Looking at words and points with attention: a benchmark for text-to-shape coherence

Andrea Amaduzzi Giuseppe Lisanti Samuele Salti Luigi Di Stefano

CVLAB, Department of Computer Science and Engineering (DISI)
University of Bologna, Italy
{andrea.amaduzzi4, giuseppe.lisanti, samuele.salti, luigi.distefano}@unibo.it

Abstract

While text-conditional 3D object generation and manipulation have seen rapid progress, the evaluation of coherence between generated 3D shapes and input textual descriptions lacks a clear benchmark. The reason is twofold: a) the low quality of the textual descriptions in the only publicly available dataset of text-shape pairs; b) the limited effectiveness of the metrics used to quantitatively assess such coherence. In this paper, we propose a comprehensive solution that addresses both weaknesses. Firstly, we employ large language models to automatically refine textual descriptions associated with shapes. Secondly, we propose a quantitative metric to assess text-to-shape coherence, through cross-attention mechanisms. To validate our approach, we conduct a user study and compare quantitatively our metric with existing ones. The refined dataset, the new metric and a set of text-shape pairs validated by the user study comprise a novel, fine-grained benchmark that we publicly release to foster research on text-to-shape coherence of text-conditioned 3D generative models.
Benchmark available at https://cvlab-unibo.github.io/CrossCoherence-Web/.

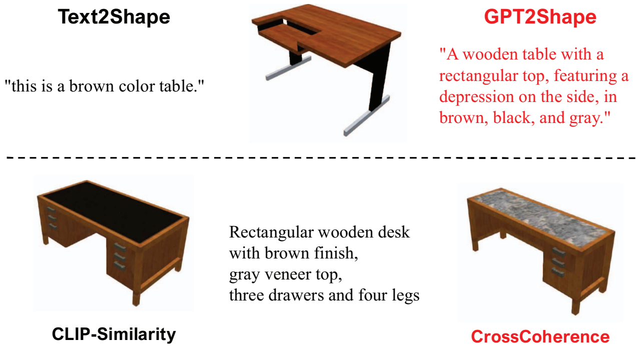

Text2Shape
"this is a brown color table."
GPT2Shape
"A wooden table with a rectangular top, featuring a depression on the side, in brown, black, and gray."
Rectangular wooden desk with brown finish, gray veneer top, three drawers and four legs
CLIP-Similarity
CrossCoherence

Figure 1. Benchmarking text-to-shape generative models calls for bett-er datasets and better metrics. **Top:** as existing datasets [6] contain many uninformative descriptions (left), we automatically create high-quality text prompts (right) by leveraging GPT-3 [4]. **Bottom:** the existing metric CLIP-Similarity judges the left shape as more coherent to the given text than the right one, whilst our novel CrossCoherence prefers the right shape over the left one.

1. Introduction

The rapid development of text-to-image generative models has enabled generation and manipulation of photorealistic images from natural language prompts, in a matter of seconds [32, 12, 24, 31, 15, 42, 33, 13, 2]. Inspired by such results, recent works [25, 17, 20, 6, 27, 40, 35, 19, 14] have started to explore the task of text-conditioned generation of 3D shapes.

Two main paradigms are emerging for text-to-shape generation: one is a direct extension of the successful approach used to learn text-conditioned generative models for images and relies on supervised learning on high-quality paired text-shape datasets [6, 20, 17]. The extension of such paradigm to 3D shapes is however severely limited by the lack of general, high-quality 3D datasets: the only datasets for text-driven shape generation are Text2Shape [6] and its extended version Text2Shape++ [14], which contain text descriptions for chairs and tables from ShapeNet [5]. However, as shown in Figure 1 (top, left), the textual descriptions provided by these datasets are often generic and fail to capture all the key, fine-grained details of the objects in terms of both geometry and appearance. The second paradigm [35, 27, 19] sidesteps the need for paired datasets by leveraging pre-trained CLIP [29] or text-to-image models [32, 12, 24, 31, 15, 42, 33, 13, 2] to learn to align 3D content creation to the input text. This paradigm has shown impressive results, but such methods are usually based on optimizing NeRF models [22, 23, 16, 27, 40, 19] out of the generated images, which results in impractical run-time costs and latency (e.g., Magic3D [19] takes 40 minutes on 8 NVIDIA A100 GPUs to generate one single shape).

Another important limitation for both paradigms is the lack of a common benchmark. It is impossible to overstate the importance of benchmarks in enabling and driving the rapid developments of deep learning: this is a cornerstone of the field since ImageNet [10] gave rise to the AlexNet breakthrough [18]. The definition of a clear benchmark for text-to-shape generation is hindered not only by the absence of high-quality paired text-3D datasets but also by metrics. Most currently adopted quantitative metrics can measure only the generative ability of a method, i.e. the realism of the generated shapes in terms of distance between their distribution and the ground-truth one. These metrics, however, are not able to quantify how closely a 3D shape fits a text, arguably the most important aspect to properly evaluate text-to-shape methods. Exceptions are metrics based on CLIP embeddings [29] which are computed from 2D renderings of the 3D shape and text embeddings. However, these approaches may turn out sensitive to rendering parameters and struggle to capture coherence with individual words or fine-grained geometric details due to the global nature of the CLIP text and visual embeddings.

