CLIP-Sculptor: Zero-Shot Generation of High-Fidelity and Diverse Shapes from Natural Language

Aditya Sanghi[†] Rao Fu[‡] Vivian Liu[§] Karl D.D. Willis[†] Hooman Shayani[†]
Amir H. Khasahmadi[†] Srinath Sridhar[‡] Daniel Ritchie[‡]
Autodesk Research[†] Brown University[‡] Columbia University[§]

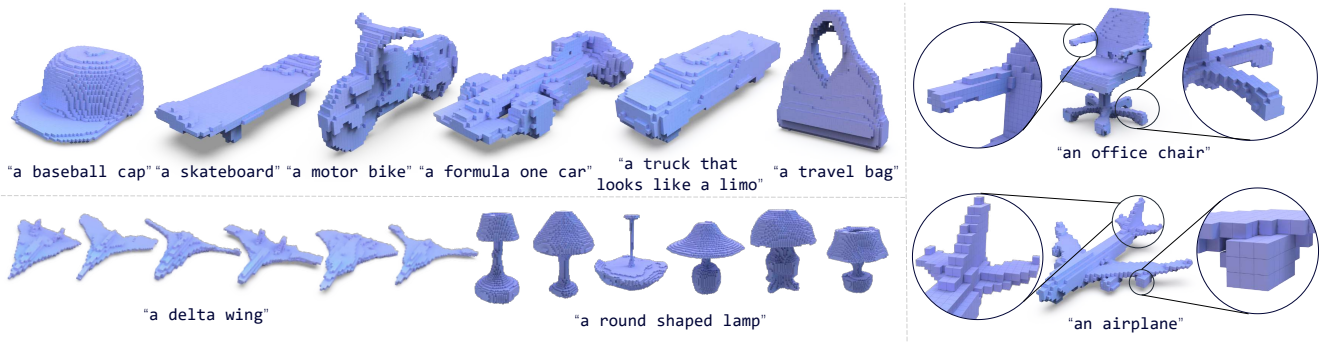

Figure 1. CLIP-Sculptor is a zero-shot text-to-shape generation method. Left Top/Bottom: It generates **diverse** shapes that reflect the semantic meaning of the text input **without requiring any text data** during training. Right: The method also generates **high-fidelity** shapes.

Abstract

Recent works have demonstrated that natural language can be used to generate and edit 3D shapes. However, these methods generate shapes with limited fidelity and diversity. We introduce CLIP-Sculptor, a method to address these constraints by producing high-fidelity and diverse 3D shapes without the need for (text, shape) pairs during training. CLIP-Sculptor achieves this in a multi-resolution approach that first generates in a low-dimensional latent space and then upscales to a higher resolution for improved shape fidelity. For improved shape diversity, we use a discrete latent space which is modeled using a transformer conditioned on CLIP's image-text embedding space. We also present a novel variant of classifier-free guidance, which improves the accuracy-diversity trade-off. Finally, we perform extensive experiments demonstrating that CLIP-Sculptor outperforms state-of-the-art baselines.

1. Introduction

In recent years, there has been rapid progress in text-conditioned image generation [31, 32, 35], which has been driven by advances in multimodal understanding learned from web-scaled paired (text, image) data. These advances have led to applications in domains ranging from content creation [21, 22, 31] to human–robot interaction [40]. Unfortunately, developing the analogue of a text-conditioned 3D shape generator is challenging because it is difficult to obtain (text, 3D shape) pairs at large scale. Prior work has attempted to address this problem by collecting text-shape paired data [2, 5, 9, 23, 26], but these approaches have been limited to a small number of object categories.

A promising way around this data bottleneck is to use weak supervision from large-scale vision/language models such as CLIP [30]. One approach is to directly optimize a 3D representation such that (text, image render) pairs are aligned when projected into the CLIP embedding space. Prior work has applied this approach to stylize 3D meshes [25, 41] and to create abstract "dreamlike" objects using neural radiance fields [18] or meshes [19]. However, neither of the aforementioned methods produce realistic object geometry, and they can require expensive optimization. Another approach, more in line with text-to-image generators [31, 32], is to train a conditional generative model. The CLIP-Forge system [37] builds such a model without paired (text, shape) data by using rendered images of shapes at training time and leveraging the CLIP embedding space to bridge the modalities of image and text at inference time. CLIP-Forge demonstrates compelling zero-shot generation

Table 1. High-level comparison between zero-shot text-to-shape generation methods. Inference time is calculated using a single NVIDIA Tesla V100 GPU.

Method	**Inference**	**Fidelity**	**Diversity**
DreamFields [18]	$>$ 24 hrs	Low	Single
Clip-Mesh [19]	30 min	Low	Single
Clip-Forge [37]	6.41 ms	Medium	Medium
CLIP-Sculptor	0.91 sec	High	High

abilities but produces low-fidelity shapes which do not capture the full diversity of shapes found in the training distribution.

