ChangeIt3D: Language-Assisted 3D Shape Edits and Deformations

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

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

Abstract. In this work, we address the task of *Language-Assisted 3D Shape Edits and Deformations* (which we name **ChangeIt3D**). Given a 3D representation of an object and free-form natural language describing desired changes or modifications to the shape of the object, the task is to transform the input object's geometry in a manner that reflects the requested changes – for example, 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 way that promotes open-ended language usage allowing fine-grained shape edits, we introduce the largest existing corpus of natural language describing shape differences, which we call **ShapeTalk**. This dataset contains over half a million discriminative utterances produced by contrasting the shapes of pairs of common 3D objects for a variety of object classes and degrees of similarity. We introduce metrics for the quantitative evaluation of language-assisted shape editing methods that reflect key desiderata within this editing setup. We also design an effective and modular framework for ChangeIt3D that can combine an arbitrary 3D generative model of shapes with our in-house, ShapeTalk-based, text-to-shape neural listener. Crucially, our modules are trained and deployed directly in a latent space of 3D shapes, bypassing the ambiguities of "lifting" 2D to 3D when using extant foundation models and thus opening a new avenue for 3D object-centric manipulation through language.

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 language promises to democratize this process and let ordinary users perform semantically plausible content addition, deletion, and modification by simply describing their intent in words – and then letting AI-powered tools translate that into edits of their 3D assets. Given the current interest in making the Metaverse useful for many people, such language-based tools are an essential future technology[4].

Corresponding author, wepage: Panos Achlioptas, `https://changeit3d.github.io`
[4] An example is the AI builder Bot recently proposed by Meta AI.

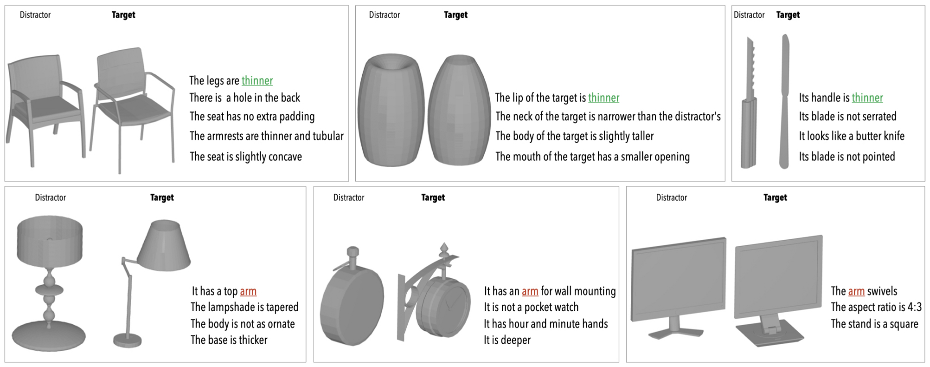

Distractor
Target
The legs are thinner
There is a hole in the back
The seat has no extra padding
The armrests are thinner and tubular
The seat is slightly concave
Distractor
Target
The lip of the target is thinner
The neck of the target is narrower than the distractor's
The body of the target is slightly taller
The mouth of the target has a smaller opening
Distractor
Target
Its handle is thinner
Its blade is not serrated
It looks like a butter knife
Its blade is not pointed
Distractor
Target
It has a top arm
The lampshade is tapered
The body is not as ornate
The base is thicker
Distractor
Target
It has an arm for wall mounting
It is not a pocket watch
It has hour and minute hands
It is deeper
Distractor
Target
The arm swivels
The aspect ratio is 4:3
The stand is a square

Fig. 1: **Samples of contrastive language in ShapeTalk.** In each sub-box *shape differences* between a target and a distractor object of the same class are enumerated by an annotator (by decreasing order of importance in the annotator's judgement). Interestingly, both *continuous* and *discrete* geometric features that shapes share **across** categories emerge in the language of ShapeTalk; e.g., humans describe the "**thinness**" of a chair leg or of a vase lip or of a blade handle (top boxes), or the presence of a semantic-part ("**arm**") that a lamp or a clock or a monitor can have (bottom boxes).

This work focuses on the task of transforming 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 3D-awareness, such as scene composition. Even if only 2D views are needed, 3D provides superior attribute disentanglement and guarantees view consistency. 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 [42,43].

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 term this task as **ChangeIt3D** and build a framework to address it, consisting of three major components: a 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 deformations.

To link 3D shapes and free-form language we introduce the **ShapeTalk** dataset with 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-grained but important differences, differences that might not rise to the surface when we describe object geometry individually, as in the PartIt work [19], where clearly different geometries can end with very similar descriptions because they share a common underlying structure. Furthermore, unlike the dataset used by Shape-

Glot [4], our goal here 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 discrimination. Examples of utterances in ShapeTalk are provided in Figure 1.