In this work, we propose the first benchmark for text-to-shape generation and manipulation that addresses both shortcomings highlighted above. Since collecting a novel dataset from scratch is a time-consuming and costly effort, we investigate on the effectiveness of pre-trained large language models (LLMs) to leverage existing datasets and improve their quality at a fraction of the cost. In particular, as shown in Figure 1 (top, right), we create an improved version of Text2Shape, which we dub GPT2Shape, whose textual descriptions have been generated by the large language model GPT-3 [4] starting from the text prompts of the original dataset. The higher quality of such text prompts has been validated through a human evaluation study. GPT2Shape provides multiple fine-grained textual descriptions for every 3D shape. The detail and accuracy of textual descriptions can enable training of generative models which can effectively capture and learn the relationships between the words in the input text and the localized details of the corresponding 3D shape. Such capability would be very hard to achieve with noisy and imprecise text prompts. We also propose a novel metric, dubbed CrossCoherence, which does not use renderings and exploits the cross-attention mechanism to quantify the coherence between a coloured point cloud and a textual description at the word and geometric detail level. We use the text-shape pairs validated through the human study to quantitatively compare with existing text-3D coherence metrics and show how CrossCoherence outperforms previous proposals (Figure 1, bottom). Finally, as a by-product of our human evaluation study, we create the Human-validated Shape-Text (HST) dataset by collecting the textual descriptions which have been coherently associated with a shape by the participants. The usefulness of the HST dataset is twofold: it provides a set of prompts and associated shapes, that can be used as a test set of our benchmark since they are not present in the training set but come from the same distribution on which CrossCoherence has been trained; it may also serve as a benchmark for the development of new text-to-shape coherence metrics.

To sum up, the main contributions of this work are:

- **GPT2Shape**, an automatically improved dataset consisting of shape-text pairs with high-quality, fine-grained textual descriptions;
- **CrossCoherence**, a state-of-the-art quantitative metric for text-to-shape coherence, which can be directly applied to RGB point clouds;
- **HST**, a human-validated test set where CrossCoherence can be used to evaluate and compare text-driven generative models.

2. Related Work

Text-to-shape coherence metrics: Several recent studies have explored the problem of text-conditioned 3D generation. To assess fidelity to the input prompt, [20], [35] and [6] compute several distance measures between a generated shape and the one associated with the input prompt in the ground-truth set, such as Intersection-over-Union (IoU), Earth's Mover Distance (EMD), Chamfer Distance (CD) and Mean Squared Error (MSE). The main limitation of such a strategy is that, in generative tasks, a ground-truth shape is just one possible correct outcome of the generation process and relatively higher or lower distances from it do not measure the quality of the output nor, in the case of text-conditioned generation, its coherence with the given text. The most convincing metrics for text-to-shape coherence are based on CLIP embeddings [29]. One such metric is CLIP R-precision [26], used in several studies [25, 20, 27, 40, 17]. It is defined as the accuracy with which CLIP retrieves the correct caption among a set of distractors given a rendering of the generated shape. Yet, such quantitative assessment is affected by several limitations: first of all, it is based on renderings, therefore it is influenced by a number of parameters, like virtual camera placement, virtual illumination, and number of rendered views; secondly, the selection of the text distractors is arbitrary; finally, as it relies purely on rendered views, whose quality is not homogeneous among 3D data representations, a comparison between 3D shapes from different representations (e.g. point clouds and meshes) can be unfair. CLIP-similarity [14], instead, computes the cosine-similarity between CLIP image features extracted from several shape renderings and CLIP text features. Due to the reliance on renderings, this metric shares the same limitations of CLIP R-precision.

ShapeCrafter [14] has been the first and only work to use ShapeGlot [1] as a text-shape fidelity metric. In its original formulation, given a set of shapes, this metric is able to discriminate which one is the most coherent with respect to an input text description, by extracting 3D features, 2D features from renderings, and text features. Despite its usefulness, there are some clear disadvantages associated with this method. Firstly, similar to CLIP-based metrics, its effectiveness relies on the parameter settings used to render the views of the shape. Secondly, the text representation employs an LSTM architecture, which has been acknowledged to have difficulties in representing long text descriptions [3].

To address all the above limitations, we propose Cross-Coherence, a metric for text-to-shape coherence that does not depend on rendering parameters, operates directly on colored point clouds to take both geometry and appearance into account, leverages LLMs to extract better text features and exploits cross-attention to assess the coherence of words with shape parts.