In this paper, we propose CLIP-Sculptor, a text-conditioned 3D shape generative model that outperforms the state of the art by improving shape diversity and fidelity with only (image, shape) pairs as supervision. CLIP-Sculptor's novelty lies in its multi-resolution, voxel-based conditional generation scheme. Without (text, shape) pairs, CLIP-Sculptor learns to generate 3D shapes of common object categories from the text-image joint embedding of CLIP. To achieve high fidelity outputs, CLIP-Sculptor adopts a multi-resolution approach: it first generates a low-resolution latent grid representation that captures the semantics from text/image, then upscales it to a higher resolution latent grid representation with a super-resolution model, and finally decodes the output geometry. To generate diverse shapes, CLIP-Sculptor adopts discrete latent representations which are obtained using a vector quantization scheme that avoids posterior collapse. To further improve shape fidelity and diversity, CLIP-Sculptor uses a masked transformer architecture. We additionally propose a novel annealed strategy for the classifier-free guidance [16]. To sum up, we make the following contributions:

- CLIP-Sculptor, a multi-resolution text-conditional shape generative model that achieves both high fidelity and diversity without the need for (text, shape) pairs.
- A novel variant of classifier-free guidance for generative models, with an annealed guidance schedule to achieve better quality for a given diversity level.

2. Related Work

Neural Discrete Representation. Discrete representations [28] were first proposed for image generation in the context of variational autoencoders as a method to improve image quality and avoid posterior collapse. Further work introduced multi-scale and hierarchical variants [8, 33] which improved generative capabilities. Recently, 3D discrete representations have emerged as well, such as discretized voxel and implicit grids [26, 42]. In this work, we take inspiration from hierarchical VQ-VAEs [8, 33] and propose an architecture of hierarchical discrete representations capable of generating high fidelity 3D shapes.

Latent Generative Models. In recent years, latent generative models have been widely adopted, because these models can effectively generate low-dimensional latent representations which can be used to generate high fidelity images and shapes. In the 3D domain, GAN-based latent models [1, 6, 17] have shown impressive results but tend to suffer from training instability and mode collapse. Flow-based models [37, 43] have been proposed to alleviate these problems, but they yield sample quality that is inferior to GAN-based models. Recent works in the image domain use diffusion [12, 36] or masking models [4] on the latent space which can increase the inference efficiency and still give quality outputs. Building on these works, we propose a hierarchical latent generative model that can further improve the fidelity and diversity of shape generations.

Classifier-free Guidance. Classifier-free guidance, proposed by [16], jointly trains a conditional and unconditional generative model, whose score estimates are mixed to establish a trade-off between diversity measured by Frechet inception distance (FID) and sample accuracy/fidelity measured by Inception score (IS). Classifier-free guidance guides unconditional samples in the direction of conditional ones and has been implemented in practice to achieve state of the art results in recent work [10–12, 34]. To the best of our knowledge, guidance scale has always been kept constant outside the concurrent exploration in [15]. Their oscillating guidance technique, however does not show improvements in sample fidelity and produces more artifacts in their generations. In this work, we find that rather using annealed scheduling can give better quality versus diversity trade-offs.

Text-to-Shape Generation. Text-to-shape generation has gained momentum in recent years. Recent works [5, 9, 23, 26] use supervised text-shape pairs to generate shapes effectively using natural language. However, a major drawback is the lack of available text-shape datasets, which limits supervised methods to generate shapes only in few categories. To solve this, several recent works have successfully leveraged the image-text embeddings of CLIP as a prior [30] by converting shapes into images using renders. One line of work uses differentiable renderers [18, 19, 25, 29, 41]. Another line of work learns a mapping from the image space to the shape space [37]. Although these methods are effective, they suffer from either long optimization time [18, 19, 25, 29] or limited quality of generated shapes [18, 19, 37]. Our method is able to generate more diverse shapes of higher fidelity within a comparably short inference time.