We approach ChangeIt3D by enabling shape edits and deformations on top of a variety of 3D generative models of shapes, including Point-Cloud Auto-Encoders (PC-AE) [3], implicit neural methods (ImNet) [10] and Shape Gradient Fields (SGF) [6]. We tackle this task by implementing a ShapeTalk-trained text-to-shape neural listener [4,24] (a network taking a query utterance and discriminating the target from the distractors), learned directly in a latent space of 3D shapes. We note that a great deal of the ShapeTalk dataset refers to shape parts. Even though the underlying shape representations we deploy do not have explicit knowledge of object 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] – that the notion of parts can be learned from language alone, without any geometric part supervision.

Making edits to an existing shape is more demanding than *ab initio* shape generation as (a) it requires understanding of the source 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 ChangeIt3D—① a new setup for doing language-driven shape deformations directly in 3D. The task is enabled by ShapeTalk—② our new large multimodal dataset with referential language, which differentiates shapes of common objects with rich level of detail. We approach this task with ③ a modular framework supporting a variety of 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 ChangeIt3D task we introduce ④ a set of intuitive evaluation metrics for the shape edits and deformations performed.

2 Related Work

3D Deformation Learning. There is a large body of literature for learning deformations of 3D shapes targeting various applications. These include target fitting [22,25,45,17,52], finding shape correspondences [16,15], doing deformation transfer [12,39] and shape completion [26], or performing deformation-aware shape retrieval [42,43]. Also, there have many works for learning free-form deformations (offsets or new positions of points) [53,22,25,45,17], learning flows/trajectories via ODE [49,23,20,6], or learning deformations based on a given cage mesh [52], keypoints [27], bounding primitives [39], or part structures [47,13,31,50]. While these works present a large space of options in implementing deformations in 3D for various purposes, none of them utilizes *natural language* to guide the desired edit – which is the focus of this work.

Language-Guided Manipulation and Editing. Recently, the big success of CLIP [36] 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 [37] introduced the idea of creating images from texts using a pretrained CLIP model, subsequent work including StyleCLIP [33], StyleGAN-NADA [11], and Paint by Word [5] extended the idea to edit a given image based on linguistic instructions and guidance. For 3D, Text2Shape [8] and the work by Ma *et al.* [29] are the pioneers introducing a framework synthesizing 3D shapes and scenes from texts before CLIP. Then, CLIP-Forge and CLIP-NeRF followed the direction while leveraging CLIP and linking 3D to 2D with either joint embedding or neural rendering. These methods, however, focused mainly on *generating* 3D shapes. For *editing*, Part2Word [40], PartGlot [24], and TGNN [21] made a first step in learning the relations between words and parts in a shape [40,24]; or words and objects in a scene [21]. However, the part or object localization was not exploited to manipulate shapes. To our knowledge, Text2Mesh [30] is an existing work demonstrating language-guided 3D shape manipulation, but its editing is limited to adding color and geometry texture. Our work introduces a framework for editing the *shape* and the *topology* of a 3D object in a fine-grained manner, based on linguistic descriptions.

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 data remain sparse. Some notable examples include ShapeGlot [4], SNARE [41], and PartIt [19]. (See a quick comparison between these and our ShapeTalk in the inset Table). With 527K utterances and a collection of $\sim$ 37k shapes, our ShapeTalk provides an order of magnitude more utterances than the runner-up (ShapeGlot [4], 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 [4], SNARE [41], and PartIt [19] do not provide a complete enumeration of most differences each distractor-target pair has, whereas our dataset *does.* In conjunction with the fact that language naturally under-specifies the full set of required changes, ShapeTalk is better equipped to handle the challenging task of language-assisted shape editing. Table 4 shows the benefit of having more utterances for the same distractor-target pair. While there exist more recent datasets connecting language and 3D such as ReferIt3D [1], ScanRefer [9] and Rel3D [14], their focus is on 3D scenes instead of objects.

Dataset	# utter.	# 3D models	Multi-Categ.	Multi-Utter.
ShapeGlot [4]	79k	7k	No	No
SNARE [41]	50k	8k	Yes	No
PartIt [19]	10k	10k	Yes	No
ShapeTalk (Ours)	528k	37k	Yes	Yes

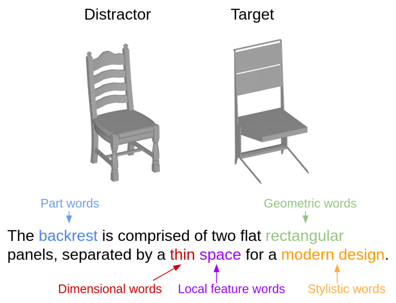
3 The ShapeTalk Data Set

We will 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 the notion of shape-wise similarity by L2-distance within the latent space of a pretrained pointcloud autoencoder [3]. Hard contexts are chosen out of pairs of objects with the highest similarity. Easy contexts are chosen out of pairs of objects with average similarity. ShapeTalk shapes are aggregated from ShapeNet [7], ModelNet [48] and PartNet [32]. After removing duplicates and models of poor quality, ShapeTalk provides discriminative utterances for a total of **35,737** shapes, across **29** object classes. Overall, ShapeTalk contains **72,491** distinct contexts ordered tuples (51.9% of which are Hard), and a total of **528,701** utterances (averaging **7.29** utterances per distinct context).

