ShapeTalk: A Language Dataset and Framework for 3D Shape Edits and Deformations

Panos Achlioptas[1,3] Ian Huang[1] Minhyuk Sung[2]
Sergey Tulyakov[3] Leonidas Guibas[1]

[1]Stanford University [2]KAIST [3]Snap Inc.

Abstract

Editing 3D geometry is a challenging task requiring specialized skills. In this work, we aim to facilitate the task of editing the geometry of 3D models through the use of natural language. For example, we may want to modify a 3D chair model to "make its legs thinner" *or to* "open a hole in its back". *To tackle this problem in a manner that promotes open-ended language use and enables fine-grained shape edits, we introduce the most extensive existing corpus of natural language utterances describing shape differences:* ShapeTalk. ShapeTalk *contains over half a million discriminative utterances produced by contrasting the shapes of common 3D objects for a variety of object classes and degrees of similarity. We also introduce a generic framework,* ChangeIt3D*, which builds on* ShapeTalk *and can use an arbitrary 3D generative model of shapes to produce edits that align the output better with the edit or deformation description. Finally, we introduce metrics for the quantitative evaluation of language-assisted shape editing methods that reflect key desiderata within this editing setup. We note that* ShapeTalk *allows methods to be trained with explicit 3D-to-language data, bypassing the necessity of "lifting" 2D to 3D using methods like neural rendering, as required by extant 2D image-language foundation models. Our code and data are publicly available at* https://changeit3d.github.io/.

1. Introduction

Visual content creation and adaptation, whether in 2D or 3D scenes, has traditionally been a time-consuming effort, requiring specialized skills, software, and multiple iterations. The use of natural language promises to democratize this process and let ordinary users perform semantically plausible content synthesis, as well as addition, deletion, and modification by describing their intent in words – and then letting AI-powered tools translate that into edits of their content. There has been very strong recent interest in and impressive results from large visual language models able to transform text into 2D images, such as *DALL-E 2* from OpenAI, or *Imagen* from Google. The same need exists for 3D asset creation for video games, movies, as well as mixed-reality experiences – though fully automated tools in the 3D area are only now starting to appear [26, 33, 35]. The task of editing 2D or 3D content via language is even more challenging, as references to extant scene components have to be resolved, while unreferenced parts of the scene should be kept unchanged as much as possible.

This work focuses on the task of modifying the *shape* of a 3D object in a fine-grained manner according to the semantics of free-form natural language. Operating directly in a 3D representation has many advantages for downstream tasks that need to be 3D-aware, such as scene composition and manipulation, interaction, etc. Even if only 2D views are needed, 3D provides superior attribute disentanglement and guarantees view consistency. Furthermore, note that modifying the 3D geometry of an object in ways that are faithful to its class semantics is itself a highly non-trivial undertaking (e.g., stretching a sedan should keep the wheels circular) and has been the focus of recent work [44, 45].

Our language-driven shape deformation task is applicable to many real-world situations: e.g., in assisting visually-impaired users, graphic designers, or artists to interact with objects of interest and change them to better fit their design needs. We build a framework, **ChangeIt3D**, to address this task, consisting of three major components: the **ShapeTalk** large-scale dataset with an order of magnitude more utterances than in previous work (Section 2), a modular architecture for implementing edits on top of a variety of 3D shape representations, and a set of evaluation metrics to quantify the quality of the performed transformations.

The ShapeTalk dataset, linking 3D shapes and free-form language, contains over half a million discriminative utterances produced by contrasting pairs of common 3D objects for a variety of object classes and degrees of similarity. Shape differentiation helps focus the language on fine-

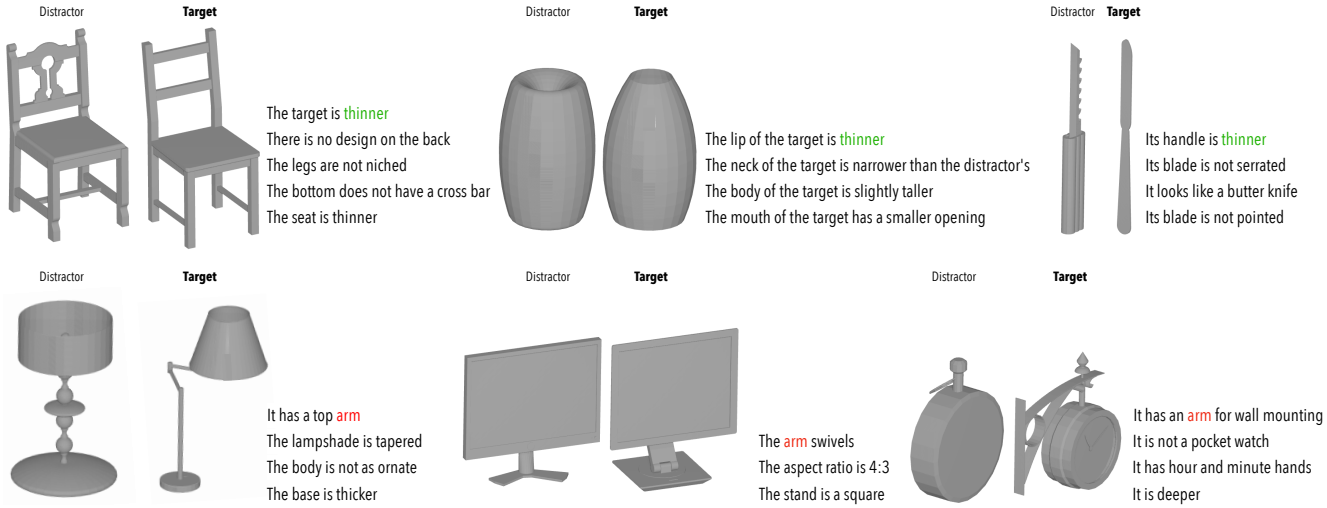


Figure 1. **Samples of contrastive utterances in ShapeTalk.** For each paired distractor-target object, ultra-fine-grained ***shape differences*** are enumerated by an annotator in decreasing order of importance in the annotator's judgment. Interestingly, both *continuous* and *discrete* geometric features that objects share **across** categories naturally emerge in the language of ShapeTalk; e.g., humans describe the "thinness" of a chair leg or a vase lip (top row) or the presence of an "arm" that a lamp or a clock might have (bottom row).