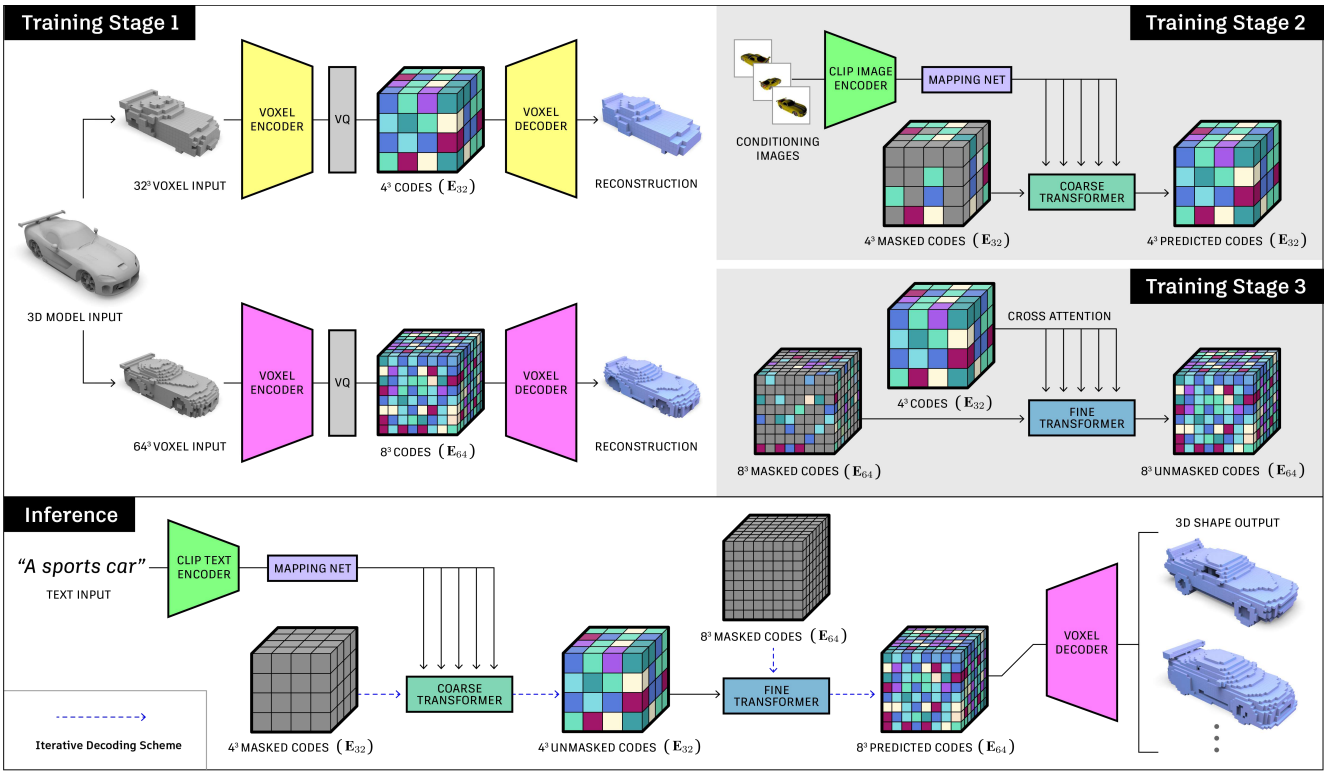

Figure 2. The CLIP-Sculptor architecture during training (top) and inference (bottom). CLIP-Sculptor is trained in three stages. In Stage 1, we train two separate VQ-VAE models for 32^3 and 64^3 resolution voxel grids. In Stage 2 we train a coarse transformer conditioned on a CLIP embedding to generate low resolution VQ-VAE latent grids $\mathbf{E}_{32}$. In Stage 3, we train a fine transformer to perform super resolution on these latent grids. During inference, a text prompt is passed through the CLIP text encoder and used to condition the coarse transformer to generate a coarse latent grid $\mathbf{E}_{32}$. This coarse grid is then used to condition the fine transformer to generate a fine latent grid $\mathbf{E}_{64}$. Finally, this fine latent grid is then passed through the Training Stage 1 64^3 VQ-VAE decoder to generate the output shape.

3. Method

CLIP-Sculptor aims to generate diverse and high-fidelity 3D shapes of common object categories from text prompts. Figure 2 illustrates the components of our CLIP-Sculptor model. We train the network using 3 stages. In the first stage, we use a vector-quantized representation [28], to efficiently represent 3D shapes in discrete voxel grids. This latent representation captures the diverse semantic meanings of the 3D shapes in common object categories and is compatible with powerful generative models. To balance computation cost and shape quality, we utilize latent representations at two resolutions. A low resolution latent representation $\mathbf{E}_{32}$ encodes geometric information of voxels $\mathbf{V}_{32}$ at resolutions of 32^3, which is useful for text-conditional generation. A high resolution latent representation $\mathbf{E}_{64}$ encodes geometric information of voxel grids $\mathbf{V}_{64}$ at resolutions of 64^3, which is useful for super-resolution and detail generation. In stage 2, we train a generative model conditioned on CLIP features obtained from images $\{\mathbf{I}_r | r \in R\}$ rendered at random views R to solve the lack of paired (text, shape) data. The conditional generative model consists of a mapping net and a *coarse* transformer $T_c(.)$ that is trained to unmask a masked low-resolution grid $\mathbf{E}_{32}$. Finally in stage 3, a *fine* transformer $T_f(.)$ is used as a super-resolution model in latent space to generate higher resolution shapes. It is trained in a similar manner to the *coarse* transformer but conditioned on a low-resolution grid $\mathbf{E}_{32}$ instead of CLIP features.

During inference, a text prompt is sequentially converted from CLIP text embedding to a low-resolution latent representation $\mathbf{E}_{32}$ to a high-resolution latent representation $\mathbf{E}_{64}$ to finally a high-resolution voxel grid $\mathbf{V}_{64}$. A confidence-based iterative decoding scheme [4] is adopted for unmasking the fully masked initial grids for coarse and fine transformers at inference. To further control shape fidelity and diversity, we propose an annealing strategy for classifier-free guidance at inference time.

In the following subsections, we will introduce the major components within CLIP-Sculptor. Section 3.1 describes the vector quantized representations at multiple resolutions;

Section 3.2 explains the conditional generation model as well as its design choices and training strategies; Section 3.3 introduces the super-resolution model; Section 3.4 explains how the classifier-free guidance is achieved at inference using an annealing strategy.

3.1. Multi-Resolution Voxel VQ-VAEs

To learn a low-dimensional latent space that effectively represents the distribution of the training shape set D, we use a vector-quantized variational autoencoder (VQ-VAE) [28]. The vector quantized latent representation avoids "posterior collapse", and it is also amenable to modeling with transformer-based generative models. To balance computational cost and shape quality, we train two separate VQ-VAEs for 32^3 and 64^3 resolution voxel grids $\mathbf{V}_{32}$ and $\mathbf{V}_{64}$, which are useful for conditional generation and super-resolution respectively.

For both VQ-VAEs, we use ResNet-based [13] volumetric CNN encoders and decoders. A vector quantization layer maps the down-sampled voxel grids into a discrete latent space by indexing the continuous latent space with an embedding codebook. In this way, the voxel grids $\mathbf{V}_{32}$ and $\mathbf{V}_{64}$ are effectively represented by low-dimensional latent representations $\mathbf{E}_{32}$ and $\mathbf{E}_{64}$, at the resolution of 4^3 and 8^3. The VQ-VAEs are trained with mean squared error loss (MSE), commitment loss [28], and an exponential moving average strategy [33, 45].