Creating linguistic descriptions capturing *all exact* differences between two shapes is a very difficult 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. Fig.2). Thus, instead of demanding our annotators to describe *all* the differences between shapes, we ask them to *enumerate* discriminative differences, up to a maximum number of differences, in decreasing order of "obviousness", whether visual or linguistic. Our **2,154** distinct annotators are instructed to provide descriptions that differentiate the two shapes within the context. They do so class-by-class, not only to lessen their cognitive burden, but also to allow them to transfer experience of annotating past examples within the same object class. 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), but do not require them to use these names in their annotations. The resultant annotations form ShapeTalk (see Figure 1.). All objects of a class have been considered as a target in some context within the dataset. All contexts have been described by *at least* two annotators. The sole exception is the chair class, where only one annotator saw each context. We augment the language for this class with 109,718 Chairs In Context utterances from the ShapeGlot dataset.

ShapeTalk utterances are highly diverse. To shed light on the types of language used in ShapeTalk, we manually 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 1 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 (81%), with descriptions often specifying dimensional (36%) and geometric characteristics (25 %), whereas stylistic information is rarer (6%). However, the distribution depends also on the category (See Supp.). 59% of utterances that reference parts and 53% of utterances about local features provide details about style, dimen-


Distractor
Target
Part words
Geometric words
The backrest is comprised of two flat rectangular panels, separated by a thin space for a modern design.
Dimensional words
Local feature words
Stylistic words

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

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

class	part	local	holi-stic	styli-stic	dimen-sional	geom-etric
all	0.81	0.05	0.18	0.06	0.36	0.25
all easy	0.78	0.04	0.20	0.07	0.30	0.25
all hard	0.82	0.06	0.16	0.05	0.40	0.24

Table 1: 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.

Table 2: Proportion of utterances containing joint information about a certain level of visual granularity (holistic, parts, local) and different characteristics (stylistic, dimensional, geometric).

	Stylistic	**Dimensional**	**Geometric**
parts	0.040	0.270	0.204
local feat.	0.004	0.009	0.016
holistic	0.017	0.083	0.039

Table 3: Proportion of utterances containing information about style, dimensions, or geometric shape at the three levels of visual granularity, for Easy, Hard and all contexts.

	All	**Easy**	**Hard**
parts	0.59	0.55	0.63
local feat.	0.53	0.50	0.55
holistic	0.72	0.69	0.75

sionality, geometric shape. A higher percentage (72%) of holistic descriptions have similar qualifications.

What kind of characteristics (geometric, stylistic or dimensional) are used most for which level of visual granularities? Table 2 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.

Language becomes more fine-grained in Hard contexts than for Easy contexts. In the hard context, utterances are 4% more likely to reference parts, 2% more likely to reference local features, and 10% more likely to describe 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 in comparison to Easy ones. As such, the 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 Supp.

4 Desiderata for Language-Assisted Visual Edits & Evaluation Metrics

In this section, we discuss criteria for evaluating the quality of any general approach targeting language-assisted visual edits. We begin by making a few salient observations regarding the nature of the ChangeIt3D task, that will guide us in the introduction of relevant metrics and in building our framework in Section 5. We note that the introduced metrics and algorithms below are flexible and can be adapted to operate with any standard 3D representation.

There are delicate ambiguities when trying to assess how well a proposed shape change addresses a discriminative utterance. For example, consider the expression "has thinner legs" interpreted as an intended geometric change for a given chair. There are two main obstacles in deciphering what exactly an appropriate change in this situation should look like. First, natural language for continuous changes is often under-specific; e.g., exactly how much thinner should the legs become? Even for expressions involving discrete elements, e.g., "has armrests", there is an infinite pool of geometries and placements that can semantically satisfy (albeit to different degrees) the meaning of such an instruction. The following two metrics (LAB and CP, see below) aim at characterizing the extent to which the requested visual change has been achieved while constraining the output and its change into a feasible set – i.e., a chair with thinner legs should still look like a chair. The second cautionary point in our quest on defining success of an output change concerns the fact that oftentimes the intent of the linguistic instruction is local. In other words, only a subset of the input geometry is being referred to, and thus is expected to change. Any ChangeIt3D method should avoid unnecessarily modifying the back of chair when prompted with "has thinner legs". This is a very subtle issue as there can of course be semantic constraints that relate part geometries. Our final metric, l-GD, aims at assessing this quality, by inspecting the geometric stability of the input/output pairs on such non-referred semantic parts.

