanks to differentiable rendering, we can update the generated 3D shape indirectly using the rendered images.

9 ANALYSIS ON THE NUMBER OF PARAMETERS

Our model incorporates only a small number of learnable parameters for shape generation in M . Note that the parameters of the SVR model are not tuned for the specific text-to-shape generation task. Therefore, they are not counted in the total number of parameters.

We provide the comparison of learnable model parameters responsible for shape generation and compare it with existing methods in Table 4.

The number of learnable parameters of CLIP-forge (Sanghi et al., 2022) is 8 times larger than ours. Note that we do not include the parameters of the CLIP-Forge auto-encoder for a fair comparison. With much fewer parameters, our model outperforms CLIP-Forge in all evaluated metrics (see Table 4). This demonstrates that our performance gain is not purely from the learnable parameters.

To better compare with Dream Fields (Jain et al., 2022) (0.61 M parameters), we design a lightweight mapper to match the total number of parameters. Specifically, the mapper is composed of three fully connected layers with 512, 256, and 256 output dimensions, yielding a total of 0.46M parameters which is smaller than Dream Field. With this new mapper, we still outperform Dream Fields in terms of metrics. Please see Figure 16 for quantitative comparison. This further demonstrates our major performance gain is not from the learnable parameters.

From the results in Table 16, we can consistently observe that our model can achieve much better performance with much fewer parameters, manifesting the efficiency of our proposed image as a stepping-stone pipeline that allows us to leverage the 3D priors in pre-trained SVR models to enable text to 3D shape synthesis without requiring paired text and 3D data.

Note that we exclude the number of parameters of the decoder D in our comparison because the only effect of fine-tuning D is to make the background white and does not contribute to the generative capability of our model. As shown in Figure 2 of the main paper, mapper M and decoder D are trained with their own losses **separately** at the same time by stopping the gradients from L_D and L_{bg} to propagate to M .

Also, to show that fine-tuning D does not improve its generative capability, we feed the same input feature f_s to D before and after fine-tuning, they generate almost the exact same 3D shape as the original SVR model, as shown in Figure 17 in the supplementary material. And in stage 2, D is not optimized. Admittedly, our generative capability benefits from the SVR model including D . However, D is not optimized for our text-to-shape generation task. Therefore, we exclude the number of parameters of decoder D for fair comparison.

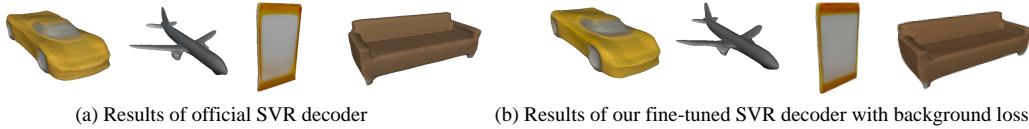


Figure 17: The generative results of the original SVR model and after our stage-1 feature space alignment using the same feature f_S .

10 FAILURE CASES

Here are some examples of failure cases of our approach.

The complex and unusual shapes, e.g., “an oval table with 3 legs.” First, our model fails to generate the shape from the text “an oval table with 3 legs.”. Our approach leverages the CLIP consistency loss in the rendered 2D image; however, in the rendered image shown in Figure 18 (a), only three legs can be seen and the remaining one is occluded, which confuses the model training.

The given text is very long, e.g., “it is grey in color, circular in shape with four legs and back support, material used is wood and overall appearance looks like unique design armless chair.” Our model fails to generate the shape from the above long description as shown in Figure 18 (b). Some attributes are missing including “armless”, “four legs”. This is partially due to the limited representative capability of a single CLIP feature for such a long sentence. We may incorporate an additional local feature like Liu et al. (2022) to handle the long text in the future.

The shapes with multiple fine-grained descriptions, e.g., “A chair with a red back and a green cushion.” As shown in Figure 18 (c), our model fails to generate the shape “A chair with a red back and a green cushion.” As studied in some recent works (Yao et al., 2022; Li et al., 2022), CLIP mainly address on the global image feature, but has inferior capability to capture fine-grained features. Therefore, our approach may fail to generate shapes where multiple fine-grained descriptions are given. In the future, we may try more recent pre-trained text-and-image embedding models to enhance the model’s capability to handle the fine-grained descriptions.