grained but important differences that might not rise to the surface when we describe object geometry individually, as in PartIt [19], where clearly different geometries can end up with very similar descriptions because they share a common underlying structure. Furthermore, unlike the dataset used by ShapeGlot [5], our goal is to obtain as complete descriptions of the geometry differences between two objects as possible, with the goal of enabling reconstruction of the differing object from the reference object and the language – going well beyond simple discrimination. Examples of utterances in ShapeTalk are provided in Figure 1.

We approach the task of language-based shape editing by enabling shape edits and deformations on top of a variety of 3D generative models of shapes, including Point-Cloud Auto-Encoders (PC-AE) [4], implicit neural methods (ImNet) [11] and Shape Gradient Fields (SGF) [8]. To this end, we train a network on ShapeTalk for a discriminative task of identifying the target within a distractor-target pair (examples in Figure 1) and show that the same network can guide edits done directly inside the latent spaces of these generative models. We note that a great deal of ShapeTalk refers to shape parts. Even though the underlying shape representations we deploy do not have explicit knowledge of parts, we demonstrate that our framework can apply a variety of part-based edits and deformations. This confirms a remarkable finding – already described in [24] and [20] – that the notion of parts can be learned from language alone, without any geometric part supervision.

As already mentioned, making edits to an existing shape is more demanding than *ab initio* shape generation as (a) it requires understanding of the input shape and its relation to the modification language, and (b) changes to parts not referenced in the modification utterance should be avoided. Hence, a further contribution of our work is a set of evaluation metrics for the modification success and quality, reflecting realism of the resulting shape, faithfulness to the language instructions, and stability or avoidance of unnecessary changes. Such metrics are essential for encouraging further progress in the field.

In summary, this work introduces ① a new large-scale multimodal dataset, ShapeTalk, with referential language that differentiates shapes of common objects with rich levels of detail, enabling a new setup for doing language-driven shape deformations directly in 3D. We approach the task of language-based shape editing with ② a modular framework supporting diverse 3D shape representations and implementing fine-grained edits guided by a 3D-aware neural-listening network. To set the stage for future developments on the task, we introduce ③ a set of intuitive evaluation metrics for the shape edits and deformations performed.

2. Related Work

Language-Guided Manipulation and Editing. Recently, the wide adoption of CLIP [37] has accelerated attempts to build language-guided editing systems — either for images, or to a lesser degree, for 3D shapes. As CLIP is trained with pairs of images and texts, most recent efforts have been made primarily in the 2D image domain. After DALL-E [38] introduced the idea of creating images from texts using a pretrained CLIP model, subsequent work including StyleCLIP [34], StyleGAN-NADA [15], and Paint by Word [7] extended the idea to edit a given image based on linguistic instructions and guidance. For 3D, Text2Shape [10] and the work by Ma *et al*. [27] pi-

oneered the introduction of a framework for synthesizing 3D shapes and scenes from texts. More recently, CLIP-Forge [40] followed this direction and leveraged CLIP in linking 3D to 2D in a joint embedding. DreamFields [23] and other works [25,35,46] leveraged the same idea to generate NeRF representations without direct image guidance. AutoSDF [31] and ShapeCrafter [14] also introduced text-condition 3D generation models, not exploiting CLIP, but using an autoregressive model learning serialized 3D voxel occupancy maps. These methods, however, focused mainly on *generating* 3D shapes, as opposed to editing, which requires language-shape grounding to resolve the edit descriptions given. Part2Word [42], PartGlot [24], and TGNN [21] made a first step in learning the relations between words and parts in a shape [24, 42]; or words and objects in a scene [21]. However, the part or object localization was not employed for shape editing. While Text2Mesh [30] demonstrated language-guided 3D shape manipulation, its editing is limited to adding vertex colors and displacing vertex positions, producing shape changes primarily at the level of textures, or at the global shape level. In contrast, our work introduces a modular framework for editing the global or local *geometry* of a 3D object *in a fine-grained manner*, and introduces the first metrics that can measure the efficacy of language-assisted editing systems directly in 3D.

Language-Shape Datasets. While there is an increasing number of novel datasets providing referential language grounded on 3D models, overall visio-linguistic data for 3D remain sparse. Some notable examples include ShapeGlot [5], SNARE [43], and PartIt [19]. A quick comparison between these and ShapeTalk can be found in Table 1.

With $536k$ utterances and a collection of $\sim$36k shapes, our dataset, ShapeTalk, provides an order of magnitude more utterances than the runner-up (ShapeGlot [5], with $\sim$79k references) and more than 3 times the number of shapes than the runner-up (PartIt [19], with $\sim$10k shapes). More importantly, ShapeGlot [5], SNARE [43], and PartIt [19] do not provide a complete enumeration of most salient differences each distractor-target pair has, whereas our dataset does. In conjunction with the fact that language naturally under-specifies the complete set of required changes, ShapeTalk is better equipped to handle the task of language-assisted shape editing. Table 5 highlights a key benefit of having complementary utterances for the same object pair: neural-based discrimination of the target is greatly boosted. While more large-scale datasets connect language to 3D, their focus is either on 3D scenes [1–3, 12, 17, 28] or videos [29], instead of prioritizing individual objects like ShapeTalk does.