3.2. CLIP-Conditioned Coarse Transformer

The conditional generation process aims to generate diverse shapes that semantically correspond to the text prompts. Inspired by the recent success of masking approaches [4] and diffusion models [12, 27, 36], we formulate this generation process as a conditional unmasking task. Given an input masked latent representation $\mathbf{E}_{32}^{msk}$ and a conditional vector $\mathbf{c}$, the coarse transformer $T_c(.)$ should produce an fully unmasked latent representation $\mathbf{E}_{32}$ that not only aligns with the semantic meaning captured by the conditional vector, but also allows for the sampling of diverse shapes at inference.

The conditional vector $\mathbf{c}$ provides the semantic guidance for the shape generation process. Since we assume no text data at the training time, we leverage the text-image joint-embedding of CLIP. The generation is guided by the CLIP image embedding from the rendered images $\{\mathbf{I}_r\}$ during training and guided by the CLIP text embedding from the text prompt during inference. We add Gaussian noise to the CLIP image embeddings [44] to alleviate the modality gap between the CLIP text and image embeddings [20]:

$$\hat{\mathbf{c}} = f_I(\mathbf{I}_r) + \frac{\gamma \cdot \epsilon \cdot \|f_I(\mathbf{I}_r)\|_2}{\|\epsilon\|_2}, \tag{1}$$

where $f_I(\cdot)$ is the CLIP image encoder, $\epsilon \sim \mathcal{N}(0, 1)$, and γ controls the level of perturbation. The conditional vector $\hat{\mathbf{c}}$ is then normalized and passed through a mapping network that consists of multi-layer perceptrons (MLP). The mapped vector $\tilde{\mathbf{c}}$ conditions on the coarse transformer $T_c(.)$ by predicting the affine transform parameters of each transformer block's layernorm. In order to control the diversity-quality balance of the generated shapes, we replace the conditional vector with the null embedding by a $\rho\%$ drop out rate for classifier-free guidance [16]. These are important design choices that affects the diversity and quality of the generated shapes at inference time.

To train the coarse transformer $T_c(.)$, we randomly replace the input indices with a special "mask" tokens by uniformly selecting a masking ratio between 0-100% [4]. Given the masked latent grid $\mathbf{E}_{32}^{msk}$ and the conditional vector $\mathbf{c}$, the training objective is to maximize the following log-likelihood: $\sum_n^N \log p(\mathbf{E}_{32,n}|\mathbf{E}_{32,n}^{msk}, \mathbf{c})/N$, where N is the training data size and $\mathbf{E}_{32}^{msk}$ is the randomly masked latent grids. As the network sees only the ground truth tokens but not its own predicted samples during training, there exists an autoregressive drift resulted by the accumulative error at inference-time sampling. To alleviate this issue, we take inspiration from the NLP literature [38] and use a similar step-unrolled training (SUT) that predicts the entire unmasked grids at each time step. The final objective is to maximize the following formulation:

$$\sum_n^N \Big(\log p(\mathbf{E}_{32,n}|\mathbf{E}_{32,n}^{msk}, \mathbf{c}) + \log p(\mathbf{E}_{32,n}|\tilde{\mathbf{E}}_{32,n}, \mathbf{c}) \Big) \Big/ N, \tag{2}$$

where $\tilde{\mathbf{E}}_{32}$ is the coarse transformer $T_c(.)$'s own prediction.

3.3. Super-Resolution with Fine Transformer

Super-resolution in the discrete latent space improves quality and upsamples the resolution of generated shapes. This process is done with a fine transformer $T_f(.)$, which takes a low-resolution latent representation $\mathbf{E}_{32}$ and a masked high-resolution latent grid as input. The fine transformer is conditioned on the coarse grid $\mathbf{E}_{32}$ via cross attention. It unmasks the high-resolution representation $\mathbf{E}_{64}$. To ensure that training data resembles the inference data, we directly use the predictions $\tilde{\mathbf{E}}_{32}$ of the coarse transformer $T_c(.)$ to train the fine transformer $T_f(.)$, instead of using the ground truth low-resolution latent representation $\mathbf{E}_{32}$.

3.4. Annealed Classifier-Free Guidance

At inference time, we first convert an input text prompt into a CLIP text embedding using the CLIP text encoder $f_T(\cdot)$. The text embedding is then fed into the mapping network and used as the conditional vector $\tilde{\mathbf{c}}$ of the coarse transformer $T_c(.)$. We use the iterative decoding scheme [4] to slowly unmask the grid over a sequence of T steps. The initial input to the coarse transformer $T_c(.)$ is a completely

masked latent grid. At each time step t, we condition the coarse transformer $T_c(.)$ with $\tilde{\mathbf{c}}$ and take the output latent grid $\mathbf{E}^{msk}_{32,t-1}$ from the prior time step as the input. The coarse transformer $T_c(.)$ will predict an unmasked latent grid $\tilde{\mathbf{E}}_{32,t}$. We mask all the tokens except for the most confident token predictions and the tokens unmasked from previous steps. This newly masked latent grid $\mathbf{E}^{msk}_{32,t}$ is the input to the next time step $t+1$. We repeat this process until the process unmasks all the tokens, which is ensured by a cosine masking schedule [4].