Specifically, with the above observations in mind, we propose the following metrics:

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 association 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 a good (oracle) neural listener, asking it if it finds the changed item more compatible with the language or 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 possibilities 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 the 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 standard pairwise shape difference metric, such as the Chamfer distance [2] or Hausdorff distance.

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 above stability metric. For example, the inset figure 3 shows a shape modification from the linguistic instruction "has thinner legs". In such a case, it is acceptable to permit the output to significantly (arbitrarily) differ from the input on those 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 above.

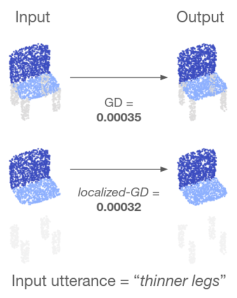


Fig. 3: By removing the referred part (legs), l-GD improves the fidelity of shape similarity focusing on the relevant.

4. **Class Preservation (CP)** 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, CP 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 [38])[5].

Discussion. Clearly the above metrics are imperfect due to the ambiguity and lack of specificity inherent in natural language[6]. There can also be conflicts between the metrics, say between LAB and CP, as faithfully obeying language

[5] 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.

[6] While ambiguity and under-specificity can be seen as drawbacks of a language-based approach, they are also strengths, as they make it possible for everyone to use the proposed tools.

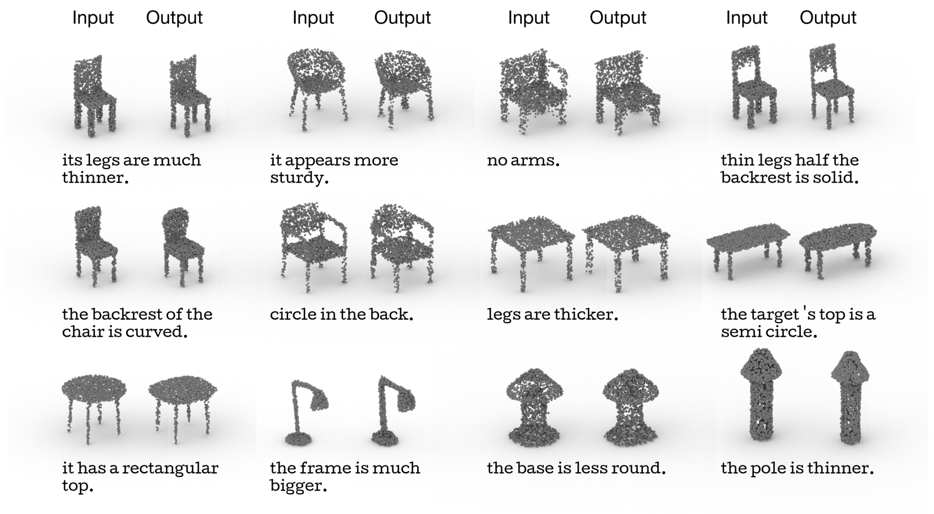

Input
Output
Input
Output
Input
Output
Input
Output
its legs are much thinner.
it appears more sturdy.
no arms.
thin legs half the backrest is solid.
the backrest of the chair is curved.
circle in the back.
legs are thicker.
the target 's top is a semi circle.
it has a rectangular top.
the frame is much bigger.
the base is less round.
the pole is thinner.

Fig. 4: **Qualitative edits produced by ChangeIt3DNet.** The results are based on an SGF backbone AE, decoding pointclouds with 2048 per shape. The achieved edits are oftentimes local, e.g., *thinner legs*, fine-grained, as in *circle in the back*, or entail high-level and complex shape understanding, e.g. *it appears more sturdy*. Remarkably, these edits are derived by ChangeIt3DNet which does **not** utilize any form an explicit geometric local prior of shapes (part-like, or otherwise); but instead learns solely from the implicit bias of training with referential language.

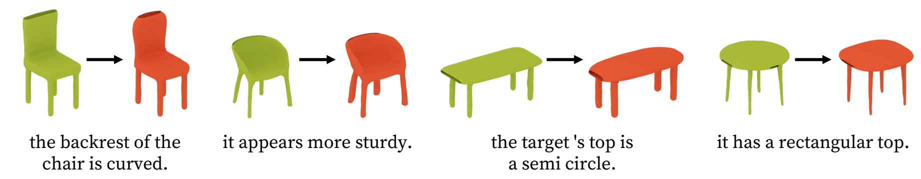

the backrest of the chair is curved.
it appears more sturdy.
the target 's top is a semi circle.
it has a rectangular top.