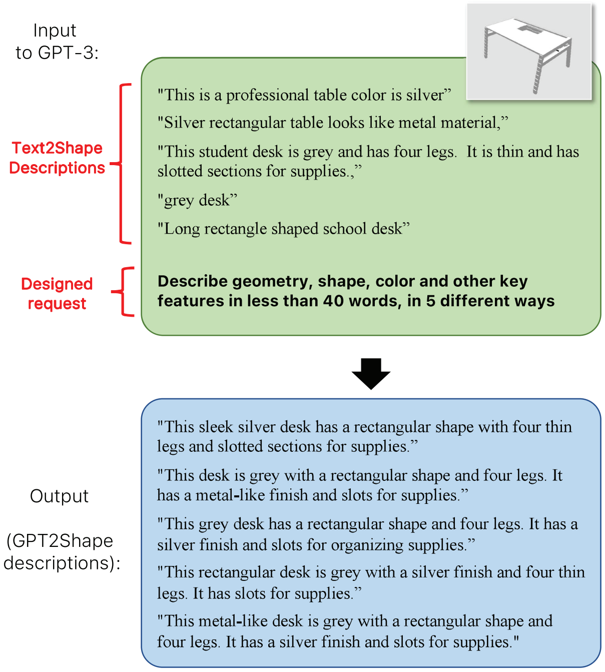
Datasets Several text-driven 3D generation models require supervision in the form of text-shape pairs. Unlike in the 2D case, where web scraping can be used to obtain large amounts of image-text pairs [7, 38, 37, 36], obtaining such labeled data is extremely difficult, due to 3D shapes not being common on the Internet and the complexity and cost of creating text descriptions for large 3D datasets like ShapeNet [5]. Text2Shape [6] has been the first dataset with shape-text pairs to be introduced. It contains text prompts for two Shapenet classes: chairs and tables. Overall, Text2Shape provides 75K shape-text pairs, with multiple text descriptions for each 3D object. However, the quality of these descriptions is highly variable: the prompts are either too generic (*"A chair with four legs"*) or contain irrelevant details (*"Table with high-quality wood, you can eat and enjoy with your family"*) or both. Sometimes the text does not describe correctly the geometry and/or appearance of the objects (*"A table with three legs"* but the table has actually four legs). In addition, many sentences contain grammatical and syntactical errors. More examples are provided in the supplemental. These nuances make the process of learning the complex relationship between 3D data and text even more difficult and limit the validity of comparisons carried out on this dataset. Text2Shape++ [14] is a dataset built upon Text2Shape, including 369K shape-text pairs. This dataset has been built specifically for the task of recursive shape generation; indeed, the original text descriptions from Text2Shape have been split into multiple smaller sentences with incremental amount of information (e.g. *"a chair"*, *"a chair with four legs"*, *"a chair with four legs and no armrests"*). Since Text2Shape++ is built starting from the same descriptions of Text2Shape, its text prompts suffer from the same limitations. Objaverse [9] is a very recent large-scale dataset that provides more than 800K 3D models together with descriptions. However, the majority of these texts specify only the class of the object and some attributes, without conveying information about its actual 3D geometry and appearance.

In response to these challenges, we propose an automatically improved version of Text2Shape, referred to as GPT2Shape, which contains new text descriptions for every object of Text2Shape. These novel sentences can describe more accurately the geometry, appearance and texture of all the objects in the dataset, as proved by our human evaluation study and exemplified in Figure 2.

3. GPT2Shape

Figure 2 presents an exemplar shape and its prompts from Text2Shape. As it can be seen from this figure, the quality of the original descriptions is highly variable and some of them are quite poor: for instance, the sentence *"grey desk"* is too generic and not able to drive any generative model towards the intended 3D shape. Improving such dataset through a new human-based annotation on the Amazon Mechanical Turk crowd-sourcing platform [8], utilized by Text2Shape, would be extremely time-consuming and expensive.

At the same time, enough details about a shape are usually present in Text2Shape if information from multiple prompts are merged, as shown again in Figure 2. On the basis of this observation, we decided to explore if an automatic improvement of the existing dataset was feasible, by exploiting LLMs. In particular, the generation of the new text sentences has been achieved through the OpenAI GPT-3 model [4] *davinci*. This model has been used in its text completion scenario in this manner: for each shape of the dataset, all its corresponding text prompts from Text2Shape have been given as input to the language model, along with the request to generate better text descriptions. The request itself was the result of a careful prompt engineering process, in order to obtain the best possible result. Figure 2 illustrates an example of this process, showing the initial text descriptions, the request to GPT-3, and the resulting sentences. As it can be noticed, the output textual descriptions contain all the relevant geometric and appearance information regarding the object. This text rephrasing process was carried out on the whole train-val-test splits of Text2Shape, so as to keep the same amount of data samples as the original dataset: 15032 shapes, with a total of 75358 shape-text pairs. In the supplementary material, multiple comparisons between text prompts from Text2Shape and GPT2Shape are provided.



"This is a professional table color is silver"

"Silver rectangular table looks like metal material,"

"This student desk is grey and has four legs. It is thin and has slotted sections for supplies.,"

"grey desk"

"Long rectangle shaped school desk"

Describe geometry, shape, color and other key features in less than 40 words, in 5 different ways

"This sleek silver desk has a rectangular shape with four thin legs and slotted sections for supplies."

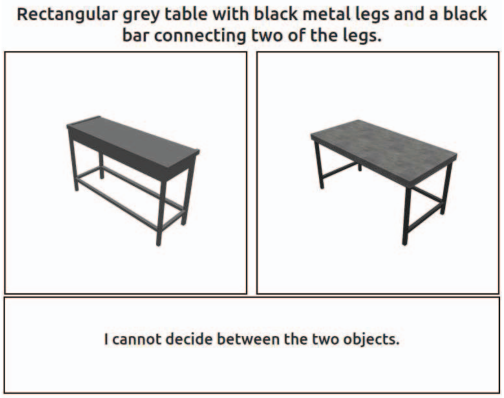
"This desk is grey with a rectangular shape and four legs. It has a metal-like finish and slots for supplies."

"This grey desk has a rectangular shape and four legs. It has a silver finish and slots for organizing supplies."

"This rectangular desk is grey with a silver finish and four thin legs. It has slots for supplies."

"This metal-like desk is grey with a rectangular shape and four legs. It has a silver finish and slots for supplies."

Figure 2. An example of the text prompts rephrasing process. The green box contains the text2shape descriptions for a shape together with the designed request; their combination represents the input prompt to GPT-3. The blue box is the output obtained from GPT-3 and contains the shape descriptions that have been used to populate our GPT2Shape dataset.



Rectangular grey table with black metal legs and a black bar connecting two of the legs.

I cannot decide between the two objects.