11 LIMITATIONS

This work still has some limitations. First, our performance is limited by the SVR model that our approach is built upon, e.g., some results in Figure 10 of the main paper and Figure 12 in this supplementary material are still not very satisfactory, because SS3D (Alwala et al., 2022) itself is struggling to create shapes with fine details. Second, we cannot generate the categories outside the image dataset due to the lack of 3D prior of the unseen category. That is why our model needs images as the stepping stone to learn what the particular category is like. However, we want to highlight the following. First, built upon SS3D (Alwala et al., 2022), our approach can generate a wide range of categories with single-view images in the wild as training data. Second, with our shape-and-texture stylization, our approach can generate imaginary and uncommon shapes outside the image dataset (in the same category). Third, it is extremely challenging to generate arbitrary category shapes from text. As far as we know, there is only one existing work, Dream Field (Jain et al., 2022), that can generate more categories than ours. However, Dream Fields only generate multi-view images instead of directly generate 3D shapes, and it cannot generate reasonable shapes in many cases as shown in Figure 5 in our main paper and Figure 3 in this supplementary material.

12 TEXT SET IN THE EXPERIMENTS

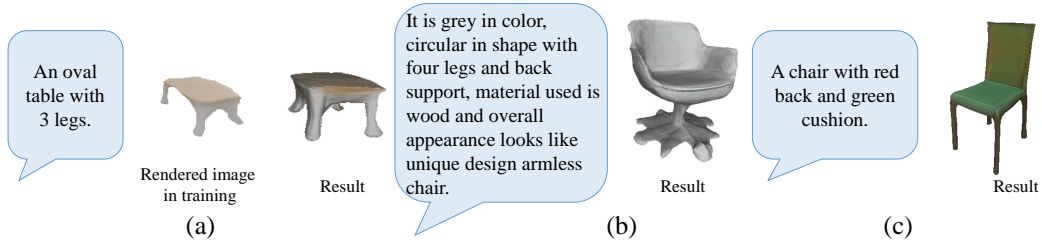


Figure 18: Failure cases of our approach.

Table 5: Summary of symbols used in paper.

Notation	Description	Notation	Description
E_S	encoder of SVR model	E_I	CLIP image encoder
E_T	CLIP text encoder	D	Decoder of SVR model
M	Mapper	M'	Mapper after stage-2 alignment
D_o	occupancy decoder	D_c	color decoder
R	rendered images	f_I	CLIP image feature
f_T	CLIP text feature	f_S	shape feature in SVR model
o	camera center	p	query point
d	cosine distance	L_M	regression loss in stage 1
L_D	original loss in SVR model	L_{bg}	background loss
L_{bg_1}	background loss in stage 1	L_{bg_2}	background loss in stage 2
L_C	CLIP consistency loss	L_P	3D prior loss
Ω_T	CLIP text feature space	Ω_S	shape feature space from the SVR model
Ω_I	CLIP image feature space		

Recent works (Chen et al., 2018; Sanghi et al., 2022; Jain et al., 2022) proposed their own text sets. However, their datasets have some limitations and are not suitable to evaluate our approach. The dataset of Text2shape (Chen et al., 2018) contains text descriptions in only two categories, *i.e.*, Table and Chair; CLIP-Forge (Sanghi et al., 2022) lacks of descriptions on the color and texture; while Dream Fields (Jain et al., 2022) utilizes text descriptions containing complex scenes and actions. To fairly evaluate our approach, we propose two text datasets on the ShapeNet (Chang et al., 2015) and CO3D (Reizenstein et al., 2021) categories, respectively, shown in Tables 6 and 7.

13 LIST OF NOTATIONS

In this section, we summarize symbols and notations used in the paper to facilitate readers to follow up. Please refer to Table 5 for more details.

REFERENCES

- Nitin Agarwal and M Gopi. Gamesh: Guided and augmented meshing for deep point networks. In *3DV*, 2020.
- Kalyan Vasudev Alwala, Abhinav Gupta, and Shubham Tulsiani. Pre-train, self-train, distill: A simple recipe for supersizing 3D reconstruction. *CVPR*, 2022.
- Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021.
- Angel X. Chang, Thomas Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], 2015.