Fig. 5: **Mesh outputs.** Green is input; red is output. The shown meshes are derived by estimating the magnitude of the predicted gradient from our pretrained SGF AE and applying marching cubes [28] to the resulting unsigned distance field. The corresponding pointcloud predictions from SGF are included in the above Figure 4.

instructions might take take us out of the class manifold for the object (*e.g.* removing the seat from a chair). In general, we would like to have algorithms that sit on some Pareto-optimal boundary regarding the metrics (so it is impossible to improve one without decreasing another). This also suggests multiple directions for future work, including (a) generating a distribution of possible modifications to show the user different options, or (b) allowing a natural language dialog with the user to iterate over the modifications — topics beyond the scope of this paper.

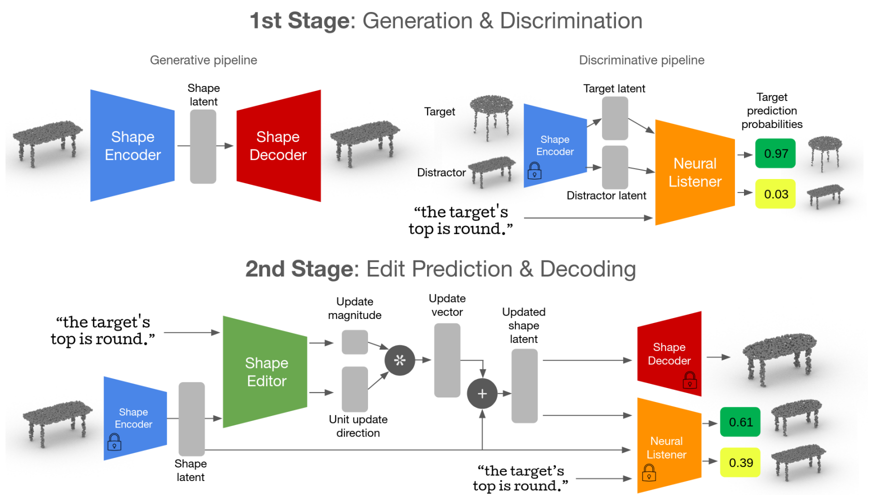


Fig. 6: **Overview of *ChangeIt3DNet*, our modular framework for *ChangeIt3D* task.** In Stage 1, we pretrain a shape autoencoder for shapes (using traditional reconstruction losses), freeze the encoder and use the encoded latents of the target and distractor to pretrain a neural listener (using classification 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.

5 Method

Due to the lack of prior work directly relevant to our task (please also see Section 2), there is a wide range of options for potential *baseline* methods. Here, we focus on one general approach and leave other possibilities for future exploration.

Figure 6 shows a high-level overview of our framework named **ChangeIt3DNet**. 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 [4,24] 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 [33].

To keep things modular, we propose to learn 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 directions in the latent space of G through predicting an update vector by regressing in a decoupled manner, its magnitude and direction. As Table 6 shows, this decoupling is empirically important. This update vector is then applied onto the source shape latent representation *additively*, promoting a direct interaction between the source representation and the edit vector. 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. The use of the original shape for L's comparative context (here termed "self-contrast") is important — we find that the alternative approach of replacing it with the ground-truth target shape has a negative influence, oftentimes leading to violation of similarity preservation in order to achieve high language compatibility (see Table 6). Figure 6 shows the *self-contrast* setup. Details on the specific implementation of L, G and E can be found in Supp.

We also experiment with an alternative variant where instead of aiming to satisfy a pretrained neural listener, this variant is being optimized to *reconstruct* the ground truth target shape, which we call refer to as the **Monolithic Model**. 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 (we experimented with both a transformer-based [44] and LSTM-based [18] utterance encoding), and an MLP-based decoder is used to output a predicted target reconstruction. While this benefits from being trained end-to-end with a well-studied reconstruction loss, such as the Chamfer loss [3], 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 the *full* collection of utterances for each context, concatenated in the annotator-specified order.

Finally, we also implement a shape retrieval model (which we call **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 verify the quality of E and the encoder, decoupled from the quality of the decoder (which, as [46] shows, can bottleneck the output quality). This naturally also gives rise to a useful search-oriented application conditioned on a source shape and a language utterance. Refer to Supp. for more details and diagrams of Monolithic Model and Neighbor Search.

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 and the 3D generative networks are trained, validated and tested on these splits.