Figure 3. Layout of the user study

4. GPT2Shape vs Text2Shape: a user study

Once the rephrased sentences had been obtained via GPT-3, a user study aimed at comparing the informativeness of the old (Text2Shape) and new (GPT2Shape) text prompts was carried out. The layout of this user study is shown in Figure 3: each user was shown a pair of views from two objects together with a text prompt. The prompt comes from the test set of either Text2Shape or GPT2Shape and describes one of the two objects (i.e. the reference object, unknown to the user) while the other one acts as a distractor. The left/right position of the reference object was randomized. The task consisted in clicking on the object which, according to the user, would be described better by the given text. In case of uncertainty, due to the prompt describing equivalently well both objects or not sufficiently well any of the two, the user could click on a third option, located under the images. This enabled us to compare the two datasets based on users' choices: the higher the number of clicks on the -unknown- reference object, the higher the informativeness of the associated text prompt.

To avoid presenting the user with a task far too simple, as it might have been the case had distractors been chosen randomly, the *reference-distractor-text* triplets were built as follows. For every reference object, we have defined two *hard* distractors and one *easy* distractor. All distractors for a shape have been selected within its class (e.g., chairs or tables) and defined on the basis of the Euclidean distance in the latent space of a PointNet++ [28] autoencoder trained on the task of RGB point cloud reconstruction on chairs and tables from ShapeNet. In this way, we obtained three pairs for each reference object. When running the user study, a pair of shapes were randomly sampled from this set, whereas the text describing the reference shape was taken from Text2Shape or GPT2Shape with 50% probability. In the supplemental, we provide more details and examples on easy-hard shape distractors and triplets from the user study.

Overall, 175 users took part in the study, contributing with 3642 data samples (i.e., clicks). In order to evaluate Text2Shape and GPT2Shape, we kept track of the answers provided by the users when presented with text prompts from both datasets. Table 1 summarizes the results by reporting the percentages of times users selected the reference object, the distractor, or felt they could not decide when presented with a text from Text2Shape or GPT2Shape. When the text came from GPT2Shape, about 75% of the answers were correct, i.e. the human selected the reference shape as the most coherent with the provided text description. On the other hand, Text2Shape allowed the users to answer correctly only 65% of the time. Moreover, the percentage of undecidable sentences is higher for Text2Shape (about 25%) compared to GPT2Shape (about 16%).

Such experimental findings validate our claim that large language models can be used to automatically filter and combine information from multiple incomplete textual descriptions. They also validate that textual descriptions in GPT2Shape are more informative than the original prompts from Text2Shape.

Dataset	reference (↑)	cannot decide (↓)	distractor (↓)
Text2Shape	65.65%	24.95%	9.40%
GPT2Shape	**75.76%**	**16.21%**	**8.02%**

Table 1. Summary of user study results

5. Human-validated Shape-Text dataset

The results of the user study were used also to aggregate a refined test set of shape-text pairs with descriptive texts from the test sets of Text2Shape and GPT2Shape, which we dub Human-validated Shape-Text (HST) dataset. When building this subset, we only selected the sentences for which all users made the same choice between the two objects, i.e., where there was unambiguous consensus among participants that the description was describing well only one of the two shapes. HST, which contains 2153 text-shape pairs, provides a human-validated benchmark to assess the quality of text-to-shape generative models. It can be used as a set of descriptive text prompts to evaluate such models with text-to-shape coherence metrics. To provide a comprehensive dataset, for every pair in the HST dataset, we also included the other object that was shown to the human evaluators, along with the indication of the correct association for the text. This extended version of HST is useful for quantitatively validating metrics for text-to-shape coherence.

6. CrossCoherence

In this section, we describe the details of our proposed metric for text-to-shape coherence, CrossCoherence. Recent works on text-conditioned generative models, both in 2D [32, 12, 24, 31, 15, 42, 33, 13, 2] and 3D [19, 40, 25, 20, 17, 27, 14] have shown that frozen large language models, like BERT [11], GPT [4] and T5 [30], trained only on text data, can be very effective text encoders for generative tasks. The solution proposed in [34] has also shown that large language models are more effective than text encoders trained on paired image-text data, such as CLIP [29]. Another key reason for the outstanding results achieved by such works is the text-conditioning scheme, which is based on the cross-attention mechanism. On the basis of such observations, we have decided to explore the use of large language models and cross-attention layers for the definition of our metric.

6.1. Architecture

The proposed architecture of CrossCoherence is shown in Figure 4. Our model takes as input a text description and a group of colored point clouds, with coordinates *(x, y, z)* as well as *(R, G, B)* values for every point. It provides as output a set of soft-max normalized scores, which are used to predict which point cloud is best described by the input text. In practice, a Point Cloud Encoder extracts meaningful local embeddings from every input point cloud. At the same time, a Text Encoder extracts text features from the input sentence. Then, for every point cloud, two *bilateral* cross-attention layers reason on the coherence between the shape and text features computed by the encoders. These layers are referred to as *bilateral* since, in a first layer, the queries of the attention maps are computed on the shape features, whereas the keys and values are obtained from the text embeddings, while in a second layer this computation is reversed. This *bilateral* layer equips the metric with the ability to reason on the mutual relation between words and 3D details, which is key to evaluating text-to-shape coherence, as shown by the experiments reported in Section 7. Finally, the resulting embeddings from both cross-attentions are concatenated, after undergoing an average pooling layer. These features are fed into an MLP in order to provide a score of coherence for each point cloud and text. At training time, the scores are softmax-normalized and a standard cross-entropy loss is used to guide the network to produce higher scores for the correct pair. As Pointcloud Encoder we used the encoder part of a PointNet++ network [28] trained for Shape Reconstruction on chairs and tables from ShapeNet. This network is the same model used to build the easy and hard distractors for the user study described in Section 4. When building the distractor pairs, we used the features coming from the third set abstraction layer, which computes a global embedding, whereas when training CrossCoherence we used the embeddings computed by the second set abstraction layer, which computes local features on the downsampled input point cloud. As for the Language Encoder, we used the frozen encoder from T5 [30].

