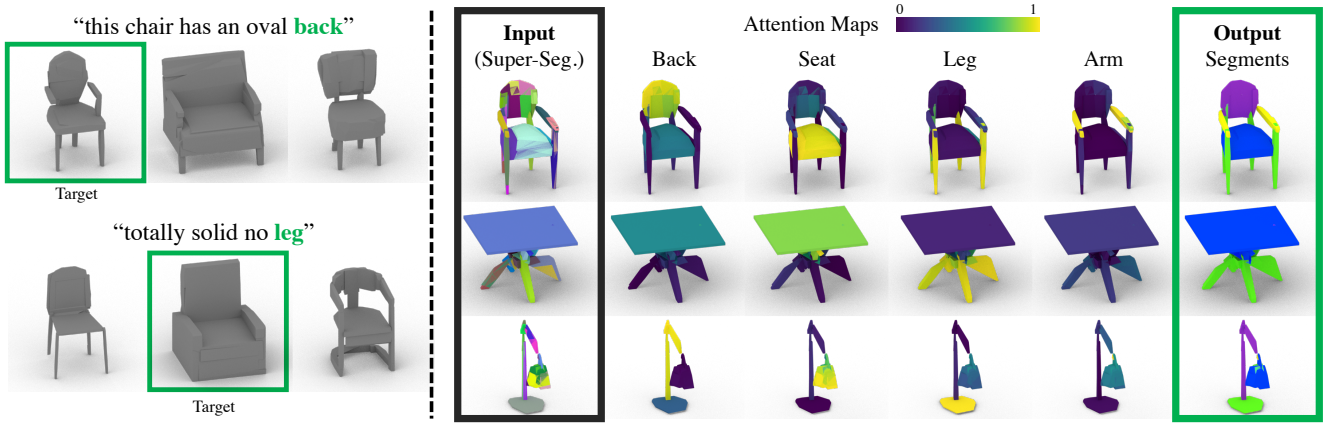
PartGlot: Learning Shape Part Segmentation from Language Reference Games

Anonymous CVPR submission

Paper ID 3830



(a) Referential Language for Shapes

(b) Language-guided 3D Part Segmentations via Neural Attention

Figure 1. **Overview.** On the left panel, we present examples of *referential language* distinguishing the **shape** of a "target" geometry (enclosed inside a green box) from two "distractor" objects. Using such language our proposed task is to estimate directly in 3D-space semantic part segmentations of objects. On the right panel, we present the key ingredients of the neural architecture to facilitate this goal: given referential language and unsupervised 3D super-segments of shapes, we learn a set of attention maps that corresponds to semantic shape parts (when properly regularized), discovered **solely** by solving the language-reference problem of identifying the target shape. Tapping on the zero-shot learning capacity of natural language learners, and the shared part-composition of common objects, we find examples of **zero-shot** segmentations on a table and lamp objects, extracted from learners and language concerning *only* chair-based comparisons.

Abstract

We introduce PartGlot, a neural framework and associated architectures for learning semantic part segmentation of 3D shape geometry, based solely on part referential language. We exploit the fact that linguistic descriptions of a shape can provide priors on the shape's parts – as natural language has evolved to reflect human perception of the compositional structure of objects, essential to their recognition and use. For training, we use the paired geometry / language data collected in the ShapeGlot work [3] for their reference game, where a speaker creates an utterance to differentiate a target shape from two distractors and the listener has to find the target based on this utterance. Our network is designed to solve this target discrimination problem, carefully incorporating a Transformer-based attention module so that the output attention can precisely highlight the semantic part or parts described in the language. Furthermore, the network operates without any direct supervision on the 3D geometry itself. Surprisingly, we further demonstrate that the learned part information is generalizable to shape classes unseen during training. Our approach opens the possibility of learning 3D shape parts from language alone, without the need for large-scale part geometry annotations, thus facilitating annotation acquisition.

1. Introduction

Object perception is often based on structural abstractions — the decomposition of an object into its parts and their inter-relationships [10, 11, 16]. Natural language reflects this aspect of human perception of 3D shapes – when a human is asked to describe an object, the description usually involves words *naming* parts and expressions about part attributes and their relationships. This implies that, conversely, language descriptions of an object can provide priors on the compositional structure of the object geometry,

including the identity of its components or parts. In this paper, we study the interplay between these two very different modalities, geometry and language, and how it can guide learning shape structure and parts.

ShapeGlot [3] explored the interplay between natural language and object geometry for the task of differentiating objects. It proposed a way to design a crowd-sourcing task to elicit more part-related referential language (utterances) about objects from users, based on a *reference game*. Specifically, one user (the speaker) is shown three related objects (a "target" shape and two "distractor" shapes) and is asked to describe how the target is *different* from the distractors. A second user (the listener) is then asked to select the one described by the first user. An interesting aspect of this work is that even though in training the (referential) neural networks are only given holistic shape representations with no part information whatsoever, they learn to depend heavily on part-related words and the corresponding visual parts of objects.

Motivated by this initial observation and using the same data, our work investigates how well a neural network can connect part names in the utterances to specific regions in the geometry of the 3D shapes. We show the remarkable fact that **geometric object part structure can emerge from language alone, without any direct geometric supervision on part segments**, highlighting the deep ties between language and geometry. In other words, we can discover semantic part segments on the geometry by exploiting solely referential language data. Even the language data we use is *pragmatic*, not guided by any comprehensive partonomy as done by previous work [17], but merely focusing on describing shape differences.