Dataset	# utter.	# 3D models	Multi-Categ.	Multi-Utter.
ShapeGlot [5]	$79k$	$5k$	No	No
SNARE [43]	$50k$	$8k$	Yes	No
PartIt [19]	$10k$	$10k$	Yes	No
ShapeTalk (Ours)	$536k$	$36k$	Yes	Yes

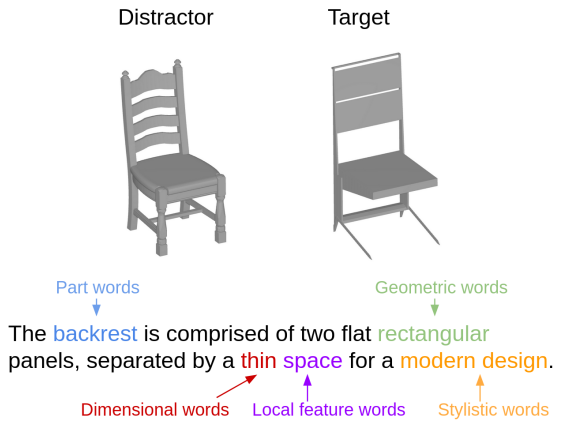
Table 1. A comparison of ShapeTalk with other preexisting datasets that capture language and 3D geometry data. ShapeTalk has more utterances, more shapes than any existing dataset, and moreover provides multiple utterances for paired shapes.

3. The ShapeTalk Dataset

We refer to a distractor-target tuple (see examples in Figure 1) as a communication context, or context for short. In order to elicit diverse and fine-grained contrastive language, ShapeTalk incorporates two types of contexts – **Hard** and **Easy** – based on a notion of shape-wise similarity derived from monochromatic rendered images of meshes and an L2 distance in the latent space of an ImageNet-pretrained [13] classifier [18]. Hard contexts are formed with pairs of objects with the highest possible similarity within an object class. Easy contexts are chosen out of pairs of objects with typical (average) similarity within their class. ShapeTalk shapes are aggregated from ShapeNet [9], ModelNet [48] and PartNet [32]. After removing duplicates and models of poor quality, ShapeTalk provides discriminative utterances for a total of **36,391** shapes, across **30** object classes. Overall, ShapeTalk contains **73,799** distinct contexts (48.6% of which are Hard), and a total of **536,596** utterances (averaging **7.27** utterances per distinct context).

Creating linguistic descriptions capturing *all exact* differences between two shapes is a daunting task, because of the brevity and intrinsic ambiguity of language. The amount of specificity required is typically overwhelming even for modestly differently-looking objects, as shown in our pilot AMT experiments (Supp. Mat. Fig. 2). Thus, instead of demanding our annotators to describe all the differences between the shapes, we ask them to *enumerate* discriminative differences, up to a maximum number, in decreasing order of "obviousness", whether visual or linguistic. Specifically, we instruct the 2,161 annotators of ShapeTalk to provide descriptions that differentiate the two shapes within a context, and do so class-by-class. This latter design choice lessens their cognitive burden allowing them to transfer experience of annotating past recent examples within the same shape class. It is also worth noting that for each class, we provide visual examples of objects with annotations of part-names as well as names for different shape styles (e.g. "bowler" hat vs. "ivy" hat), without requiring their usage in the annotations. Typical resultant annotations of ShapeTalk can be seen in Figure 1.

ShapeTalk utterances are highly diverse. To shed light on the types of language used in ShapeTalk, we manually



Category	Examples
Part	legs, handles
Local	edges, points, holes
Stylistic	modern, classic
Dimensional	big, small, long, short
Geometric	hexagonal, triangular

Figure 2. Word categories. As this utterance references both parts and local features, it is *not* "holistic".

class	part	local	holistic	stylistic	dimensional	geometric
all	0.80	0.05	0.19	0.06	0.36	0.25
all easy	0.78	0.04	0.21	0.06	0.32	0.25
all hard	0.82	0.06	0.17	0.05	0.39	0.25

Table 2. The proportion of utterances containing different types of information, analyzed based on the words that they contain. The stats across categories are shown in Supp. Mat.

curate a large subset of *word* groups from user utterances into 5 different categories shown in Figure 2. We connect an utterance to these categories according to word membership. Note that in addition to these categories, an utterance can also contain "holistic" shape information if it does not reference any **Parts** or **Local** features, but rather describes the whole shape. Table 2 shows the proportion of utterances that contain different kinds of information, according to word membership.

Note that across all the classes, most utterances refer to shape parts (80%), with descriptions often specifying dimensional (36%) and geometric characteristics (25%), whereas stylistic information is rarer (5%). However, the distribution depends also on the category (see Supp. Mat.). 59% of utterances that reference parts and 53% of utterances about local features provide details about style, dimensionality, geometric shape. A higher percentage (71%) of holistic descriptions have similar qualifications.

Table 3 shows the number of utterances that contain joint information about different kinds of characteristics and different levels of visual granularities. For parts and holistic utterances, far more descriptions specify dimension than style and shape combined. For local features, however, geometric details are more popular than style and dimension words combined. Table 4 shows that language becomes more fine-grained in Hard contexts than for Easy contexts. In the hard context, utterances are more likely to reference parts, local features and dimensions. On the other hand, in Easy contexts, holistic and stylistic language are more common. The discrepancy can be further shown across different visual granularities, where the proportion of utterances that contain information about style, dimension or geometric shape is always higher for Hard contexts than Easy ones. As such, Easy and Hard contexts provide a sensible way to vary language granularity. More details regarding the curation process, the collected data, and its analysis can be found in the online Supplemental Materials [6].

	Stylistic	Dimensional	Geometric
parts	0.039	0.269	0.202
locals	0.004	0.009	0.016
holistic	0.018	0.084	0.041

Table 3. Proportion of utterances containing joint information about a certain level of visual granularity (holistic, parts, local) and different characteristics (stylistic, dimensional, geometric).