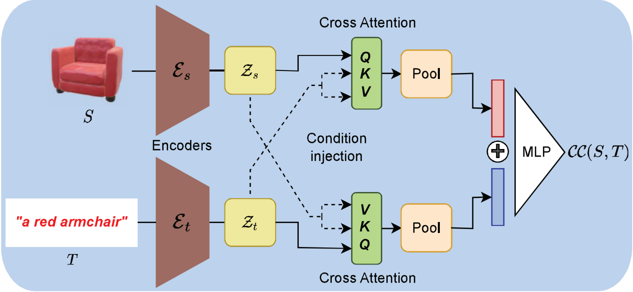


Figure 4. Architecture of CrossCoherence

6.2. Training

During training, the network was fed with *(shapes, text)* pairs and target vectors indicating the ground-truth shape corresponding to the textual description. Based on the outcome of the user study, we trained CrossCoherence on the training set of GPT2Shape. Before training, following the same strategy applied for the human evaluation survey, described in Section 4, for every shape in the training set of GPT2Shape, we have extracted one easy and two hard distractors. This *easy-hard* training strategy was found to be more effective than constructing the pairs randomly due to the network learning to deal with cases where the input point clouds are very similar to each other, which in turn is conducive to better generalization to unseen examples.

Method	**Acc.** on chairs (↑)	**Acc.** on full HST (↑)
ShapeGlot	70.21%	-
CLIP-Similarity	77.79%	77.06%
CrossCoherence	**81.04%**	**80.45%**

Table 2. Comparison between ShapeGlot, CrossCoherence and CLIP-Similarity. The second column reports accuracy for methods trained and tested only on chairs, while the third column is for methods trained and tested on full HST.

Test set	ShapeGlot	CrossCoherence	CLIP R-precision
HST chairs	5.36%	**17.26%**	9.87%
full HST	-	**16.85%**	9.38%

Table 3. Comparison between ShapeGlot, CrossCoherence and CLIP R-precision based on the R-precision protocol. The second row reports the results for methods trained and tested only on chairs, while the third row is for methods trained and tested on full HST.

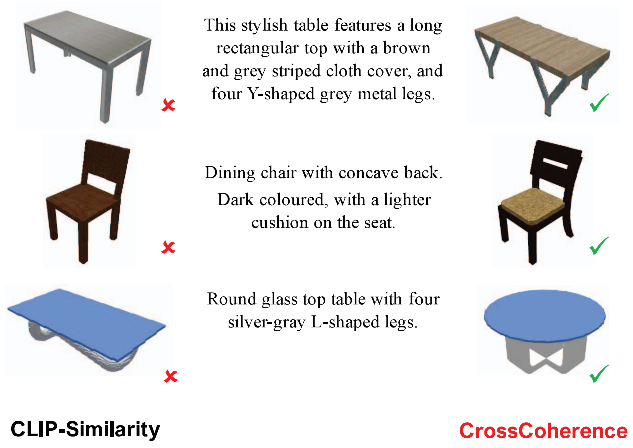
7. Experimental Results

We performed several experiments to assess the effectiveness of CrossCoherence.

7.1. Comparison with existing metrics

In this section, we evaluate the accuracy in assessing text-to-shape coherence for our proposed CrossCoherence metric, as well as for Shapeglot [1], CLIP-Similarity [14, 29] and CLIP R-precision [26, 29]. The accuracy computation relies on the triplets of HST (Section 5) and consists in two different evaluation protocols. When comparing with CLIP-Similarity, we report how often each metric yields a higher score to the ground truth shape than to the other one in the triplet for the given text description. In contrast, when dealing with CLIP R-precision, we adopt the protocol introduced in [16] and used by [25, 20, 27, 40]: given a text prompt and the corresponding ground-truth 3D shape from HST, we construct a set of 153 text descriptions. This set comprises the ground-truth prompt and 152 other texts randomly sampled from a designated test set, in our case HST. Every metric will predict which text prompt within the set exhibits the highest coherence to the given 3D shape. By comparing the predicted prompt to the ground-truth prompt, we can assess the performance of the evaluation metric. To evaluate both CLIP-Similarity and CLIP R-precision, we conducted experiments using various CLIP base models and different numbers of renderings. When dealing with more than one rendered view, we used the maximum similarity across the views, as in [14]. The configuration that achieved the best performance, which we consider here, relies on the ViT-L/14 encoder and 20 renderings. Yet, we deem it worth highlighting that the accuracy of CLIP-based metrics varies significantly depending on the chosen configuration (CLIP base model, 3D data representation, rendering parameters), as shown in additional experiments reported in the supplementary material. As for ShapeGlot [1], we trained the network on the same dataset as CrossCoherence, i.e. easy and hard samples from GPT2Shape, as described in Section 6. However, while training CrossCoherence relies only on point clouds and text prompts, ShapeGlot consumes also rendered views of objects' meshes and requires a pre-trained VGG encoder to extract features from these renderings. Unfortunately, the authors did not release the weights of the VGG model and made available only pre-computed features for the *chair* class of Text2Shape. Hence, we can assess the accuracy of ShapeGlot only on the chair subset of HST.