During this iterative decoding scheme, we apply a new variant of classifier-free guidance [16]. Classifier-free guidance extrapolates an unconditional sample in the direction of a conditional sample. The extrapolation is controlled by an adaptive *guidance scale* parameter. Varying this parameter results in a trade-off between accuracy of text-to-shape generation and sample diversity. Instead of using a constant guidance scale for all time steps [16], we propose to vary the guidance scale over time according to an annealing schedule. The intuition is that during the initial steps of sampling when the input to the coarse transformer is mostly masked indices, a larger guidance scale is important to keep the network "on task"; later on in the sampling process, a lower guidance scale can help produce more sample diversity. Overall, the guidance is formulated as:

$$\hat{P}_t(c) = P_t(0) + a(t)\Big(P_t(c) - P_t(0)\Big), \tag{3}$$

where $a(t)$ is the guidance scale annealing schedule, $P_t(c)$ denotes the conditional distribution $p(\mathbf{E}_{32,t}|\mathbf{E}^{msk}_{32,t-1}, \mathbf{c})$, and $P_t(0)$ denotes $p(\mathbf{E}_{32,t}|\mathbf{E}^{msk}_{32,t-1}, \mathbf{0})$. Note that $a(t)$ is continuous and usually monotonically decreasing. By setting $a(t) = k$ for some constant k, this equation is equal to the classifier-free guidance scheme with a constant guidance scale. In this paper, we experimentally evaluate other annealing schedules [4], including linear, cosine, and square root functions.

Finally, the unmasked coarse latent grid $\tilde{\mathbf{E}}_{32}$ is fed to the fine transformer T_f, which uses the iterative decoding scheme to unmask an initially masked fine latent grid $\mathbf{E}^{msk}_{64}$. The unmasked fine grid is then passed to the 64^3 VQ-VAE decoder to obtain the final high-resolution voxel shape.

4. Experiments

In this section, we report experimental results to evaluate the generation fidelity, diversity, and class accuracy of CLIP-Sculptor. We provide full details of the experiment setup in the supplementary. We run the experiments 3 times for each of the aforementioned measures and report the mean in each case, except for the final comparisons with baselines (Table 2), where we use the best seed. Additional results can also be found in the supplementary.

Dataset. We conduct our experiments on two subsets of the ShapeNet(v2) dataset [3]. The first subset, *ShapeNet13*, contains 13 categories from ShapeNet as used in [7,24]. We use the same train/test split as specified in [24]. Our second subset is *ShapeNet55* which contains all 55 ShapeNet categories. For ShapeNet55, we render images as described in [7]; the training dataset contains 51,784 datapoints, whereas the test set contains 6,101 datapoints.

Evaluation Metrics. To assess the generative capabilities of CLIP-Sculptor, we use Fréchet Inception Distance (**FID**) ([14]) to evaluate shape diversity and quality, and use Classifier Accuracy (**Acc**) to evaluate the text-shape correspondence. These metrics follow [37] where they first generated single mean shapes for 234 predetermined text queries and then passed all shapes through a classifier to measure **Acc**. The latent space of this classifier is also used for **FID**.

Baseline. We compare the performance of our method against CLIP-Forge [37], CLIP-Mesh [19] and DreamFields [18], which are currently state of the art for zero-shot text-to-shape generation. We also set up a comparison method called Zero-Shot AutoSDF(ZS-ASDF), which trains the supervised method AutoSDF [26] with the CLIP features as CLIP-Sculptor, to highlight the superiority of our conditional generation model. CLIP-Forge reports results on single shape generation for 234 text queries. These single shapes were generated using the mean of the prior (the Gaussian distribution), so we refer to these results as CF-MS. Note that this does not capture the diversity of shapes that can be generated given a text query. To capture results on more shape generations, we sample 32 shapes instead of one using a Gaussian (CF-G), truncated Gaussian (CF-TG) or clipped Gaussian (CF-CG) distribution. We also generate 32 shapes for ZS-ASDF and our method for fair comparison. We only compare qualitatively with DreamFields and CLIP-Mesh as the optimization time per query is significant (Table 1), and it does not use prior knowledge from the shape dataset.

4.1. Evaluating Shape Diversity and Accuracy

In this section, we first quantitatively evaluate the diversity and accuracy of a given shape matching a given text query on the ShapeNet13 dataset. We compare with CLIP-Forge and ZS-ASDF using the **Acc** and **FID** metrics. The results are shown in Table 2. The first four columns represent CLIP-Forge(CF) and its different sampling techniques. The fifth column represents the result from AutoSDF trained with CLIP features(ZS-ASDF). The other columns represent our method with different annealing scale strategies. Three major things can be observed. First, all variants of our method outperform CLIP-Forge and ZS-ASDF significantly. This indicates that the method produces more diverse and higher fidelity shapes of increasing accuracy. Second, it can be seen that annealing strategies