	All	Easy	Hard
parts	0.59	0.56	0.63
locals	0.53	0.50	0.55
holistic	0.71	0.70	0.73

Table 4. Proportion of utterances containing information about style, dimensions, or geometric shape at the three levels of visual granularity, for Easy, Hard and all contexts.

4. Desiderata for Language-Assisted Visual Edits & Evaluation Metrics

In this section, we present intuitive evaluation metrics to measure the quality of an arbitrary method doing language-assisted 3D shape editing. Importantly, it must be highlighted that delicate ambiguities must be navigated when assessing the output of such methods. For instance, language descriptions are naturally ambiguous, especially when describing shape changes (e.g., exactly how thin are "thinner legs"?). Additionally, how should we measure if a method achieves a minimal edit, as opposed to causing drastic edits even when the description itself demands an edit in a localized part? Last, how should we compare the semantic alignment of different outputs with respect to the same description? With these questions in mind, we present the metrics below, including human evaluation results showing their correlation with human perception (see [6]).

1. **Linguistic Association Boost (LAB)**. If the applied visual change reflects the semantics of the language, then the modified item should have higher visio-linguistic associa-

tion with the input instruction than the original, unmodified item. **Algorithm**: Use a pretrained neural listener (a network finding the target shape from the distractors given a query utterance) to measure the *difference* in the predicted association score between each of the two (input/output) items and the instruction.
Assuming an oracle neural listener, assessing if it finds the changed item more compatible with the language than the input is an obvious choice – but, it does not explicitly address the under-specificity problem of the language mentioned above. All the following metrics constrain the space of options in the change space, and as such LAB should always be considered together with at least one of them (preferring Pareto-optimal methods in the combined metrics).
2. **Geometric Difference (GD)**. For two output shapes derived by the same input shape and language, and with both scoring equally in LAB, it makes sense (on average) to prefer the shape that is shape-wise more similar to the input. For such an output is more likely to have preserved the overall *identity* of the input, *possibly* in areas that are not referred in the language. **Algorithm**: Here one can use any pairwise shape-related distance, such as the commonly used Chamfer [4], or arbitrarily more complex ones [16].
3. ***localized*-Geometric Difference (*l*-GD).** If at test time we have access to the geometry of the referenced shape elements (e.g., part information) involved in the change for both the original and resulting shapes, it makes sense to *remove* them from the calculation of the above metric. I.e., in such a case, it is acceptable to permit the output to arbitrarily differ from the input on the referred part(s) – but not in others. **Algorithm**: Given a set of linguistic instructions concerning specific semantic parts and a segmentation algorithm that can deduce semantic parts of shapes, use it to predict the shape parts of the input and output, and remove all parts mentioned in language before computing the GD.
4. **Class Distortion (CD)**. Our last metric is complementary to the previous two metrics that focus on the 'minimality' of change and aims at constraining the solution space of possible outputs differently. Simply put, CD expects the output to preserve the input shape's *class*, prioritizing thus *realistic* looking outputs. The necessary underlying assumption here is the use of language that reflects differences among same-class objects (which is the case for ShapeTalk). **Algorithm**: Given a shape classifier (typically another pretrained network), compute the absolute *difference* of the probability assigned to the underlying class between the input/output shapes. We remark that many distance functions for probabilities, e.g., EMD, KL-Divergence, etc., can be used here to measure shape-class deviations among the input/output, making this metric more similar to existing ones for generation quality (e.g., Inception Score [39])[1].

[1]Unlike *ab initio* generation however, the conditional nature of *ChangeIt3D* allows us to directly compare each input-output pair of a test collection, and in our limited exploration we did not observe a significant benefit by using different probability distances.

5. Method

Figure 3 shows a high-level overview of our framework named **ChangeIt3D**. Specifically, in this framework we link a 3D generative model G (here instantiated as an autoencoder) to a *neural listener* L, a discriminative network that, given an utterance and a pair of input shapes, assigns high probability to the most utterance-compatible shape within the pair. See [5, 24] for such listeners. L is used to guide the 3D generation by "editing" the input shape in G's latent space to be more utterance-compatible. This high-level idea has recently been used in image editing work [34].

For modularity, we train the editing process via a two-stage approach. As the generative model G needs to capture sufficient geometric information, we pretrain an autoencoder during the first stage to achieve good reconstructions. Once G is pretrained, L is trained to associate higher utterance-compatibility with the *target* shape than with the *distractor* shape, using the latent representations given by the pretrained network G as input.

In the second stage of our approach, we link the frozen networks L and G together via the Shape Editor E, a low-capacity network that learns to find editing latents in the space of G through predicting an update vector by regressing its magnitude and direction independently. This 'update vector' is then applied onto the source shape latent representation *additively*, promoting a direct interaction between the source representation and the underlying edit's latent. During this stage, our model uses frozen weights for the encoder, decoder and neural listener L, and learns the weights for the shape editor E so as to 1) preserve similarity to the original input shape through regularization of the update magnitude and 2) maximize the L-evaluated utterance compatibility of the updated shape latent representation over that of the original shape.

We note that the use of the original shape for L's comparative context (here termed "self-contrast") is extremely important — we find that the alternative approach of replacing it with the ground-truth 'target' shape has a negative influence, frequently leading to violation of similarity preservation in order to achieve high language compatibility (see Table 7). Figure 3 shows the *self-contrast* setup.