Table 6: Texts on ShapeNet (Chang et al., 2015). They are utilized to measure FID (Table 1 of the main paper), and employed in Human Perceptual Evaluation (Table 1) and A/B/C Test (Table 2).

a glass single leg circular table	a wooden double layers table
a square metal table	a round shaped single legged wooden table
this is a bar stool with metal arches as a design feature	a children chair with little legs
a swivel chair with wheels	a red recliner seems comfortable
a red car	a green SUV
a large black truck	a long luxury black car
army fighter jet	a black airplane with long white wings
a blue airplane with short wings	boeing 747
a big ship for transportation	a boat with sail
a watercraft	a wooden boat
a blue sofa	sofa with legs
a sofa with black backrest	a small sofa
a long brown bench	a marble bench
a metal bench	concrete bench
a military sniper rifle with long barrel	a rifle with magazines
a short rifle	rifle shotgun
a computer monitor	a display
a monitor with square base	a TV monitor
a cabinet with cylindrical legs	a cupboard
a long brown cabinet	a wardrobe
a desk lamp	bedside lamp
lamp supported by a long pillar	mushroom-like lamp
a mobile phone	a small cell phone
a mobile phone with black screen	an iphone
a columnar loudspeaker	a loudspeaker with metal surface
a wooden loudspeaker	a cylindrical loudspeaker

Table 7: Texts on CO3D (Reizenstein et al., 2021).

A big apple	A red apple	A green bottle	A tall cylindrical bottle
A white cup	A wooden cup	A large black microwave	A white cuboid microwave
A black skateboard	A green long skateboard	A cute toytruck	A large toy truck
A blue backpack	A red big backpack	A white bowl	A big wooden bowl
A red round frisbee	A blue large frisbee	A big blue motorcycle	A black large wheels motorcycle
A circular stop sign	A triangle stop sign	Tv screen	A grey big tv screen
A basketball	A tennis ball	A large broccoli	A green broccoli
A hairdryer	A yellow hairdryer	A black mouse	A white mouse
A cuboid big suitcase	A large size tall suitcase	A round umbrella	A big black umbrella
A big banana	A long banana	A cream round cake	A chocolate mooncake
A blue handbag	A red big handbag	An orange	A large round orange
A teddybear	A cute teddybear	A blue fat vase	A blue tall vase
A black baseball bat	A long wooden baseball bat	A blue car	A red car
An egg hotdog	A sausage hotdog	A black parkingmeter	A white tall parkingmeter
A black toaster	A round toaster	Tall wineglass	Single leg big wineglass
A brown baseball glove	A black big baseball glove	A big carrot	A long carrot
A red hydrant	A yellow hydrant	A large round pizza	A tomato meat pizza
A white toilet	A fat white toilet	A stone bench	A wooden long bench
A gray iphone	A black phone	A long black keyboard	A short white keyboard
A short tree	A tall green tree	A toy bus	One decker toy bus
A blue bicycle	A black bicycle	A blue chair	A wooden chair
A red kite	A long blue kite	A TV remote	A long white remote
A book with blue cover	A black book	Brown couch	A long brown couch
A open laptop	A black laptop	An egg sandwich	A meat sandwich
A cute toy train	A short blue toy train	Chocolate donut	Big circular donut

- 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 *ACCV*, 2018.
- Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojic, and Sanja Fidler. GET3D: A generative model of high quality 3D textured shapes learned from images. *NeurIPS*, 2022.
- 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. *NIPS*, 2017.
- Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021.
- Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with Drefam Fields. In *CVPR*, 2022.
- Juncheng Li, Xin He, Longhui Wei, Long Qian, Linchao Zhu, Lingxi Xie, Yueting Zhuang, Qi Tian, and Siliang Tang. Fine-grained semantically aligned vision-language pre-training. *arXiv preprint arXiv:2208.02515*, 2022.
- Zhengzhe Liu, Yi Wang, Xiaojuan Qi, and Chi-Wing Fu. Towards implicit text-guided 3D shape generation. In *CVPR*, 2022.
- 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. In *ECCV*, 2020.
- Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3D models, materials, and lighting from images. In *CVPR*, 2022.
- 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.
- Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *CVPR*, 2020.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.
- Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *ICCV*, 2021.
- Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. CLIP-Forge: Towards zero-shot text-to-shape generation. In *CVPR*, 2022.
- Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *ICLR*, 2022.
- 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. In *CVPR*, 2022.
- Nikola Zubić and Pietro Liò. An effective loss function for generating 3D models from single 2D image without rendering. *arXiv preprint arXiv:2103.03390*, 2021.