Our framework is based on a variant of the neural listener pipeline in ShapeGlot, taking a language utterance plus three 3D shapes in point cloud format and predicting the probability of how likely each of the shapes is to be the target described by the utterance. For this learning task, we explore the application of a Transformer-based attention module [33] to learn the region corresponding to each part described in the utterance as attention focus. Simply plugging in an attention module, however, does not produce any meaningful regions aligned with the semantic parts. Hence, we make several important changes that lead the network to learn meaningful part segmentation masks as a byproduct of learning to identify the target shape. Our experimental results demonstrate that the essential architectural components in our network significantly improve the performance of part segmentation. Also, in the case when the full set of part names are given at training time, we show that this additional information can be leveraged to better detect and segment parts. Furthermore, we show that our network can generalize to out-of-distribution categories of shapes – specifically, with training done on *Chairs*, good semantic masks can be extracted out of instances of *Tables* and *Lamps*.

Beyond studying the capability of neural networks to jointly understand language and shape, this work also suggests a new potential way to collect data for object part segmentation. Object or scene segmentation is a fundamental problem in many vision tasks, but the advance of learning-based segmentation techniques is gated by the availability of large-scale human segmentation annotations of 2D images or 3D models. Particularly for 3D, collecting manual annotations on 3D objects requires a huge amount of human effort and cost. In contrast to this, uttering a language description is a much more natural way for people to provide information about object structure and geometry. We hope to see a lot more work on how 3D segmentation can be improved using the language description of objects, without direct geometry supervision.

2. Related Work

Language and Shape Works that explore the intersection between language and geometry have taken many forms, from resolving language references [2, 3, 31], to generating language descriptions of a shape [3, 18], to generating a shape given a language description [21, 30]. Most relevant to our work are the ones that attempt the language reference game, where the task is to select based on a language description a target shape out of a set of potential candidates either in a collection of individual 3D shapes [3, 31] or within a scene [2, 19, 29, 35, 38, 40]. While most of these works treat the reference game as a classification problem on the set of candidates, [19] outputs a segmentation mask over the scene. However, unlike our method, their work 1) applies to 3D scenes instead of individual shapes, and 2) requires full supervision for the segmentation task. To the best of our knowledge, our work is the first to derive part-level segmentation masks from spatial attention as a byproduct of learning to play the language reference game.

Transformers Not only have Transformers demonstrated superior performance in several tasks [4, 9, 15, 20, 33, 36], but they also are characterized by interpretability of the attention map and can discover meaningful correspondences between different modalities [36]. In addition to being applied in the 2D visual domain [4, 9, 15], transformers have also been used in the 3D spatial domain for a variety of tasks, operating most commonly on point clouds. A variety of attention mechanisms has been introduced [27]. For instance, [37] adapts the transformer architecture for point cloud completion. Works like [12, 23, 39] have shown superior performance on the semantic segmentation task by including modules that employ self-attention over point clouds. However, in all the above cases, the segmentation

masks were developed with heavy supervision, and not extracted using the attention over the spatial domain. They furthermore do not attempt to leverage information from other modalities. Here, instead of using self-attention over only the spatial domain, we use cross-attention between multiple modalities — a byproduct of learning language references — for the segmentation task.

Self-Supervised or Weakly-Supervised Segmentation [41] proposed a weakly-supervised shape co-segmentation method. Two key elements are the part prior network and low-rank loss. It first trains the part prior network to learn a part prior by denoising unlabeled segmented parts from random noise. From this pre-trained part prior network, the co-segmentation network is optimized to output consistent segmentations via low-rank loss. Low-rank loss regularizes the network to maximize the similarity of the part feature belonging to the same part by minimizing the rank of the matrix consisting of part features of the same part across all test shapes. [34] also utilizes two key elements introduced in [41] for the fine-grained segmentation without part semantic tags. Those networks are trained in a label-agnostic manner, but still require segmentation information to train the part prior network. Our model does not require any part prior, but learns geometry from language on the fly.

Shape Decomposition Recently, there have been many works [6, 7, 7, 8, 13, 24, 25, 32] for shape decomposition. [14, 25, 32] abstract a complex shape into multiple primitives, cuboids, superquadrics or gaussians, by regressing the parameters of the primitive that fit to the target shape. [6, 8] decompose a shape as a collection of convexes. [7, 24] learn an implicit field to represent the shape. Those works have demonstrated to abstract the shape into multiple primitives, but those primitives lack of semantics. So, they usually assigned the label for each primitive by hand in the test time.

3. Attention-Based Part Segmentation

3.1. Background and Overview

We investigate the capability of a neural network (particularly, an attention module) to learn semantic parts of a 3D object solely from referential language of an object without any explicit supervision of its part segmentations. For this, we deploy a listening comprehension task similar to ShapeGlot's [3] as our basic learning objective. Specifically, given a triplet of shapes and an utterance differentiating one of them, our main task is to learn to identify the referred target shape (Figure 2). For this task, a variety of viable neural network architectures can be designed to assign a probability to each shape indicating its congruence with the underlying utterance. This work is the first to demonstrate that by carefully incorporating an attention module over the 3D spatial domain of the visual stimulus (e.g., attention over unordered sets of 3D point clouds), our network can not only learn to identify the target shape but also discover 3D regions of the parts described in the utterance as a byproduct of solving the reference task.