For baselines, we experiment with an alternative variant called the **Monolithic Model**, where the system is optimized to *reconstruct* the ground truth target shape. Given a context (*i.e.* a pair of target and distractor shapes) and an utterance, an encoding of the distractor is fused together with a learned representation of the utterance , and an MLP-based decoder is used to output a predicted target reconstruction. While this benefits from being trained end-to-end

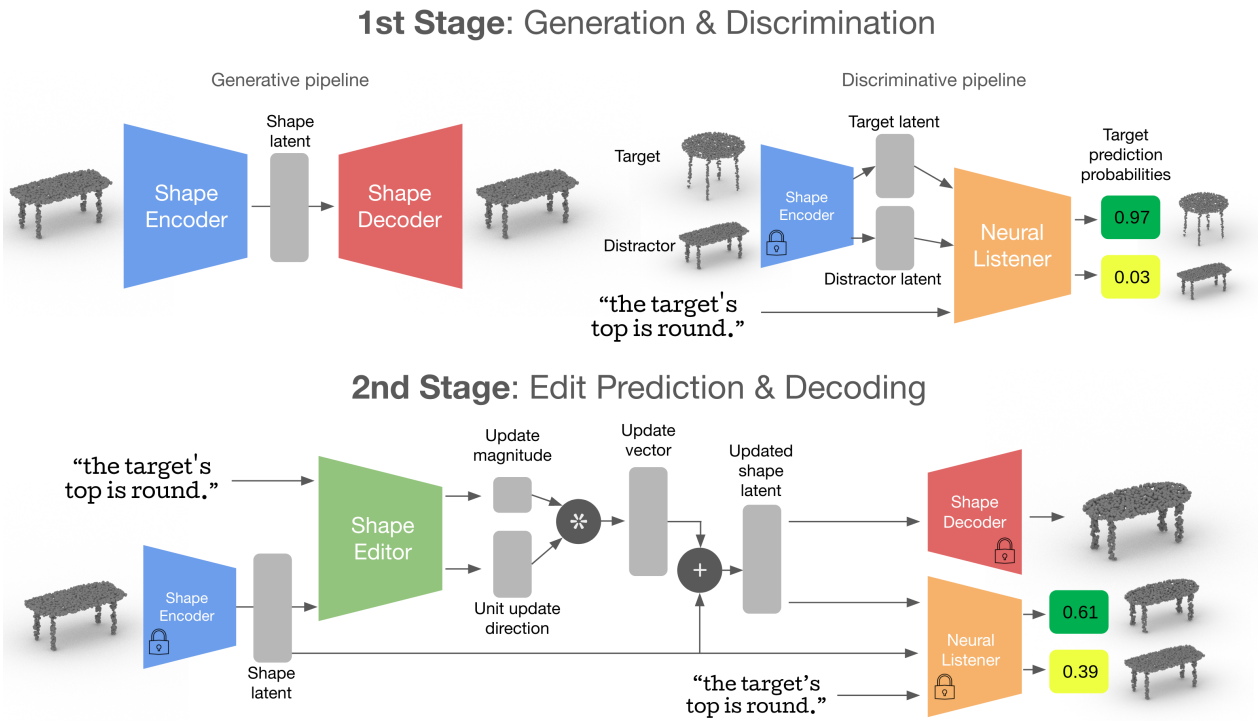


Figure 3. **Overview of *ChangeIt3D*, our modular framework for language-driven shape editing.** In Stage 1, we pretrain a shape autoencoder for shapes (using traditional reconstruction losses), freeze the encoder and use the encoded latents of the target and distractor to pretrain a neural listener (using classification losses). In Stage 2, we use the pretrained autoencoder and neural listener to train a shape editor module to edit shapes within the encoded latent space in a way that is both consistent with the language instruction and also minimal. All modules with locks indicate frozen weights.

with a well-studied reconstruction loss, such as the Chamfer loss [4], it suffers more from the ambiguity of language, as many changes to the input shape are undescribed by the language input but nonetheless expected by the reconstruction task. To combat this, we input to this system the complete collection of utterances acquired for each context, concatenated in the annotator-specified order.

Finally, we also implement a shape retrieval model (called **Neighbor Search**) that uses the trained encoder within G and the trained shape editor E to predict an updated latent code. Then, instead of decoding, we retrieve a nearest-neighbor among shapes in the training examples. This allows us to inspect the quality of E decoupled from the quality of the underlying decoder of ChangeIt3D (which can bottleneck the output quality [47]). Refer to Supp. Mat. for implementation details regarding all ablated models.

6. Experimental Results

We separate *the shapes* involved in ShapeTalk into disjoint train/validation/test sets that include 85%, 5%, 10% of all underlying shapes, respectively. Our neural listeners, 3D editors, and shape reconstruction (generative) networks follow proper variations of these splits.

6.1. Neural Comprehension of Referential Language for 3D Shapes

Backbone	Overall	*Subpopulations* Easy	Hard	First	Last	Multi
ImNet	68.0	72.6	63.4	72.4	64.9	78.4
SGF	70.7	75.3	66.1	74.9	68.0	79.9
PC-AE	**71.3**	**75.4**	**67.2**	**75.2**	**70.4**	**81.5**
CLIP	75.5	78.5	72.4	79.5	72.2	85.8

Table 5. **Neural comprehension performance**. *Backbone* indicates the underlying latent space. *Overall* reports average prediction accuracy as percent of the test set across all ShapeTalk classes. *Subpopulations* show the average accuracy among subsets of the test data, i.e. *Easy*: less visually similar pairs, *Hard*: pairs having the highest visual similarity, *First*: utterances indicated as most salient by an annotator (uttered first), *Last*: least salient utterances. *Multi* reports the accuracy when we train/test listeners with the input being all utterances given by an annotator in a context.