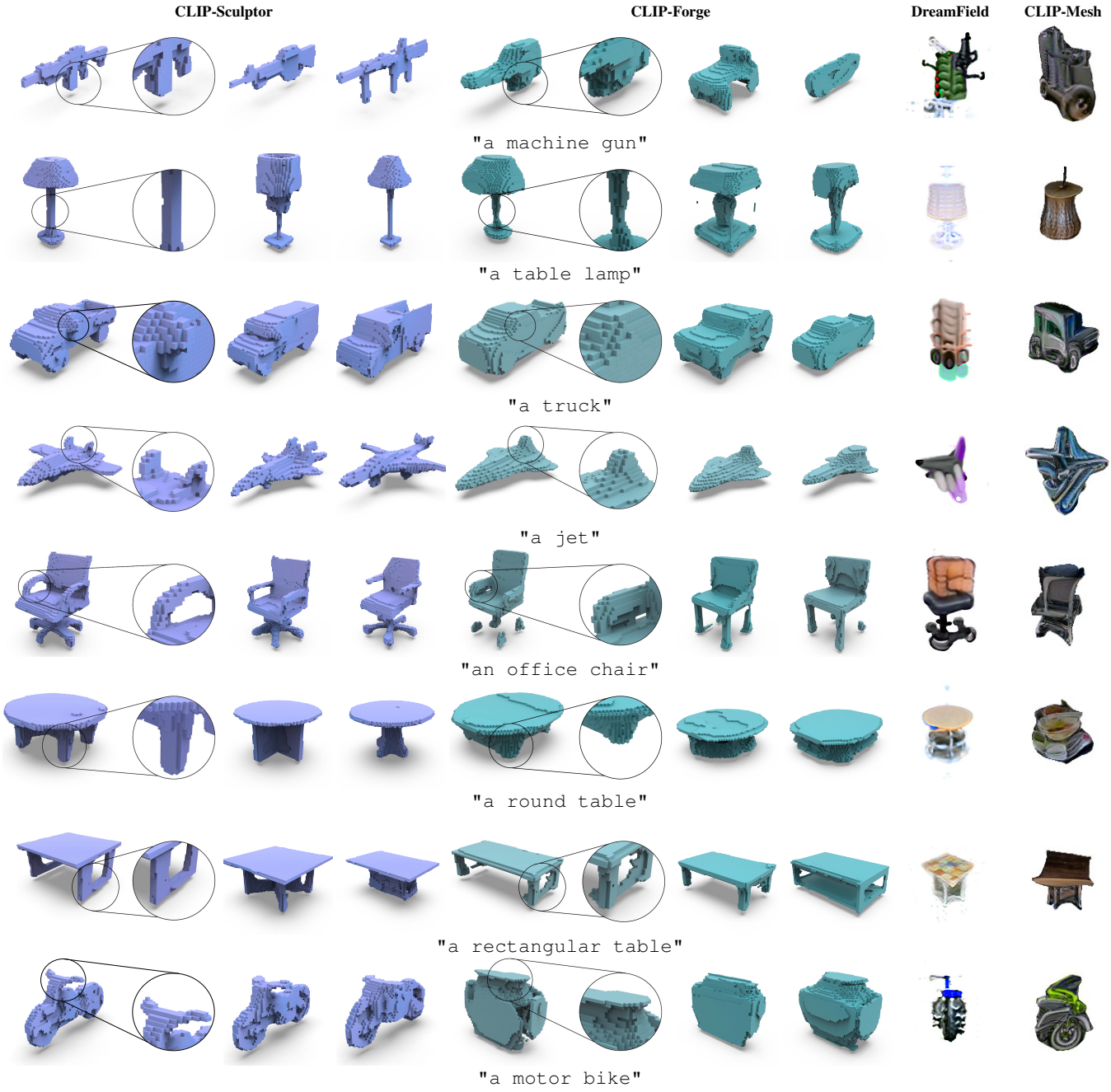

Figure 3. Qualitative comparison of CLIP-Sculptor (rendered in purple), CLIP-Forge [37](rendered in green), DreamField [18], and CLIP-Mesh [19] on text-conditioned generation. For each text prompt, the first generated shape by CLIP-Sculptor and CLIP-Forge [37] is zoomed in for detail comparison. Comparing with other methods, CLIP-Sculptor generates a broader category of shapes that correspond with the text prompt. The generated shapes are more diverse and has higher fidelity.

for guidance scale give a better accuracy-diversity trade-off than constant scale guidance. Finally, we note that simply augmenting AutoSDF (another VQ-VAE) with CLIP features does not lead to improved text to shape generation, indicating the importance of our architecture design decisions.

We also qualitatively compare our method to CLIP-Forge, CLIP-Mesh, and DreamFields. The results are shown in Figure 3. It can be observed that our method generates higher quality shapes (view "a machine gun"), with more detail (view "an office chair") and higher diversity (view "a jet"). CLIP-Forge usually produces the same shape

Table 2. Comparisons of CLIP-Forge(CF) baseline (across different sampling strategies), AutoSDF augmented with CLIP (ZS-ASDF) with CLIP-Sculptor (CS) on Accuracy(ACC) and FID. CLIP-Sculptor outperforms other methods in both metrics, which proves it generates shapes with higher fidelity and better diversity.

Method	CF-*MS*	CF-*G*	CF-*TG*	CF-*CG*	ZS-ASDF	CS-*const.*	CS-*sqrt*	CS-*linear*	CS-*cosine*
FID ↓	2425.25	2233.48	2141.61	2100.67	7332.93	1821.78	**1480.11**	1629.51	1725.63
ACC ↑	83.33	62.81	68.71	71.11	39.14	86.59	87.08	**87.50**	87.27

Table 3. Ablations on the major components of CLIP-Sculptor. CLIP-Sculptor achieves the best performance comparing with other design choice, which proves the effectiveness of the major components.

(a) Effect of varying the *Noise* parameter.

γ	×	0.5	0.8	1.0	1.2	1.5
FID↓	1720.02	1764.98	1484.61	1703.38	**1447.91**	1478.17
ACC↑	64.87	75.73	77.41	79.09	**79.63**	78.47

(b) Effect of varying number of layers (L) in the mapping network.