We adapt the original neural network architecture in ShapeGlot to better facilitate our goal of recognizing and segmenting object parts with language alone. First, we focus on the application of neural listeners operating *solely* on 3D geometric representations – 3D point clouds – and ignore 2D image-based projections used in ShapeGlot. Second, we also explore the effect of partitioning the input point cloud into subgroups, namely *super-segments* (analogous to superpixels [1] in 2D), and cast the semantic part-prediction problem over those larger entities. Crucially, super-segments (groups of points) can be derived with a self-supervised approach (in our experiments, we use the output of BSP-Net [6]); hence their utilization does not undermine our goal of annotation-free 3D part segmentations. Third, we add a Transformer-based [33] attention module taking the utterance or a part name as a query. We also change the architecture of the geometry encoder to make the neural network seek an appropriate local region for attention; more details are described below. We investigate two different setups of the problem: with and without knowledge of the full set of part names during training.

3.2. Part-Name-Agnostic (PN-Agnostic) Learning

We first describe a learning scenario where the set of part names is *not* known during training. The association between part names (words) and regions in the 3D shape in this case has to be learned solely from the connection between utterances (a single or multiple sentences) and the entire 3D shapes.

The network architecture is illustrated in Figure 2. The input utterance $\mathbf{u}$ is encoded into two encoders: *attention* encoder $f_a(\cdot)$ which decides "where to look" and *classification* encoder $f_c(\cdot)$ which determines "whether it is the target shape or not". For both encoders, we use an utterance encoder similar to the one used in the ShapeGlot; the token codes of the words in a sentence are randomly initialized and then processed via an LSTM sequentially with the standard bilinear word attention mechanism [22]. The output of the *attention* encoder $f_a(\mathbf{u})$ becomes the *query* vector in the subsequent Transformer [33], and the output of the *classification* encoder $f_c(\mathbf{u})$ is concatenated with the output of the Transformer (a weighted sum of the super-segment features) and is used to predict the classification probabilities.

For the three input shapes $\{o_1, o_2, o_3\}$, the target and two distractors, each of which is represented a set of super-segments $o = \{\mathbf{s_i}\}$, we extract a key $g_k(\mathbf{s_i})$ and a value $g_v(\mathbf{s_i})$ vector of each super-segment s_i using PointNet [26]. In the following single cross attention layer, the attention

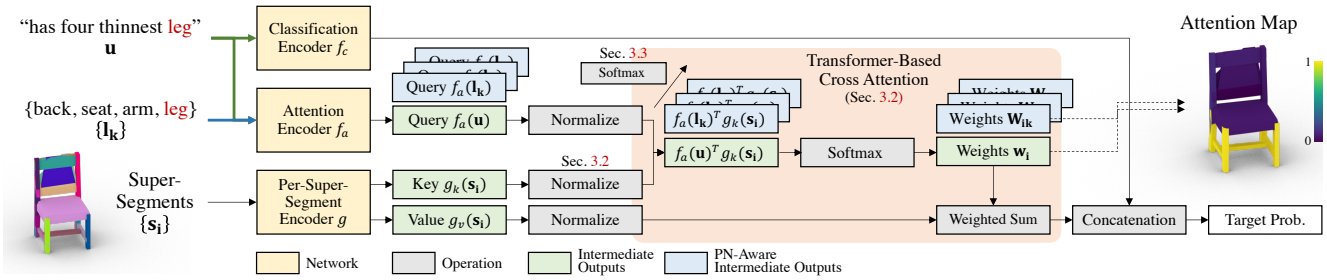


Figure 2. A higher-level overview of our architecture playing a reference game. There are three main encoders: *Classification* Encoder f_c, *Attention* Encoder f_a and Per-Super Segment Encoder g. The cross attention module aggregates Per-Super-Segment features based on Query to output the shape feature. The concatenation of the output of *Classification* Encoder $f_c(\mathbf{u})$ and the shape feature produces the final classification probability. The attention map contains the semantic information corresponding to input language.

from the utterance $\mathbf{u}$ to each super-segment $\mathbf{s_i}$ is calculated by taking a dot product of the query and key — let $\mathbf{x}$ be a vector where $\mathbf{x}_i = f_a(\mathbf{u})^T g_k(\mathbf{s_i})$ — and then normalizing them over super-segments using a softmax:

$$\mathbf{w_i} = \sigma(\mathbf{x})_i = \frac{e^{\mathbf{x}_i}}{\sum_i e^{\mathbf{x}_i}}. \tag{1}$$

The resulting probability distribution over the super-segments $\{\mathbf{w_i}\}$ becomes the attention expected to indicate the part described in the utterance $\mathbf{u}$. The value vectors $\{g_v(\mathbf{s_i})\}$ are then aggregated by taking the probabilities $\{\mathbf{w_i}\}$ as weights in a weighted mean, concatenated with the output of the classification encoder $f_c(\mathbf{u})$, and fed to an MLP to predict the classification score of each object.

A crucial detail in this architecture is to *normalize* the query $f_a(\mathbf{u})$, key $g_k(\mathbf{s_i})$, and value $g_v(\mathbf{s_i})$ vectors to have a unit norm. Although missing this normalization does not affect the accuracy of target shape discrimination, it largely influences the attention and helps align the attention to a semantic part in practice since the weights in the attention can vary according to the different norms of the value vectors $g_v(\mathbf{s_i})$. The effect of the normalization is shown in Section 4.2. Note that, in Equation 1, we also do not divide the dot product of the query and key by the square root of the vector dimension as typically done in Transformer since all the vectors are normalized.