For our neural listeners we use the context-free architecture introduced in ShapeGlot [5] throughout this work. One salient change we apply to it is replacing its LSTM-based utterance encoder with a Transformer-based architecture similar to what was done in PartGlot [24]. With the lis-

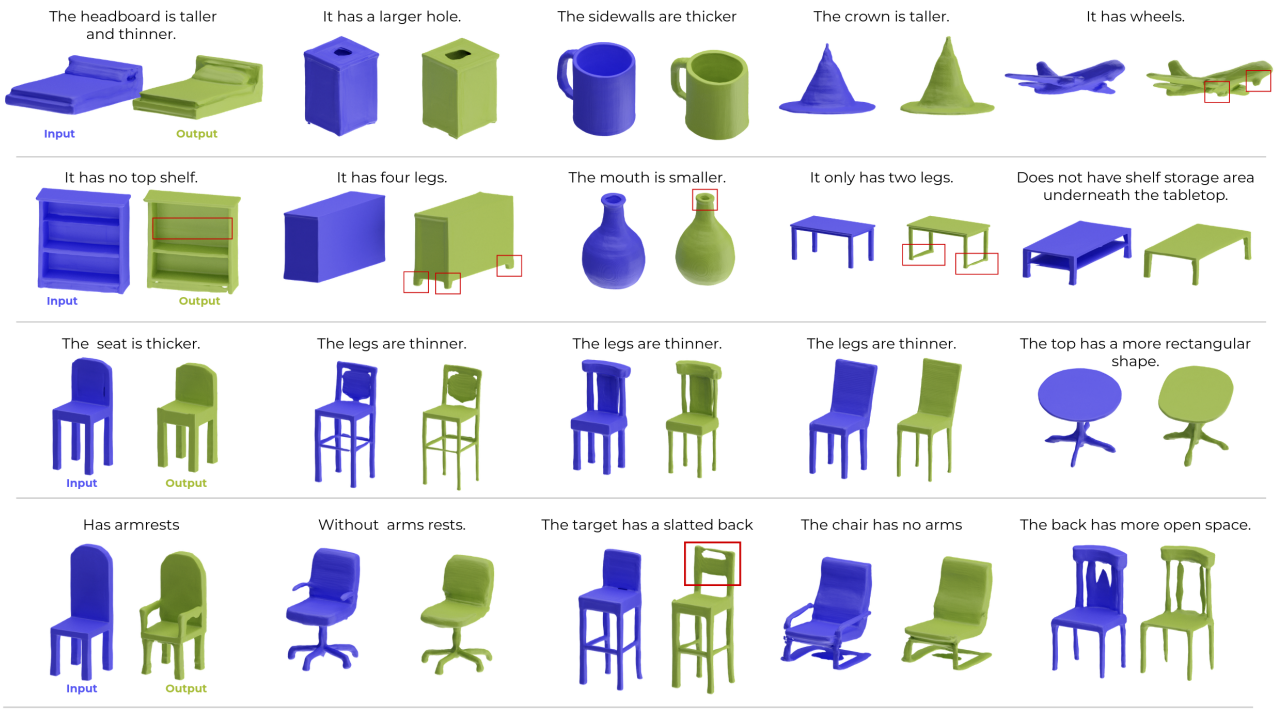


Figure 4. **Qualitative edits produced by ChangeIt3D**. The results are based on an ImNet backbone AE, with the meshes extracted using marching cubes. The achieved edits, covering a large variety of classes, are oftentimes local and fine-grained, deal with both discrete and continuous changes, and at times entail high-level and complex shape understanding. Remarkably, while the utterances invariably refer to shape parts, these edits derived by ChangeIt3D do **not** utilize any explicit knowledge of local shape geometry (part-like, or otherwise), but instead build solely on the implicit bias of training with referential language.

tening architecture being fixed, we ablate its performance when operating with latent shape representations derived by three widely adopted autoencoders (AEs): IMNet [11], ShapeGF (SGF) [8], and PC-AE [4]. These AEs use different 3D shape representations (e.g, input pointclouds vs. implicits), thereby offering flexibility to our approach.

Specifically, Table 5 shows the percentage accuracies for the neural-listening comprehension task with each of the pretrained AE backbones. On average, the accuracy for the PC-AE backbone is the highest, and close to that of the SGF, both overall, and in almost all of the subpopulations. Surprisingly, it appears that ImNet fails in comparison to capture some of the fine-grained characteristics that are necessary for shape discrimination within its latent space. Interestingly, the "obviousness" captured by the order of the utterance enumeration can also be seen in the neural-listening results. Namely, for all three methods, we find that methods tested on the first-enumerated utterances perform better compared to when tested with the lastly-enumerated ones. Furthermore, the *complementary* nature among the different utterances of the same context in ShapeTalk is clear – for all generative pipelines, neural-listening accuracy is significantly higher (an increase of **10.2%** for PC-AE) when operating with a concatenation of all context utterances (*Multi* column). In addition to the above table, we remark that when using a publicly available pretrained CLIP model [37] without fine-tuning it on ShapeTalk, its accuracy was **53.0%**, close to random guessing. This suggests that the fine-grained language and shape-differences within ShapeTalk are not encompassed in the large-scale multimodal dataset CLIP was trained on. On the other hand, by training the neural-listener within the latent space of a large pre-trained Open-CLIP model [22] with ShapeTalk and monochromatic 2D renderings of its shapes, we see significant performance gains (last row, Table 5).

6.2. Editing Experiments

Given the performance advantage of PC-AE in the listening-comprehension experiments (Table 5) we will use it to continue our editing-based study. Edit results based on the other two generative pipelines are also provided in the Supp. Mat. [6]. Moreover for the quantitative analysis of our editing experiments we will focus on three classes of ShapeTalk: **chairs**, **tables** and **lamps**. These classes have the advantage of being relatively large, and come with part annotations, enabling the deployment of all our

evaluation metrics. Finally, since our CD and *l*-GD metrics require an oracle shape classifier and a part segmentation model, respectively, we use the same splits used for ChangeIt3Dto train: 1) **for CD:** a PointNet-based [36] object classifier predicting the 30 classes of ShapeTalk, and operating with surface-extracted pointclouds of 2048 points per shape. This model achieves an average prediction accuracy of 89.0%. 2) **for *l*-GD:** a PointNet-based architecture adapted for the prediction of semantic parts of 3D pointclouds, using pointclouds with 2500 points per shape and part labels extracted from the ShapeNetParts [49].