L	0	1	2	3	4	5
FID↓	2874.87	1716.73	1518.97	1447.91	1532.50	**1424.46**
ACC↑	62.72	78.70	79.17	**79.63**	77.16	76.09

(c) Comparison with baselines on super resolution.

Method	3D-UNet	64^3 -DS	CLIP-Sculptor
FID↓	2056.92	2196.96	**1910.28**
ACC↑	86.65	77.92	**86.85**

with small variations (view "a rectangular table"). We also note that DreamFields and CLIP-Mesh produce very abstract results which may not be useful in many applications. Finally, we also show results on ShapeNet55 in the last row of Figure 3. We could not get CLIP-Forge to produce sensible shapes for most text queries on ShapeNet55 (which we attribute to the data imbalance issue of ShapeNet55), whereas our method produced high fidelity shapes.

4.2. Major Components for Conditional Generation

Effect of Noise Parameter. During Stage 2 Training, to better align the image and text embeddings, we added Gaussian noise at varying levels of γ. We investigate the effect of (γ) added to the image condition vector and show the results in Table 3 (a). We keep the number of mapping layers fixed at three in this experiment. We observe that adding Gaussian noise drastically improves both the diversity and accuracy of generation, with the optimal noise parameter being around 1.2. These results indicate that adding noise during training helps align the text features seen at inference and the image features seen at training.

Size of Mapping Network. We next probe the importance of the mapping network. We show the results in Table 3 (b), where L represents the number of layers of the mapping function. In the case of 0 layers, we directly project conditional embeddings to the layernorm parameters using a linear layer. We make two observations from the results: 1) A common mapping network improves both the accuracy and FID. 2) Increasing the number of mapping network layers beyond a certain number decreases accuracy.

Classifier-Free Guidance. We next explore the relationship between dropping out image conditioning at $\rho\%$ during training and using *constant* classifier-free guidance during test time. In Table 4, we vary the scale parameter, $a(t) = k$, across the columns. It can be seen that as we increase the scale, accuracy typically increases while FID decreases. This indicates that the method is giving more accurate results on a given text query at the cost of shape diversity. A low dropout rate (5-15%) gives a good trade-off between accuracy and diversity.

Step-Unrolled Training (SUT). Finally, to further improve the accuracy we investigate the use of step-unrolled training [39]. The results are shown in the last row of Table 4. From the table it can be observed that step-unrolled training enables higher accuracy across all scales. This indicates that the model learns to unmask samples it would encounter during inference time.

4.3. Annealing Strategy for Scale Parameter

An important idea we propose in this paper is a scale annealing technique for classifier-free guidance. We employ classifier-free guidance to identify a better accuracy-diversity trade-off with different annealing schedules: constant, linear, cosine and square root. To determine which annealing technique works the best, we fix the FID values and plotted the accuracy at each fixed FID value (Figure 4). As different scale parameters give different FID values, we conducted an extensive grid search over the starting scale parameter. Note that finding the exact FID is not always feasible, so we pick the closest FID. Figure 4 shows results of FID versus accuracy on three different runs of the Coarse Transformer. We find that across all three runs, the accuracy is typically lower for a given FID for constant schedules as

Table 4. Classifier-Free Guidance and Step-Unrolled Training (SUT) experiment results. $\rho\%$ is the drop out rate for classifier-free guidance. SUT indicates the use of Step-Unrolled Training. The remaining columns indicate the variation in scale parameter.

$\rho\%$	SUT	$a(t) = 3$		$a(t) = 2.5$		$a(t) = 2$		$a(t) = 1.5$		$a(t) = 1$	
		FID↓	**Acc**↑	**FID**↓	**Acc**↑	**FID**↓	**Acc**↑	**FID**↓	**Acc**↑	**FID**↓	**Acc**↑
5	×	2059.0	85.73	1970.1	85.47	1790.9	85.43	1536.5	83.98	1227.5	79.17
10	×	1893.2	84.46	1821.2	84.50	1684.3	83.41	1522.4	82.34	1348.4	77.76
15	×	2086.9	85.03	1964.7	84.99	1851.9	84.36	1660.9	83.14	1485.5	78.19
20	×	2062.5	83.39	1972.3	82.95	1892.9	82.74	1733.3	81.13	1566.6	75.94
5	✓	2039.8	87.69	2011.1	87.39	1811.8	87.40	1678.6	86.24	1517.9	82.18

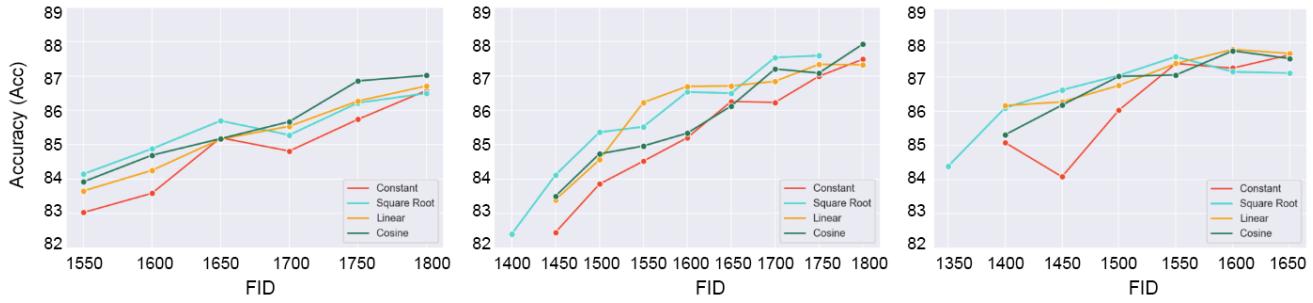

Figure 4. **FID** and **Acc** results with different classifier-free guidance scale annealing strategies (constant, cosine, square root, linear) across three different runs (different seeds) of the CLIP-Sculptor Stage 2 Coarse Transformer.

opposed to other schedules. This is especially the case at the lower range of FID values. The results indicate that having a large scale at the beginning of sampling is more important than later stages, especially in use cases where diversity is paramount.