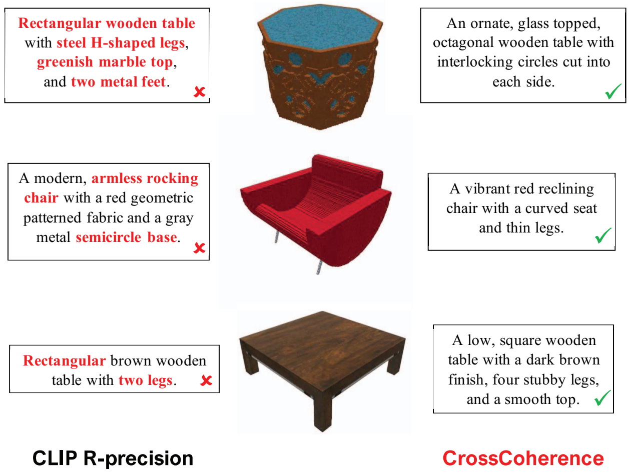
The outcome of our evaluation is shown in Tables 2 and 3. These results indicate that CrossCoherence is the


This stylish table features a long rectangular top with a brown and grey striped cloth cover, and four Y-shaped grey metal legs.

Dining chair with concave back. Dark coloured, with a lighter cushion on the seat.

Round glass top table with four silver-gray L-shaped legs.

Figure 5. Qualitative results of CLIP-Similarity and CrossCoherence on HST dataset. The green check indicates the shape associated with the prompt, while the red cross identifies the distractor. For all these triplets, CLIP-Similarity prefers the left shape while CrossCoherence the right one.


Rectangular wooden table with **steel H-shaped legs**, **greenish marble top**, and **two metal feet**. ✗

An ornate, glass topped, octagonal wooden table with interlocking circles cut into each side. ✓

A modern, **armless rocking chair** with a red geometric patterned fabric and a gray metal **semicircle base**. ✗

A vibrant red reclining chair with a curved seat and thin legs. ✓

Rectangular brown wooden table with **two legs**. ✗

A low, square wooden table with a dark brown finish, four stubby legs, and a smooth top. ✓

Figure 6. Qualitative results of CLIP R-precision and CrossCoherence on HST dataset. The green check indicates the ground-truth text. For all these triplets, CLIP R-precision retrieves the left text while CrossCoherence the right one.

most effective metric. We can observe how ShapeGlot performance is the least satisfactory. Moreover, the gap in accuracy between this metric and CrossCoherence suggests that modern text-analysis tools like LLMs and cross-attention layers are key to creating a more effective metric. Indeed, the main differences between CrossCoherence and ShapeGlot include the use of LLMs instead of LSTMs as well as the incorporation of the cross-attention mechanism. Finally, in both settings, CLIP-based metrics turn out to be the strongest competitors, yet, CrossCoherence outperforms CLIP-Similarity by more than 3% and is almost 100% more accurate than CLIP in the challenging R-precision setting which is nowadays the most widely used protocol to evaluate text-to-shape coherence.

Qualitative examples of the choices made by CLIP-Similarity and CrossCoherence are provided in Figure 5. The errors made by CLIP-Similarity seem to be caused by the use of global embeddings to compare shapes and text, which prevents the metric from evaluating fine details when judging the shape-text agreement. For instance, CLIP-Similarity does not take into account the shape of the table legs in the first row, the presence of the chair cushion in the second row and the geometry of the glass table in the last example. As shown in Figure 6, CLIP R-precision suffers from the same weakness as CLIP-Similarity in understanding fine-grained details of visual input and text. For example, CLIP R-precision provides high scores for text prompts that contain multiple wrong references to the geometry and appearance of the input shape, i.e. *"Rectangular wooden table"* in the first row, *"Armless rocking chair"* in the second example, *"two legs"* in the last row. This weakness has been highlighted in [39] and [43], which discuss how CLIP models, probably due to their contrastive pretraining strategy, have difficulty in processing fine-grained descriptions of visual content. Indeed, the performance of such models on the Attribution, Relation, and Order (ARO) benchmark [43] show that CLIP exhibits poor relational understanding and tends to behave like bag-of-words models. The use of cross-attention between local embeddings in CrossCoherence, on the other hand, enables to capture and link together important shape and text details, leading to the selection of the reference shape or text, as shown in Figures 5 and 6, respectively. Ablation experiments concerning the design choices behind our architecture are reported in the supplementary material, together with additional qualitative comparisons between the metrics.

7.2. Text-to-shape coherence of generative methods

A text-to-shape coherence metric can be used to evaluate text-conditioned 3D shape generative models. Here we address this setting and compare CrossCoherence, CLIP-Similarity [14] and CLIP R-precision [26] in their ability to judge upon the coherence to the input text of the shapes generated by *Point-E* [25], *Shap-E* [17] and the model proposed by Liu et al. [20]. These models employ distinct data representations, with Point-E being the current state-of-the-art method in text-driven point cloud generation, Shap-E using both NeRF (neural radiance field) and STF (signed textured field), Liu et al. [20] employing occupancy fields.

We used the pretrained weights provided by the authors for Point-E and Shap-E, while we trained from scratch Liu et al. [20] on GPT2Shape. We then used CLIP R-Precision, CLIP-Similarity and CrossCoherence to assess which method generates the 3D data most faithful to the input prompt. We adopted the same experimental protocols as in Section 7.1, the only difference being that here the 3D shapes compared to the text prompts are not the ground-truths from HST but have been generated by either Shap-E or Point-E or Liu et al.