6.1 Neural Comprehension of Referential Language for 3D Shapes

Table 4: **Neural comprehension performance**. *Backbone* indicates the underlying model. *Overall* reports average prediction accuracy for the entire test set across all classes of ShapeTalk. *Subpopulations* show average accuracy among the specified parts of the data, i.e. *First*: on utterances indicated as most salient by an annotator (uttered first), *Last*: least salient utterance (uttered fifth), *Easy*: accuracy involving least visually similar pairs, *Hard*: accuracy with pairs having the highest visual similarity. *Concat* reports the accuracy when we concatenate, and train/test the same architectures with all the utterances given by an annotator in a context.

Backbone		*Subpopulations*				
	Overall	**First**	**Last**	**Easy**	**Hard**	**Concat**
ImNet	65.2	67.0	61.3	68.8	61.9	77.8
PC-AE	68.7	70.2	**64.8**	72.4	65.3	81.2
SGF	**69.6**	**72.0**	64.1	**74.1**	65.4	**81.3**

For our neural listeners we use the context-free architecture introduced in ShapeGlot [4] 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] (this improved its training stability and final attained performance). The exact hyper-parameters, regularization and model selection are described in detail in the Supp. With the listening architecture being fixed, we ablate its performance when operating with latent shape representations derived by three widely adopted generative pipelines (AEs): IM-Net [10], ShapeGF (SGF) [6], and PC-AE [3]. Importantly, these AEs operate with different underlying 3D shape representations (e.g, input pointclouds vs. implicits) showcasing the flexibility of our approach.

Specifically, Table 4 shows the percentage accuracies for the neural-listening comprehension task with each of the pretrained AE backbones. On average, the accuracy for the SGF backbone is the highest, and close to that of the PC-AE, 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 **11.7%** for SGF) when operating with a concatenation of all context utterances. In addition to the above table, we **remark** that when using a publicly available pretrained CLIP model [36] without fine-tuning it on ShapeTalk, its accuracy was **52.6%**, 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.

6.2 Editing Experiments

Given the performance advantage of ShapeGF in the listening-comprehension experiments (Table 4) 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. Moreover for our editing experiments we will focus on three classes of ShapeTalk: **chairs**, **tables** and **lamps**. These classes have the advantage that are relatively large: each has more than two thousands models, enabling the training of generative models of sufficient quality; and come with parts annotations, thus enabling the deployment of all our evaluation metrics. Finally, since our CP and l-GD metrics require an oracle shape classifier and a part segmentation model, respectively, we use the same splits used for ChangIt3DNet to train: 1) **for CP:** a PointNet++ [35] object classifier predicting the 29 classes of ShapeTalk, and operating with surface-extracted pointclouds of 2048 points per shape. This model achieves an average prediction accuracy of 89.4%. 2) **for l-GD:** a PointNet-based [34] architecture adapted for the prediction of semantic parts of 3D pointclouds. We use pointclouds with 2500 points per shape, annotated with part labels extracted from the ShapeNetParts [51] – the average number of parts/mIOU in our categories is 3.75/86.7%.

Table 5 shows an evaluation of the Monolithic and Search-based baselines against ChangeIt3DNet. Our model's ability to preserve details from the source shape (minimal GD score) while maximizing the LAB score demonstrates a general ability to change shapes as to become consistent 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 produces high LAB score, demonstrating that the shape editor module learns to produce updated latents that capture the described characteristics. However, its GD metric tells us that this in general produces large variations in the shape as a whole, and identity preservation of the input shape is largely violated. This further reveals the sparsity of the shape space with respect to linguistically articulable shape differences, and highlights the importance of being able to *edit shapes* instead of *retrieving* them from a database. Figure 4 shows qualitative examples of decoded shape-edits with our ShapeGF-based model, using pointclouds across our three main shape categories. We also demonstrate meshes extracted from SGF by estimating unsigned distance fields from its predicted gradients (see[6] for details) for a few of these pointclouds (Figure 5). More examples including *failure cases* are in the Supp.

Table 6 shows the evaluation metrics on variants of the ChangeIt3DNet 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

Table 5: **Quantitative comparisons for three baselines solving *ChangeIt3D*.** A *'monolithic'* approach that directly 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 (*ChangeIt3DNet*).

Baselines	**GD** (↓)	**LAB** (↑)	**CP** (↓)	***l*-GD** (↓)
Monolithic	0.563	51.8	1.4	1.50
Neighbor-Search	1.388	76.8	**0.2**	4.09
ChangeIt3DNet (main)	**0.356**	**77.7**	3.8	**0.818**

Table 6: **Ablations for *ChangeIt3DNet*.** 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 shape classes (chair, table, lamp). GD and *l*-GD are based on Chamfer distance, scaled by 10e-4; LAB and CP are percentages.