4.4. Super-Resolution

Finally, we investigate the importance of hierarchical super-resolution. We compare our method with two baselines. In the first baseline, we directly use the 32^3 resolution results from the coarse transformer and use a 3D-UNET based super-resolution network to translate from 32^3 to 64^3. For the second baseline, we train a transformer directly on 64^3 VQ-VAE that is conditioned on text features as opposed to a coarse resolution grid. We refer to this as 64^3-DS (64^3 direct synthesis). The results are shown in Table 3(c). It can be seen from the table that latent-based super-resolution outperforms the baselines in both accuracy and diversity.

5. Conclusion

We present CLIP-Sculptor, a text-to-3D-shape generation method capable of producing shapes of high fidelity and diversity without the need for (text, shape) pairs. To achieve this, CLIP-Sculptor leverages CLIP and implements super-resolution in a discrete latent space with a hierarchical architecture and a novel annealed variant of classifier-free guidance on a mask-based model. We validate CLIP-Sculptor by comparing it with a number of baselines in terms of FID and accuracy, finding that CLIP-Sculptor is the new state of the art for this problem. In experimenting with different guidance scale scheduling, we find that constant scale scheduling did not always work the best, an important finding for the diffusion modeling community that may improve generation quality. Our paper is an important step towards improved quality and diversity of 3D text-to-shape generation outcomes.

Limitations and Societal Impact. Despite providing higher quality over baseline methods, CLIP-Sculptor still suffers from limitations in its ability to capture smaller details (e.g., chair with a hole on its back) and generate shapes from text prompts involving counting (e.g., chair with *four* slats). Furthermore, CLIP-Sculptor also fails on many text queries which are not present in the prior knowledge of CLIP. The method also lacks the ability to produce texture. In future work, we will address these limitations by exploring implicit representations that can capture even finer details and by investigating neural networks that can count.

Text-to-shape approaches can make the generation of 3D content more accessible and efficient. Within the text-to-image domain there has been concern that automatic approaches may replace artists. Because our approach produces voxels, a representation that lends well to direct manipulation and handoff to 3D designers, we believe that our text-to-3D approach can augment 3D design workflows rather than replace them.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 2

[2] Panos Achlioptas, Judy Fan, X.D. Robert Hawkins, D. Noah Goodman, and J. Leonidas Guibas. ShapeGlot: Learning language for shape differentiation. *CoRR*, abs/1905.02925, 2019. 1

[3] Angel X. Chang, Thomas Funkhouser, Leonidas 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, 2015. cite arxiv:1512.03012. 5

[4] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022. 2, 3, 4, 5

[5] 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. 1, 2

[6] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[7] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 5

[8] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020. 2

[9] Rao Fu, Xiao Zhan, Yiwen Chen, Daniel Ritchie, and Srinath Sridhar. Shapecrafter: A recursive text-conditioned 3d shape generation model, 2022. 1, 2

[10] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors, 2022. 2

[11] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. 2

[12] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022. 2, 4

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[14] 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. 5

[15] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models, 2022. 2

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2, 4, 5

[17] Moritz Ibing, Isaak Lim, and Leif Kobbelt. 3d shape generation with grid-based implicit functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13559–13568, 2021. 2

[18] 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

[19] Nasir Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. *ACM Transactions on Graphics (TOG), Proc. SIGGRAPH Asia*, 2022. 1, 2, 5, 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. 4

[21] Vivian Liu, Han Qiao, and Lydia Chilton. Opal: Multimodal image generation for news illustration, 2022. 1

[22] Vivian Liu, Jo Vermeulen, George Fitzmaurice, and Justin Matejka. 3dall-e: Integrating text-to-image ai in 3d design workflows, 2022. 1

[23] Zhengzhe Liu, Yi Wang, Xiaojuan Qi, and Chi-Wing Fu. Towards implicit text-guided 3d shape generation. *arXiv preprint arXiv:2203.14622*, 2022. 1, 2

[24] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[25] 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. 1, 2

[26] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. *arXiv preprint arXiv:2203.09516*, 2022. 1, 2, 5

[27] 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, 2021. 4

[28] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017. 2, 3, 4

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

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1, 2

[31] 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. 1

[32] 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. 1

[33] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 2, 4

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2

[35] 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 (CVPR)*, pages 10684–10695, June 2022. 1

[36] 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, 4

[37] 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, 5, 6

[38] Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation. *arXiv preprint arXiv:2112.06749*, 2021. 4

[39] Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation, 2021. 7

[40] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021. 1

[41] 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

[42] Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6239–6249, 2022. 2

[43] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. 2

[44] 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. 4

[45] Adrian Łańcucki, Jan Chorowski, Guillaume Sanchez, Ricard Marxer, Nanxin Chen, Hans J. G. A. Dolfing, Sameer Khurana, Tanel Alumäe, and Antoine Laurent. Robust training of vector quantized bottleneck models, 2020. 4