Results are reported in Tables 4 and 5. Since Shap-E

Method	CrossCoherence	CLIP-Similarity	CLIP-Similarity
Point-E [25]	27.33%	24.36%	27.04%
Shap-E [17]	**38.02%**	**65.47%** (STF)	**57.18%** (NeRF)
Liu et al. [20]	34.65%	10.18%	15.78%

Table 4. Quantitative comparison between generative models, using CrossCoherence and CLIP-Similarity. Each entry in a column reports how often a metric considers a prompt more coherent with the shape generated by one method versus the others.

Method	CrossCoherence	CLIP R-precision	
Point-E [25]	4.92%	5.95%	
Shap-E [17]	**7.09%**	**20.41%** (STF)	**15.78%** (NeRF)
Liu et al. [20]	5.09%	5.83%	

Table 5. Quantitative comparison between generative models, using CrossCoherence and CLIP R-precision according to the R-precision protocol.

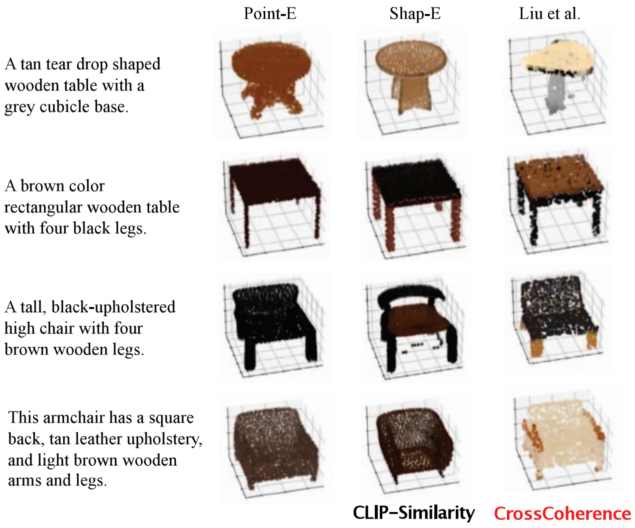

Point-E
Shap-E
Liu et al.
A tan tear drop shaped wooden table with a grey cubicle base.
A brown color rectangular wooden table with four black legs.
A tall, black-upholstered high chair with four brown wooden legs.
This armchair has a square back, tan leather upholstery, and light brown wooden arms and legs.
CLIP-Similarity
CrossCoherence

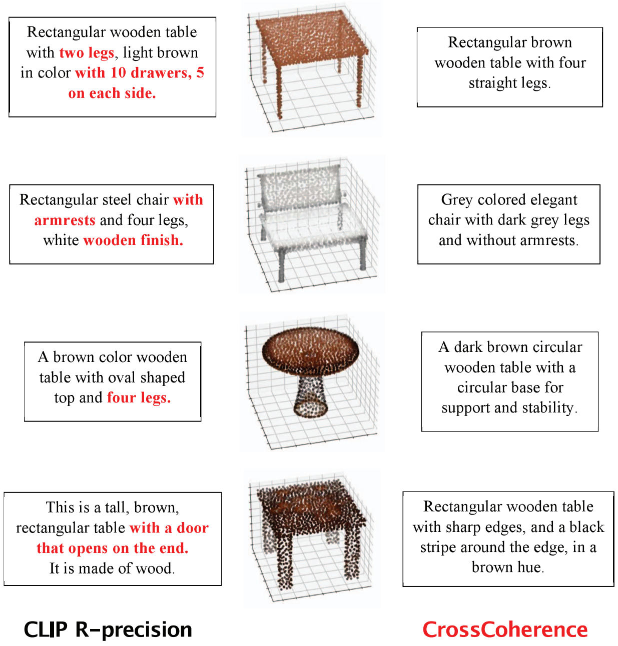
Figure 7. Examples of shapes generated by Point-E (left), Shap-E (center) and Liu et al. (right) for the given text prompt. As highlighted, for all these triplets, CLIP-Similarity prefers the shape in the middle while CrossCoherence the one on the right.

represents the generated shape as an STF and a NeRF, for CLIP-Similarity and CLIP R-precision we report the results computed on renderings from both representations. The differences in performance for Shap-E suggest again that CLIP-based metrics are significantly affected by the rendering process. In these experiments, we notice that all metrics agree on Shap-E being the method that generates the 3D shapes closest to the input text. However, the ranking of the other two methods in relation to Shap-E is very different. In fact, CrossCoherence ranks the method of Liu et al. [20] second, with a modest gap of 4% in the first protocol and 2% in the second. In contrast, CLIP-similarity and CLIP R-precision rank Point-E in second place with a very large gap in performance with respect to Shap-E (around 30% in the first protocol, and from 10% to 15% in the second one). By inspecting some triplets on which CrossCoherence and CLIP-Similarity disagree, shown in Figure 7, we can notice how CLIP-similarity seems unable to recognize certain geometric and color details, present in the shapes generated by Liu et al. [20], which are critical for determining the shape most coherent to the given text prompt. In the first row of Figure 7 CLIP-Similarity prefers the table in the middle although it lacks the *"grey cubicle base"* and the *"tear drop shape"* described in the text. In the second row, CLIP-Similarity is not able to localize the parts containing the colors specified in the text. As discussed in Section 7.1, such weakness may be caused by the difficulty of CLIP in processing fine-grained text descriptions and understanding the spatial relations among the objects' parts.