One more important observation is that the method used to process *set* data is critical. Following PointNet [26], many neural networks processing set data use the idea of combining local features with a *global* feature, which is created by aggregating all the local features using a symmetric function such as max-pool. In our pipeline, we find that the concatenation of the global feature to each super-segment feature results in totally meaningless attention since the Transformer does not need to attend to a specific region, as each point can provide global shape information to complete the reference task. Hence, all super-segments are processed independently with the shared network module.

At test time, we obtain the attention of a part by leveraging a similar setup as CLIP [28]; a template language expression is used as the input utterance. In our experiments, we specifically use an expression: "`a chair with {part name}`". Given a set of part names, a segment per part is achieved by taking super-segments whose probability of the part attention is higher than the probabilities of any other attentions.

3.3. Part-Name-Aware (PN-Aware) Learning

In the case when the set of part names $\{\mathbf{l_k}\}$ is predefined at the training time, we leverage this additional supervision to better align the output attentions to the given set of parts. Note that there is still no part segmentation supervision, since only the set of part names is given. In this setup, we also assume that each utterance describes one and only one part in the given set.

From the architecture introduced in Section 3.2, we first change *attention* encoder $f_a(\cdot)$ to take not the input utterance $\mathbf{u}$ but the part name $\mathbf{l}$ described in the utterance as the input. Hence, a single-layer MLP for the part name latent token is used instead of an LSTM. In the test time, we also do not need to use a template expression; the part name can be directly fed to the attention encoder. Also, since now the set of part names is given, we propose to *jointly* normalize the attentions of different part names, which is essential to improving attention-based part segmentation. We specifically collect the dot products of the query and key vectors $f_a(\mathbf{l_k})^T g_k(\mathbf{s_i})$ for all the part names $\{\mathbf{l_k}\}$ and super-segments $\{\mathbf{s_i}\}$. Let $\mathbf{X}$ be a matrix where $\mathbf{X_{ik}} = f_a(\mathbf{l_k})^T g_k(\mathbf{s_i})$. Then, we apply softmax to $\mathbf{X}$ *twice*; along the set of part names first (along k) and then along the super-segments (along i):

$$\mathbf{Y_{ik}} = \sigma(\mathbf{X_{i,:}})_k \tag{2}$$

$$\mathbf{W_{ik}} = \sigma(\mathbf{Y_{:,k}})_i, \tag{3}$$

where $\sigma(\cdot)$ is the softmax, and $\mathbf{W} = \{\mathbf{W_{ik}}\}$ is the final

weights. The first *additional* softmax along with the part names (k) plays the role of making $\mathbf{X_{ik}}$ be *spikier* for each super-segment, enforcing a super-segment to belong to *only one* part name. This can thus avoid overlaps across the attention maps of different part names. We empirically find that still the final attention weights should be normalized over the super-segments to achieve the best performance. We show a comparison across different cases of applying softmax in our ablation study (Section 4.2).

Regularization Based on Cross Entropy To further enforce *partitioning* of the output segments — ensuring that a point is assigned to *one and only one* part name — we introduce a regularization loss based on cross entropy. Given $\mathbf{Y}$ (the output of the first softmax) and for each super-segment $\mathbf{s_i}$, we find the part name $\mathbf{l_k}$ that gives the highest probability $\mathbf{Y_{ik}}$ and compute the cross entropy loss by considering that the part as the ground truth label:

$$\mathcal{L}_{\text{CE}} = \sum_i \sum_k -\mathbb{1}\left(k = \arg\max_{k'}(\mathbf{Y_{ik'}})\right) \log(\mathbf{Y_{ik}}) \quad (4)$$

In addition to the double softmax, the regularization loss makes $\mathbf{Y}$ even spikier and further avoids overlaps across the attention maps of different part names. The ablation study in our experiments analyzes the effect in practice (Section 4.2).

4. Experiments

4.1. Dataset and Evaluation

In our experiments, we use the Chair in Context (***CiC***) dataset introduced in ShapeGlot [3]. ***CiC*** includes sets of triplets of *chairs* from ShapeNet [5] (a target and two distractors) and an utterance for the target chair, created by human speakers playing the grounded reference game.

Utterance Preprocessing We first preprocess the utterances of ***CiC*** by fixing typos, converting plural nouns to singular nouns, and dividing a compound word into single words, e.g., "armrest" to "arm rest". For the PN-Aware setup, we choose the following four part names as the given set: *back*, *seat*, *leg*, and *arm*, which are also chair part segments annotated in ShapeNet [5]. We also only use chair triplets in ***CiC*** where their associated utterance describes only one of these parts. After the preprocessing, the dataset contains 40,660 sets and 4509 unique shapes. We split the sets into train, validation, and test datasets with an 80%-10%-10% ratio. Since the numbers of the utterances describing each part are imbalanced, during training, we sample the utterances with the probabilities inversely proportional to the numbers of each part utterances.

Figure 3. Super-segments generated by a pretrained model of BSP-Net [6]. The colors are randomly assigned to super-segments.

Table 1. Super-segment statistics.