Table 6 shows an evaluation of the Monolithic and Search-based baselines against ChangeIt3D. Our model's ability to preserve details from the source shape (minimal GD score) while maximizing the LAB score demonstrates a general ability to edit to follow language descriptions. Specifically, the reconstruction-based monolithic model does poorly on the LAB metric, as learning to reconstruct based on language is a difficult problem without capturing all of shape change information within utterances, which rarely happens even when utterances are concatenated. As such, the associated learning signals for this task can be noisy, leading to it producing shapes that neither preserves similarity and nor increases compatibility with the language. Meanwhile, our search-based baseline has a higher LAB score, demonstrating that the shape editor module learns to produce updated latents that capture the described characteristics. However, its GD and l-GD metrics tells us that this in general produces large variations in the shape as a whole and the identity of the input shape is largely violated. As expected, the CD metric is lowest with this baseline which outputs ground-truth training examples. Figure 4 shows qualitative examples of decoded shape-edits with our ImNet-based models. These examples showcase the ability of ShapeTalk and ChangeIt3D to give rise to nuanced yet semantically correct edits to shapes. Moreover, our method appears to be able to preserve the overall identity of the input shape, and oftentimes create localized/minimal shape edits that can cover both structural and continuous changes. Future work, like methods proposed by [20], should be used to further improve the locality of such edits. More examples, including failure cases, are in the Supp. Mat.

Table 7 shows the evaluation metrics on variants of the ChangeIt3D architecture when varying whether the shape editor module predictions are decoupled (expressed as a product of a magnitude and a unit norm latent direction) and whether the neural listener compares the update candidate with the original distractor (self-contrast) or the groundtruth target. Our results show that a decoupled shape editor performs better, and that self-contrast significantly improves identity preservation (GD), CD and localized GD, but sacrifices the utterance-compatibility (LAB). Last, we observe that using self-contrast has a positive effect in reducing mode-collapse [4]. The output shape *distribution* of such editors tends to be closer to the groundtruth of each shape category as indicated by the Fréchet pointcloud distance [41] measurement (FPD column).

Baselines	**GD** (↓)	***l*-GD** (↓)	**CD** (↓)	**LAB** (↑)
Monolithic	0.76	40.40	20.1	2.5
Neighbor-Search	0.51	1.02	**4.5**	12.0
ChangeIt3D	**0.33**	**0.99**	5.4	**35.1**

Table 6. **Quantitative comparisons for three baselines.** A *'monolithic'* approach that learns how to reconstruct the target from the distractor based on *all* linguistic differences expressed by an annotator; **vs.** a search-based approach that instead of decoding a reconstruction, finds its closest *training example* in the generator's latent space **vs.** our final, modular approach that disentangles the generation from the discrimination problems (*ChangeIt3D*).

Decoupled	*Self Contrast*	**GD** (↓)	***l*-GD** (↓)	**CD**(↓)	**FPD** (↓)	**LAB** (↑)
✗	✗	0.45	1.12	8.3	42.0	43.4
	✓	0.33	1.09	5.4	22.9	34.9
✓	✗	0.43	1.17	8.0	38.6	**43.5**
	✓	**0.33**	**0.99**	**5.4**	**22.4**	35.1

Table 7. **Ablations for *ChangeIt3D*.** We report the effect of two choices. First, we measure the effect of decoupling the produced editing-latent in a unit-norm direction latent and a scalar-magnitude, instead of a single joint latent (*Decoupled*). Second, we report the effect of applying the listening-based loss between the input distractor and its edited version (*Self-contrast*) vs. contrasting the edited version against a separate ground-truth target from ShapeTalk. We use the metrics of Section 4, on averages over three classes (chair, table, lamp). GD and *l*-GD are based on Chamfer distance, scaled by 10e2; LAB and CP are percentages.

7. Conclusion

In this paper we have introduced a new dataset for the task of language-assisted 3D shape editing, ShapeTalk, which is an order of magnitude larger that any other comparable dataset. We have illustrated the potential of this data by experimenting with ChangeIt3D, a highly modular framework for coupling a neural listener with an arbitrary 3D latent representation; and propose intuitive evaluation metrics for the underlying task. We hope that future works will build on this foundation and make language-guided 3D shape editing widely accessible and useful.

Acknowledgements. We thank the anonymous reviewers for their valuable comments. This work is funded by an ARL grant W911NF-21-2-0104, a Vannevar Bush Faculty Fellowship, and gifts from the Snap and Adobe corporations. M. Sung acknowledges the support of the NRF grant (RS-2023-00209723) and IITP grant (2022-0-00594) funded by the Korean government (MSIT), and grants from Adobe, ETRI, KT, and Samsung Electronics.

References

[1] Ahmed Abdelreheem, Kyle Olszewski, Hsin-Ying Lee, Peter Wonka, and Panos Achlioptas. ScanEnts3D: Exploiting phrase-to-3d-object correspondences for improved visio-linguistic models in 3d scenes. *Computing Research Repository (CoRR)*, abs/2212.06250, 2022. 3

[2] Panos Achlioptas. *Learning to generate and differentiate 3D objects using geometry & language*. PhD thesis, Stanford University, 2021. 3

[3] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. ReferIt3D: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[4] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Learning representations and generative models for 3D point clouds. In *International Conference on Machine Learning (ICML)*, 2018. 2, 5, 6, 7, 8

[5] Panos Achlioptas, Judy Fan, Robert XD Hawkins, Noah D Goodman, and Leonidas J. Guibas. ShapeGlot: Learning language for shape differentiation. In *International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 5, 6