Decoupled	*Self Contrast*	**GD** (↓)	**LAB** (↑)	**CP**(↓)	***l*-GD** (↓)
✗	✗	0.593	82.2	5.3	1.335
	✓	0.366	76.9	3.9	0.855
✓	✗	0.538	**82.6**	5.0	1.219
	✓	**0.356**	77.7	**3.8**	**0.818**

a decoupled shape editor is better regardless of whether the neural-listener uses self-contrast, and that self-contrast improves identity preservation (GD), CP and localized GD, but sacrifices some utterance-compatibility (LAB). This trade-off between identity preservation and utterance-compatibility is further explore in more ablation experiments presented in the Supp.

7 Conclusion

In this paper we introduce a new task, that of language-driven edits and deformations of 3D models, called *ChangeIt3D*. Towards this objective we release a new dataset, called *ShapeTalk*, containing over 500K contrasting language utterances differentiating two shapes – an order of magnitude larger that any other comparable dataset. We illustrate the potential of this data by exhibiting a general framework for adding a neural listener to a variety of backbone 3D representations and propose evaluation metrics for the *ChangeIt3D* task.

We hope that future works will build on this foundation and make language-guided 3D shape editing widely accessible and useful.

References

1. Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., Guibas, L.J.: ReferIt3D: Neural listeners for fine-grained 3d object identification in real-world scenes. In: European Conference on Computer Vision (ECCV) (2020)
2. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.J.: Representation learning and adversarial generation of 3D point clouds. Workshop on Implicit Models, International Conference of Machine Learning (ICML) (2017)
3. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.J.: Learning representations and generative models for 3D point clouds. In: International Conference on Machine Learning (ICML) (2018)
4. Achlioptas, P., Fan, J., Hawkins, R.X., Goodman, N.D., Guibas, L.J.: ShapeGlot: Learning language for shape differentiation. In: International Conference on Computer Vision (ICCV) (2019)
5. Bau, D., Andonian, A., Cui, A., Park, Y., Jahanian, A., Oliva, A., Torralba, A.: Paint by word. Computing Research Repository (CoRR) abs/2103.10951 (2021)
6. Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S.J., Snavely, N., Hariharan, B.: Learning gradient fields for shape generation. In: European Conference on Computer Vision (ECCV) (2020)
7. Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An information-rich 3D model repository. Computing Research Repository (CoRR) abs/1512.03012 (2015)
8. Chen, K., Choy, C.B., Savva, M., Chang, A.X., Funkhouser, T., Savarese, S.: Text2shape: Generating shapes from natural language by learning joint embeddings. Computing Research Repository (CoRR) abs/1803.08495 (2018)
9. Chen, Z.D., Chang, A.X., Nießner, M.: ScanRefer: 3D object localization in RGB-D scans using natural language. Computing Research Repository (CoRR) abs/1912.08830 (2019)
10. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
11. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: StyleGAN-NADA: CLIP-guided domain adaptation of image generators. Computing Research Repository (CoRR) abs/2108.00946 (2021)
12. Gao, L., Yang, J., Qiao, Y.L., Lai, Y.K., Rosin, P.L., Xu, W., Xia, S.: Automatic unpaired shape deformation transfer. In: ACM SIGGRAPH Asia (2018)
13. Gao, L., Yang, J., Wu, T., Yuan, Y.J., Fu, H., Lai, Y.K., Zhang, H.: Sdm-net: Deep generative network for structured deformable mesh. In: ACM SIGGRAPH Asia (2019)
14. Goyal, A., Yang, K., Yang, D., Deng, J.: Rel3D: A minimally contrastive benchmark for grounding spatial relations in 3D. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
15. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: 3D-CODED: 3D correspondences by deep deformation. In: European Conference on Computer Vision (ECCV) (2018)
16. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: Deep self-supervised cycle-consistent deformation for few-shot shape segmentation. In: Eurographics Symposium on Geometry Processing (SGP) (2019)
17. Hanocka, R., Fish, N., Wang, Z., Giryes, R., Fleishman, S., Cohen-Or, D.: ALIGNet: Partial-Shape agnostic alignment via unsupervised learning. ACM Transactions on Graphics (TOG) (2018)