	Min.	Max.	Mean
# Super-Segs	4	47.4	20.6
# Pts in Super-Segs	0	1550.0	90.3

Super-Segment Generation The super-segments of each shape are produced by a pre-trained BSP-Net [6], provided by the authors; see examples in Figure 3. Then, each super-segment is represented with a small set of points, generated by randomly sampling 2048 points over the entire shape and assigning them to super-segments based on proximity — a point is assigned to *one and only one* super-segment whose signed distance to the point is the minimum, and thus the super-segments *partition* the point cloud. See Table 1 for the statistics of the number of super-segments and the number of points in each super-segment. We further sample the points per super-segment so that the maximum number of points becomes 512.

Segmentation Evaluation At test time, we obtain segments of the four parts — *back*, *seat*, *leg*, and *arm* — as the attention. Depending on whether the setup is PN-Aware or PN-agnostic, either the template sentence mentioned in Section 3.2 or the part name itself is fed to the *attention* encoder $f_a(\cdot)$ and used to generate the attention. The super-segments are assigned to the part name with the highest probability in the attention. The segmentation are evaluated based on ground truth part segmentation annotated in ShapeNet [5]. The standard mIoU is used as the evaluation metric of the segmentation. The *average* mIoU indicates taking mean *per instance* and averaging over the shapes.

4.2. Results

The quantitative and qualitative results of our experiments are summarized in Table 2 and Figure 4. We first show two comparisons PN-Agnostic (Section 3.2) vs. PN-Aware (Section 3.3), and super-segments vs. points. We then show the results of the ablation study for each crucial component in our pipeline. We also show the results of analyzing the effect of few-shot learning when assuming that the ground truth part segments are annotated in a few shapes. In the end, we also demonstrate that our framework learns general part information that can be transferred to other shape classes (e.g., Tables and Lamps), and we also

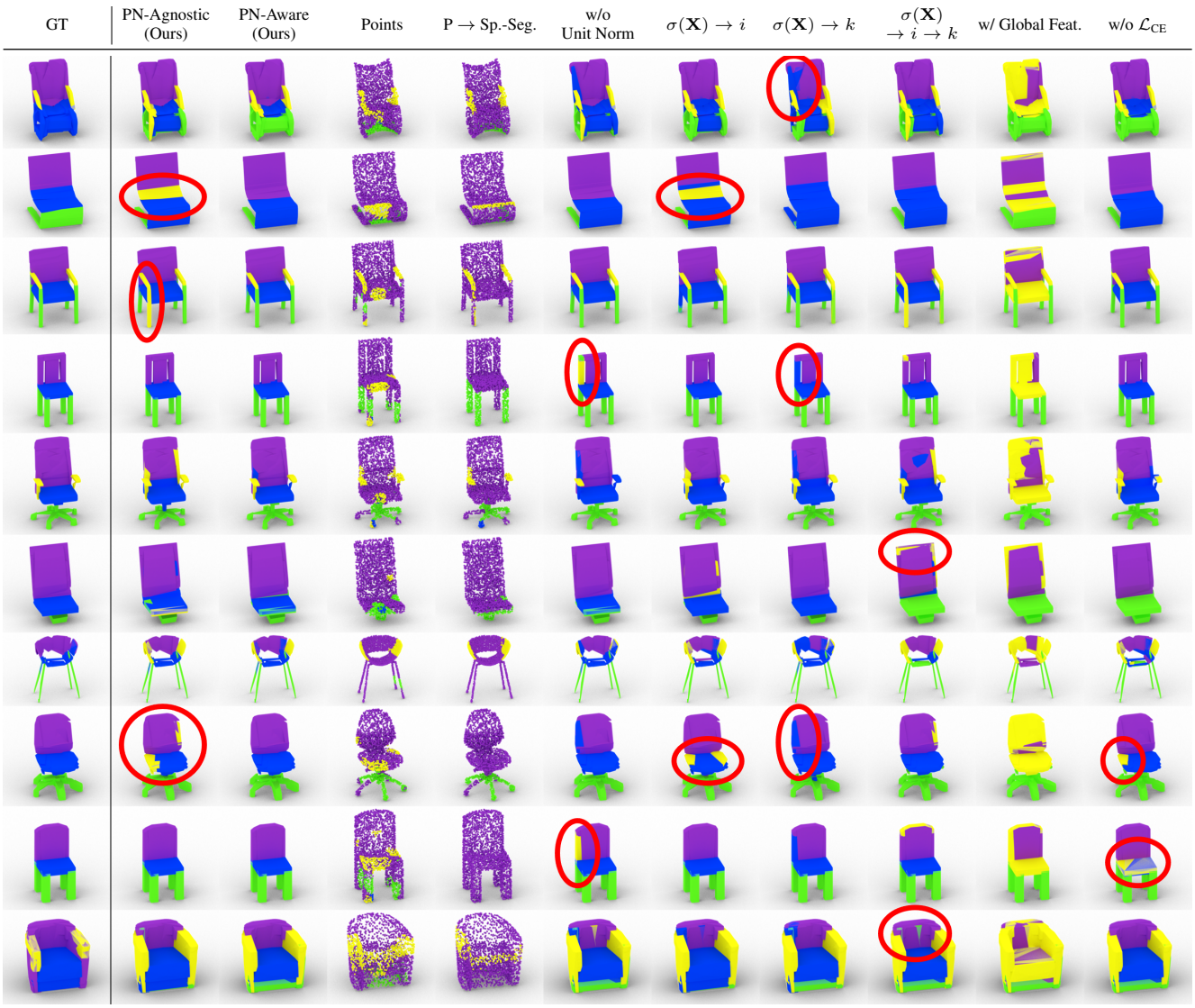


Figure 4. Samples of extracted part segmentation across variations of our method, indicated by the 4 colors. Purple, blue, green and yellow indicate the prediction as *back*, *seat*, *leg*, and *arm*, respectively. The colors assigned to the super-segments in the ground truth column (GT) are computed based on ground truth point cloud part segmentation from ShapeNet [5]. Note that our PN-Agnostic and PN-Aware setups produced the best segmentation masks.

visualize the word attention in the utterance encoding.