In Figure 8 we report some qualitative examples to motivate the large gap between CrossCoherence and CLIP R-Precision. Here, it is possible to observe that even if in some cases CLIP R-precision retrieves the ground-truth text as the most coherent with the shape from Shap-E, these descriptions contain details that are not really present in the generated shape (e.g., *"two legs"* in the first row, *"with armrests"* in the second row, *"four legs"* in the third row and *"with a door that opens on the end"* in the last row). This behavior arises because Shap-E fails to generate all the specified details in the ground-truth text and CLIP R-precision cannot accurately recognize the absence of these elements in the generated shape. On the contrary, CrossCoherence is able to retrieve more comprehensive and accurate text prompts which are more coherent with the generated shapes. In the supplementary, we provide additional qualitative examples comparing CrossCoherence with CLIP-based metrics.

8. Conclusions, Limitations and Future Work

In this work, we presented the first benchmark for text-to-shape coherence, which includes an automatically improved paired dataset of shapes and texts, **GPT2Shape**, a shape-text coherence quantitative metric, **CrossCoherence**, and a human-validated test set, **HST**. Through extensive comparisons with existing works and a user study, we have quantitatively demonstrated the superior quality of our dataset and evaluation metric. Our benchmark may enable the rapidly growing field of text-driven shape generation to perform accurate comparisons along the important dimension of text-to-shape coherence.


Rectangular wooden table with **two legs**, light brown in color **with 10 drawers, 5 on each side.**

Rectangular brown wooden table with four straight legs.

Rectangular steel chair **with armrests** and four legs, white **wooden finish.**

Grey colored elegant chair with dark grey legs and without armrests.

A brown color wooden table with oval shaped top and **four legs.**

A dark brown circular wooden table with a circular base for support and stability.

This is a tall, brown, rectangular table **with a door that opens on the end.** It is made of wood.

Rectangular wooden table with sharp edges, and a black stripe around the edge, in a brown hue.

Figure 8. Examples of shapes generated by Shap-E with the corresponding text retrieved by CLIP R-precision, on the right, and CrossCoherence, on the left.

Our results suggest that caution should be taken when using metrics that rely on rendered views to rank generative models, such as CLIP-based metrics, as they exhibit failures due to their inability to capture intricate nuances present in both textual descriptions and 3D shapes as well as a strong dependency on the rendering parameters.

While we have demonstrated the potential of leveraging a LLM to improve the quality of the descriptions, it is important to note that our approach did not incorporate the shape as an additional input to the LLM itself. To further enhance descriptions and expand the benchmark beyond the ShapeNet categories covered in Text2Shape, one promising avenue is to explore the use of a Visual-Question-Answering models on 3D shape renderings. This extension may enable even more meaningful descriptions and address a limitation in our current work.

Finally, in light of this limitation, the proposed strategy of text refinement as well as the evaluation metric may be readily extended to a wider set of 3D objects. Interestingly, two contemporary publications [21, 41], released during the writing of this manuscript, proposed methods to build large-scale paired text-shape datasets, which we plan to leverage to further develop our work.

References

[1] Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. Shapeglot: Learning language for shape differentiation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8938–8947, 2019. 3, 6

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 1, 5

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. 3

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 2, 3, 5

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 3

[6] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 100–116. Springer, 2019. 1, 2, 3

[7] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 3

[8] Kevin Crowston. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the Future of ICT Research. Methods and Approaches: IFIP WG 8.2, Working Conference, Tampa, FL, USA, December 13-14, 2012. Proceedings*, pages 210–221. Springer, 2012. 3

[9] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 3

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5

[12] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 1, 5

[13] Zhida Feng, Zhenyu Zhang, Xintong Yu, Yewei Fang, Lanxin Li, Xuyi Chen, Yuxiang Lu, Jiaxiang Liu, Weichong Yin, Shikun Feng, et al. Ernie-vilg 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. *arXiv preprint arXiv:2210.15257*, 2022. 1, 5

[14] Rao Fu, Xiao Zhan, Yiwen Chen, Daniel Ritchie, and Srinath Sridhar. Shapecrafter: A recursive text-conditioned 3d shape generation model. *arXiv preprint arXiv:2207.09446*, 2022. 1, 2, 3, 5, 6, 7

[15] 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. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV*, pages 89–106. Springer, 2022. 1, 5

[16] 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, 6

[17] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 1, 2, 5, 7

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 2

[19] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 1, 5

[20] Zhengzhe Liu, Yi Wang, Xiaojuan Qi, and Chi-Wing Fu. Towards implicit text-guided 3d shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17896–17906, 2022. 1, 2, 5, 6, 7, 8

[21] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *arXiv preprint arXiv:2306.07279*, 2023. 8

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

[23] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 1

[24] 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. 1, 5

[25] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 1, 2, 5, 6, 7

[26] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. 2, 6, 7

[27] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2, 5, 6

[28] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 4, 5

[29] 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. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. 1, 2, 5, 6

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 5

[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, 5

[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, 5

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1, 5

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 5

[35] 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

[36] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 3

[37] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 3

[38] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 3

[39] Michael Tschannen, Manoj Kumar, Andreas Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. *arXiv preprint arXiv:2306.07915*, 2023. 7

[40] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. *arXiv preprint arXiv:2212.14704*, 2022. 1, 2, 5, 6

[41] Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. *arXiv preprint arXiv:2305.08275*, 2023. 8

[42] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 1, 5

[43] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022. 7