18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation (1997)
19. Hong, Y., Li, Q., Zhu, S.C., Huang, S.: VLGrammar: Grounded grammar induction of vision and language. International Conference on Computer Vision (ICCV) (2021)
20. Huang, J., Jiang, C.M., Leng, B., Wang, B., Guibas, L.: MeshODE: A robust and scalable framework for mesh deformation. Computing Research Repository (CoRR) abs/2005.11617 (2020)
21. Huang, P.H., Lee, H.H., Chen, H.T., Liu, T.L.: Text-guided graph neural networks for referring 3d instance segmentation. In: AAAI Conference on Artificial Intelligence (2021)
22. Jack, D., Pontes, J.K., Sridharan, S., Fookes, C., Shirazi, S., Maire, F., Eriksson, A.: Learning free-Form deformations for 3D object reconstruction. In: Asian Conference on Computer Vision (2018)
23. Jiang, C., Huang, J., Tagliasacchi, A., Guibas, L., et al.: Shapeflow: Learnable deformations among 3D shapes. Advances in Neural Information Processing Systems (NeurIPS) (2020)
24. Koo, J., Huang, I., Achlioptas, P., Guibas, L.J., Sung, M.: PartGlot: Learning shape part segmentation from language reference games. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
25. Kurenkov, A., Ji, J., Garg, A., Mehta, V., Gwak, J., Choy, C.B., Savarese, S.: DeformNet: Free-Form deformation network for 3d shape reconstruction from a single image. In: Winter Conference on Applications of Computer Vision (WACV) (2018)
26. Litany, O., Bronstein, A., Bronstein, M., Makadia, A.: Deformable shape completion with graph convolutional autoencoders. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
27. Liu, M., Sung, M., Mech, R., Su, H.: DeepMetaHandles: Learning deformation meta-handles of 3d meshes with biharmonic coordinates. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
28. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: ACM SIGGRAPH (1987)
29. Ma, R., Patil, A.G., Fisher, M., Li, M., Pirk, S., Hua, B.S., Yeung, S.K., Tong, X., Guibas, L., Zhang, H.: Language-driven synthesis of 3D scenes from scene databases. In: ACM SIGGRAPH Asia (2018)
30. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. Computing Research Repository (CoRR) abs/2112.03221 (2021)
31. Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., Guibas, L.: Structedit: Learning structural shape variations. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
32. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: 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)
33. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: StyleCLIP: Text-driven manipulation of stylegan imagery. In: International Conference on Computer Vision (ICCV) (2021)
34. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

35. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. Computing Research Repository (CoRR) abs/2103.00020 (2021)
37. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. International Conference on Machine Learning (ICML) (2021)
38. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems (NeurIPS) (2016)
39. Sung, M., Jiang, Z., Achlioptas, P., Mitra, N.J., Guibas, L.J.: DeformSyncNet: Deformation transfer via synchronized shape deformation spaces. In: ACM SIGGRAPH Asia (2020)
40. Tang, C., Yang, X., Wu, B., Han, Z., Chang, Y.: Part2Word: Learning joint embedding of point clouds and text by matching parts to words. Computing Research Repository (CoRR) abs/2107.01872 (2021)
41. Thomason, J., Shridhar, M., Bisk, Y., Paxton, C., Zettlemoyer, L.: Language grounding with 3d objects. Computing Research Repository (CoRR) abs/2107.12514 (2021)
42. Uy, M.A., Huang, J., Sung, M., Birdal, T., Guibas, L.: Deformation-aware 3d model embedding and retrival. In: European Conference on Computer Vision (ECCV) (2020)
43. Uy, M.A., Kim, V.G., Sung, M., Aigerman, N., Chaudhuri, S., Guibas, L.: Joint learning of 3d shape retrieval and deformation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
45. Wang, W., Ceylan, D., Mech, R., Neumann, U.: 3DN: 3d deformation network. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
46. Wojna, Z., Ferrari, V., Guadarrama, S., Silberman, N., Chen, L.C., Fathi, A., Uijlings, J.: The devil is in the decoder: Classification, regression and gans. International Journal of Computer Vision 127(11), 1694–1706 (2019)
47. Wu, Z., Wang, X., Lin, D., Lischinski, D., Cohen-Or, D., Huang, H.: Sagnet: structure-aware generative network for 3d-shape modeling. In: ACM SIGGRAPH (2019)
48. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: A deep representation for volumetric shapes. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
49. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: PointFlow: 3d point cloud generation with continuous normalizing flows. In: International Conference on Computer Vision (ICCV) (2020)
50. Yang, J., Mo, K., Lai, Y.K., Guibas, L.J., Gao, L.: DSM-Net: Disentangled structured mesh net for controllable generation of fine geometry. ACM Transactions on Graphics (TOG) (2021)
51. Yi, L., Kim, V.G., Ceylan, D., Shen, I., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L.J.: A scalable active framework for region annotation in 3d shape collections. ACM Transactions on Graphics (TOG) (2016)

52. Yifan, W., Aigerman, N., Kim, V.G., Chaudhuri, S., Sorkine-Hornung, O.: Neural cages for detail-preserving 3d deformations. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
53. Yumer, E., Mitra, N.J.: Learning semantic deformation flows with 3d convolutional networks. In: European Conference on Computer Vision (ECCV) (2016)