PN-Agnostic vs. PN-Aware We first compare the two cases described in Section 3: when leveraging the set of part names in training (PN-Aware) or not (PN-Agnostic). The mIoUs are reported in rows 1 and 2 of Table 2. While PN-Agnostic (row 1) works well in most cases, it particularly shows a low mIoU for *arm* compared to the case of learning with part names (40.6 vs 70.4). The arm is an optional part that may not exist in some shapes, and by definition of mIoU (used in PointNet [26]), it becomes zero when there even exists a *single* super-segment assigned to *arm* while *arm* does not exist. We observe that such failure cases happen when the full set of part names are not leveraged during training (see the examples in the *second* and *eighth* row in Figure 4), although these cases are greatly reduced after the part names are used (PN-Aware) with our essential components in the network. The reasons are further analyzed in the ablation study below. Note that the accuracy of target shape classification is almost the same for both cases. The cross-part mIoUs are reported in the supplementary.

For the rest of the experiments, we show the results for PN-Aware with various setups.

Table 2. Quantitative Results of all experiments: [Id 1, 2] Comparison of two baselines; [Id 3-5]: Comparison of the input granularity; [Id 7-12]: ablation cases; [Id 13-15]: few-shot learning results. For each experiment, the model was selected with the highest classification accuracy on the validation set. **Bold** indicates the highest mIoU *except* for the few-shot learning results.

Id	Method	Segmentation mIoU(%)					Classif.
		Back	Seat	Leg	Arm	Avg.	Acc.(%)
PN-Agnostic (Sec. 3.2) vs. PN-Aware (Sec. 3.3)							
1	PN-Agnostic (Ours)	82.2	78.8	75.5	40.6	69.3	61.6
2	PN-Aware (Ours)	**84.9**	**83.6**	**78.9**	70.4	**79.4**	61.5
Points vs. Super-Segments (w/ PN-Aware)							
3	Points	40.7	0.2	38.1	10.8	22.5	57.2
4	P $\rightarrow$ Sp.-Seg.	39.2	0	44.1	63.3	36.6	57.2
5	**Sp.-Seg. (Ours)**	**84.9**	**83.6**	**78.9**	70.4	**79.4**	61.5
6	Upper Bound*	89.8	88.9	85.2	92.3	89.1	-
Ablation Study (w/ PN-Aware)							
7	w/o Unit Norm	78.5	81.0	77.4	54.4	72.8	63.0
8	$\sigma(\mathbf{X}) \rightarrow i$	80.8	77.5	75.3	56.6	72.5	63.4
9	$\sigma(\mathbf{X}) \rightarrow k$	73.8	76.1	75.8	**79.8**	76.4	61.9
10	$\sigma(\mathbf{X}) \rightarrow i \rightarrow k$	79.4	80.3	74.1	35.1	67.2	59.0
11	w/ Global Feat.	38.6	0.2	77.7	4.6	30.3	62.2
12	w/o $\mathcal{L}_{\text{CE}}$	82.6	79.7	77.4	71.4	77.8	59.8
Few-Shot Learning (w/ PN-Aware)							
13	k=1	85.5	83.5	78.4	73.2	80.1	59.4
14	k=8	86.1	84.2	78.9	70.6	79.9	60.0
15	k=32	86.9	84.8	79.5	76.5	81.9	59.7

Super-Segments vs. Points We also demonstrate the advantage of using super-segments as input in our pipeline. We compare our case with two baselines: 1) using the raw point cloud as input (row 3 in Table 2) and 2) using the point cloud but *projecting* the prediction result to the super-segments in test time (row 4 in Table 2). For the second, each point belonging to a super-segment votes for the part names, and the super-segment takes the part name of the majority. Row 6 in Table 2 shows an upper bound, when the part names are assigned to super-segments based on the ground truth segmentation of the underlying point cloud. When comparing our case using super-segments (row 2 and 5 in Table 2 — these two are the same) with these two cases, the mIoUs are significantly improved, and even our results are close to the upper bound. The low mIOUs of the second case (projecting point results to super-segments) show the value of these super-segments to be used in the training process, rather than just used in a post-processing step. See also the fifth column of Figure 4 for poor qualitative results. The target shape classification accuracy is also a bit increased when super-segments are used instead of points.

Ablation Study We also demonstrate through an ablation study that the details in our network pipeline are crucial for the part segmentation performance. From rows 7 to 12 in Table 2, we report the results of the following cases (in order): 1) when the query $f_a(u)$, key $g_k(s_i)$, and value $g_v(s_i)$ vectors are not normalized, 2) when the softmax σ is applied across the super-segments only, 3) across the part names only, 4) across super-segments *first* and *then* part names (a reverse order), 5) when adding a global feature to the feature of each super-segment, and 6) when not using the cross-entropy-based regularization loss (Equation 4).