[6] Panos Achlioptas, Ian Huang, Minhyuk Sung, Sergey Tulyakov, and Leonidas Guibas. *Supplementary Material for ChangeIt3D: Language-Assisted 3D Shape Edits and Deformations*, (accessed March 1st, 2023). Available at https://changeit3d.github.io/materials/changeIt3D_supplemental_material.pdf, version 1.0. 4, 7

[7] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *Computing Research Repository (CoRR)*, abs/2103.10951, 2021. 2

[8] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge J. Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 7

[9] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing 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. *Computing Research Repository (CoRR)*, abs/1512.03012, 2015. 3

[10] 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. *Computing Research Repository (CoRR)*, abs/1803.08495, 2018. 2

[11] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 7

[12] Z. Dave Chen, Angel X. Chang, and Matthias Nießner. ScanRefer: 3D object localization in RGB-D scans using natural language. *Computing Research Repository (CoRR)*, abs/1912.08830, 2019. 3

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 3

[14] Rao Fu, Xiao Zhan, Yiwen Chen, Daniel Ritchie, and Srinath Sridhar. ShapeCrafter: A recursive text-conditioned 3d shape generation model. *Computing Research Repository (CoRR)*, abs/2207.09446, 2022. 3

[15] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *Computing Research Repository (CoRR)*, abs/2108.00946, 2021. 2

[16] Vignesh Ganapathi-Subramanian, Olga Diamanti, and Leonidas J. Guibas. Modular latent spaces for shape correspondences. *Computer Graphics Forum*, 2018. 5

[17] Ankit Goyal, Kaiyu Yang, Dawei Yang, and Jia Deng. Rel3D: A minimally contrastive benchmark for grounding spatial relations in 3D. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Computing Research Repository (CoRR)*, abs/1512.03385, 2015. 3

[19] Yining Hong, Qing Li, Song-Chun Zhu, and Siyuan Huang. VLGrammar: Grounded grammar induction of vision and language. *International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[20] Ian Huang, Panos Achlioptas, Tianyi Zhang, Sergey Tulyakov, Minhyuk Sung, and Leonidas Guibas. LADIS: Language disentanglement for 3d shape editing. In *Findings of EMNLP*, 2022. 2, 8

[21] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *AAAI Conference on Artificial Intelligence*, 2021. 3

[22] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. *OpenCLIP*, 2021. 7

[23] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[24] Juil Koo, Ian Huang, Panos Achlioptas, Leonidas J. Guibas, and Minhyuk Sung. PartGlot: Learning shape part segmentation from language reference games. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 5, 6

[25] Han-Hung Lee and Angel X Chang. Understanding pure CLIP guidance for voxel grid nerf models. *arXiv preprint arXiv:2209.15172*, 2022. 3

[26] Zhengzhe Liu, Yi Wang, Xiaojuan Qi, and Chi-Wing Fu. Towards implicit text-guided 3d shape generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[27] Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, Sören Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong,

Leonidas Guibas, and Hao Zhang. Language-driven synthesis of 3D scenes from scene databases. In *ACM SIGGRAPH Asia*, 2018. 2

[28] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *Computing Research Repository (CoRR)*, abs/2210.07474, 2022. 3

[29] Willi Menapace, Aliaksandr Siarohin, Stéphane Lathuilière, Panos Achlioptas, Vladislav Golyanik, Elisa Ricci, and Sergey Tulyakov. Plotting behind the scenes: Towards learnable game engines. *Computing Research Repository (CoRR)*, abs/2303.13472, 2023. 3

[30] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. *Computing Research Repository (CoRR)*, abs/2112.03221, 2021. 3

[31] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. AutoSDF: Shape priors for 3d completion, reconstruction and generation. In *CVPR*, 2022. 3

[32] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[33] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *Computing Research Repository (CoRR)*, abs/2212.08751, 2022. 1

[34] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of stylegan imagery. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 5

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

[36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 8

[37] 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. *Computing Research Repository (CoRR)*, abs/2103.00020, 2021. 2, 7

[38] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *International Conference on Machine Learning (ICML)*, 2021. 2

[39] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 5

[40] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. CLIP-Forge: Towards zero-shot text-to-shape generation. *Computing Research Repository (CoRR)*, abs/2110.02624, 2021. 3

[41] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. *Computing Research Repository (CoRR)*, abs/1905.06292, 2019. 8

[42] Chuan Tang, Xi Yang, Bojian Wu, Zhizhong Han, and Yi Chang. Part2Word: Learning joint embedding of point clouds and text by matching parts to words. *Computing Research Repository (CoRR)*, abs/2107.01872, 2021. 3

[43] Jesse Thomason, Mohit Shridhar, Yonatan Bisk, Chris Paxton, and Luke Zettlemoyer. Language grounding with 3d objects. *Computing Research Repository (CoRR)*, abs/2107.12514, 2021. 3

[44] Mikaela Angelina Uy, Jingwei Huang, Minhyuk Sung, Tolga Birdal, and Leonidas Guibas. Deformation-aware 3d model embedding and retrival. In *European Conference on Computer Vision (ECCV)*, 2020. 1

[45] Mikaela Angelina Uy, Vladimir G. Kim, Minhyuk Sung, Noam Aigerman, Siddhartha Chaudhuri, and Leonidas Guibas. Joint learning of 3d shape retrieval and deformation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[46] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. CLIP-NeRF: Text-and-image driven manipulation of neural radiance fields. *Computing Research Repository (CoRR)*, abs/2112.05139, 2021. 3

[47] Zbigniew Wojna, Vittorio Ferrari, Sergio Guadarrama, Nathan Silberman, Liang-Chieh Chen, Alireza Fathi, and Jasper Uijlings. The devil is in the decoder: Classification, regression and gans. *International Journal of Computer Vision*, 127(11):1694–1706, 2019. 6

[48] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[49] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas J. Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (TOG)*, 2016. 8