From the results in the table and also in Figure 4, we can draw several conclusions. First, the normalization of query, key, and value vectors and also the softmax σ along with the part names before the super-segments improve overall mIoUs and particularly help detect optional parts accurately. See the arms in the sixth and seventh columns of Figure 4 compared to ours in the third column. Interestingly, with these, the attention is improved to be better aligned with the semantic parts while the accuracy of target shape classification is slightly decreased. Second, it is crucial to have the double softmax in the order of part names first and then super-segments. When the order is switched (the ninth column), or applying softmax only across part names (the eighth column), the overall quality of segmentation becomes worse; see the red circles in the eighth and ninth columns of Figure 4 for some failure examples. Third, as discussed in Section 3, the attention is not aligned with the semantic parts *at all* when a global feature is concatenated to a local feature of super-segments (the tenth column in Figure 4). The global feature is obtained by max-pooling the local features, following the idea of PointNet [26]. This result is obvious since the network can access the global shape information from any super-segment without carefully *attending* to a specific region. Last, the cross-entropy-based regularization improves mIoUs particularly for *seat*, which has the smallest utterances in the training dataset (2215 out of 32600), and also increases the target shape classification accuracy.

Few-Shot Learning We further investigate whether part segment annotations on a few shapes can improve the segmentation accuracy in a few-shot learning setup. Here, we only consider the PN-Aware case and also assume that a few shapes (1, 8, and 32) are given with ground truth part segmentation. Note that 1, 8, and 32 are very small numbers compared to 4509 number of entire shapes in the training dataset. We test exploiting the additional supervision by learning per-point classification with a cross entropy loss and the given annotated shapes after each epoch of the target shape discrimination task learning the attention. The results in Table 2 (rows 13-15) show improvements of mIoUs with the few-shot learning. Figure 5 also illustrates an example that the segmentation boundaries are refined even with a single-shot.

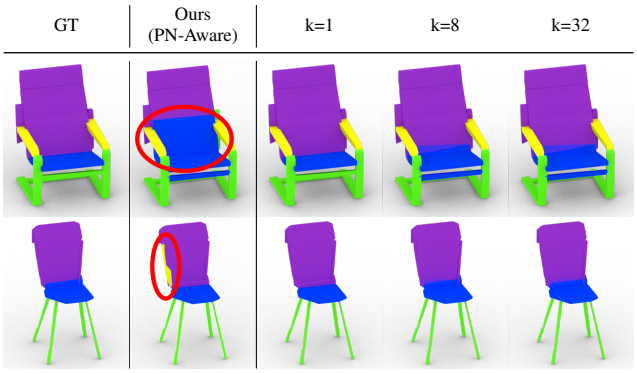


Figure 5. The effect of few-shot learning. Even a very small amount of ground truth eliminates things that the model might be confused about without supervision, such as the boundary between *back* and *seat*, and predicting the edge of the back as an arm.

Table 3. Out-of-distribution quantitative results. Our model can also segment out-of-distribution shapes. This table shows mIoUs with the attention maps learned with Chair parts and the part segments of the other classes in ShapeNet. Semantically corresponding parts have higher mIoUs, e.g., Chair *leg* → Table *leg* and Lamp *base*.

Other Classes		Chair (w/ PN-Aware)			
		Back	Seat	Leg	Arm
Table	Top	11.0	**78.2**	1.2	3.5
	Leg	4.5	2.8	**66.2**	11.0
	Connector	**26.5**	3.2	2.1	15.7
Lamps	Base	2.0	1.0	**44.6**	9.8
	Shade	27.5	**38.9**	7.1	16.6
	Canopy	4.9	7.0	5.1	**20.8**
	Tube	**21.4**	7.7	20.6	2.2

4.3. Out-of-Distribution Test

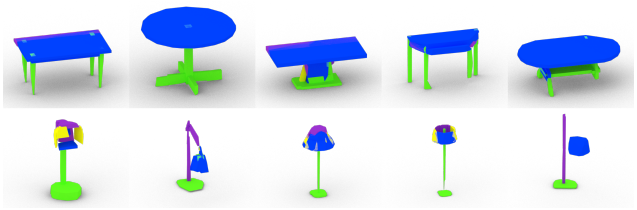


Figure 6. Out-of-Distribution Test. Each color indicates each predicted part as shown in Fig. 4. In each category, the model predicts the part as the semantically matching part in a chair, predicting lower part as *leg*.

We experiment how much the part segment information learned from the Chairs in the ***CiC*** dataset can be *zero-shot generalizable* to the other shape categories, namely, Tables and Lamps. Table 3 shows mIoUs across the parts of Chairs and the parts of Tables and Lamps. The results show very strong correlations between Chair *seat* and Table *top* as well as Chair *leg* and Table *leg*. Figure 6 also clearly shows boundaries of the Table *top* and *leg* segments. Even the information about Chair parts is well-generalizable to Lamps, a category that is largely different geometrically from Chairs. The second part of the table illustrates that Lamp *base* can be detected as Chair *leg*, and also Lamp *shade* is discriminated as Chair *back* and *seat*. The qualitative results are also shown in the second row of Figure 6.

4.4. Word Attention Visualizations

Figure 7. Word Attention of PN-Agnostic. Two encoders attend different words in the utterance to play different roles: "where to look" and "what shape it should be".

Our utterance encoders, *attention* encoder and *classification* encoder (which use the same architecture as ShapeGlot [3]) also learn attention over *words*, and we visualize the word attention for some examples in Figure 7. The color changes from dark blue to yellow when the attention weights for words increases from 0 to 1. Interestingly, the *attention* encoder mainly attends the *nouns* indicating the parts (the sentences above in each row), while the *classification* encoder rather focuses on a general context (the sentences below).

5. Conclusion

We proposed PartGlot, a framework learning part segmentation of 3D shape from linguistic descriptions. Without any direct supervision on part segmentation, our network classifying the target shape described by a given utterance can detect and segment part regions through an attention module. We not only introduced the first proposal of language-based 3D part segmentation but also designed a network curated for the emergence of part structure from the attention. We also proposed how predefined part names can be exploited in training to achieve the best performance. We finally demonstrated the part information learned by the network is transferable to other classes of shapes.

Negative Societal Impacts Our network can be potentially biased by harmful languages. However, we do not foresee any near-future risks of learning the connection between language and shape.

Limitations and Future Work While we learn attention both on words in an utterance and super-segments in a shape, learning correlation between these two is a challenging task. We plan to investigate this in future research.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 2012. 3

[2] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In *ECCV*, 2020. 2

[3] Panos Achlioptas, Judy Fan, X.D. Robert Hawkins, D. Noah Goodman, and J. Leonidas Guibas. ShapeGlot: Learning language for shape differentiation. 2019. 1, 2, 3, 5, 8

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[5] 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. *CoRR*, abs/1512.03012, 2015. 5, 6

[6] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. BSP-Net: Generating compact meshes via binary space partitioning. In *CVPR*, 2020. 3, 5

[7] Zhiqin Chen, Kangxue Yin, Matt Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-NET: Branched autoencoder for shape co-segmentation. In *ICCV*, 2019. 3

[8] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. CvxNet: Learnable convex decomposition. In *CVPR*, 2020. 3

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[10] Anastasia Dubrovina, Fei Xia, Panos Achlioptas, Mira Shalah, and Guibas J. Leonidas. Composite shape modeling via latent space factorization. *CoRR*, abs/1901.02968, 2019. 1

[11] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 2010. 1

[12] Mingtao Feng, Liang Zhang, Xuefei Lin, Syed Zulqarnain Gilani, and Ajmal Mian. Point attention network for semantic segmentation of 3D point clouds. *PR*, 2020. 2

[13] Natasha Gelfand and Leonidas J Guibas. Shape segmentation using local slippage analysis. In *SGP*, 2004. 3

[14] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *ICCV*, 2019. 3

[15] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. PCT: Point cloud transformer. 2021. 2

[16] Geoffrey Hinton. How to represent part-whole hierarchies in a neural network. *CoRR*, abs/2102.12627. 1

[17] Yining Hong, Qing Li, Song-Chun Zhu, and Siyuan Huang. VLGrammar: Grounded grammar induction of vision and language. *CoRR*, abs/2103.12975, 2021. 2

[18] Yining Hong, Qing Li, Song-Chun Zhu, and Siyuan Huang. VLGrammar: Grounded grammar induction of vision and language. In *ICCV*, 2021. 2

[19] 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*, 2021. 2

[20] Drew A Hudson and C. Lawrence Zitnick. Generative adversarial transformers. 2021. 2

[21] Faria Huq, Nafees Ahmed, and Anindya Iqbal. Static and animated 3D scene generation from free-form text descriptions. *CoRR*, abs/2010.01549, 2020. 2

[22] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. *CoRR*, abs/1805.07932, 2018. 3

[23] Kirill Mazur and Victor Lempitsky. Cloud Transformers: A universal approach to point cloud processing tasks. In *ICCV*, 2021. 2

[24] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural Parts: Learning expressive 3D shape abstractions with invertible neural networks. In *CVPR*, 2021. 3

[25] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics Revisited: Learning 3D shape parsing beyond cuboids. In *CVPR*, 2019. 3

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 3, 4, 6, 7

[27] Shi Qiu, Yunfan Wu, Saeed Anwar, and Chongyi Li. Investigating attention mechanism in 3D point cloud object detection. *CoRR*, abs/2108.00620, 2021. 2

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4

[29] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. LanguageRefer: Spatial-language model for 3D visual grounding. *CoRR*, abs/2107.03438, 2021. 2

[30] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. CLIP-Forge: Towards zero-shot text-to-shape generation. *CoRR*, abs/2110.02624, 2021. 2

[31] Jesse Thomason, Mohit Shridhar, Yonatan Bisk, Chris Paxton, and Luke Zettlemoyer. Language grounding with 3D objects. *CoRR*, abs/2107.12514, 2021. 2

[32] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, 2017. 3

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3

[34] Xiaogang Wang, Xun Sun, Xinyu Cao, Kai Xu, and Bin Zhou. Learning fine-grained segmentation of 3D shapes without part labels. In *CVPR*, 2021. 3

[35] Xiaoshi Wu, Hadar Averbuch-Elor, Jin Sun, and Noah Snavely. Towers of Babel: Combining images, language, and 3D geometry for learning multimodal vision. In *ICCV*, 2021. 2

[36] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[37] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PoinTr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, 2021. 2

[38] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. InstanceRefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *ICCV*, 2021. 2

[39] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 2

[40] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3DVG-Transformer: Relation modeling for visual grounding on point clouds. In *ICCV*, 2021. 2

[41] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas J. Guibas, and Hao Zhang. AdaCoSeg: Adaptive shape co-segmentation with group consistency loss. In *CVPR*, 2020. 3