

# ShapeGlot: Learning Language for Shape Differentiation

Panos Achlioptas\*      Judy Fan      Robert Hawkins  
Noah Goodman      Leonidas Guibas  
Stanford University

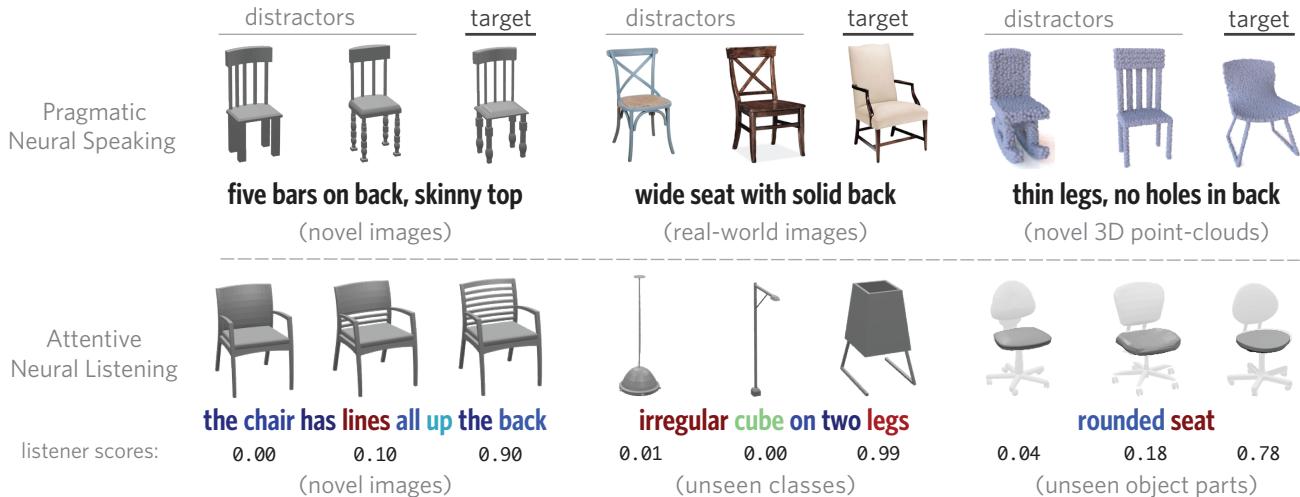


Figure 1: We introduce a large corpus of utterances that refer to the shape of objects and develop neural speakers and listeners with broad generalization capacity. **Top row:** A neural speaker generates utterances to distinguish a “target” from two “distractors” in *unseen*: images of synthetic data (left), out-of-distribution (OOD) *real-world* images (center), and 3D point-clouds of CAD models (right). **Bottom row:** A neural listener interprets human-generated utterances for *unseen* (left-to-right): images of synthetic data, OOD object *categories* (here, lamps), and OOD object *parts*. Listener scores indicate model interpretation about which object the utterance refers to. Words are color-coded according to the attention placed by the neural listener (warmer color indicates higher attention).

### Abstract

*In this work we explore how fine-grained differences between the **shapes** of common objects are expressed in language, grounded on **images and 3D models** of the objects. We first build a large scale, carefully controlled dataset of human utterances that each refers to a 2D rendering of a 3D CAD model so as to distinguish it from a set of shape-wise similar alternatives. Using this dataset, we develop neural language understanding (listening) and production (speaking) models that vary in their grounding (pure 3D forms via point-clouds vs. rendered 2D images), the degree of pragmatic reasoning captured (e.g. speakers that reason about a listener or not), and the neural ar-*

chitecture (e.g. with or without attention). We find models that perform well with both synthetic and human partners, and with held out utterances and objects. We also find that these models are amenable to **zero-shot** transfer learning to novel object classes (e.g. transfer from training on chairs to testing on lamps), as well as to real-world images drawn from furniture catalogs. Lesion studies indicate that the neural listeners depend heavily on part-related words and associate these words correctly with **visual parts** of objects (without any explicit network training on object parts), and that transfer to novel classes is most successful when known part-words are available. This work illustrates a practical approach to language grounding, and provides a case study in the relationship between object shape and linguistic structure when it comes to **object differentiation**.

\*Corresponding author: optas@cs.stanford.edu  
 Project webpage: <https://www.bit.ly/shapeplot>

## 1. Introduction

Objects are best understood in terms of their structure and function, both of which are built on a foundation of object parts and their relations [10, 9, 55, 8]. Natural languages have been optimized across human history to solve the problem of efficiently communicating the aspects of the world most relevant to one’s current goals [22, 12]. As such, languages can provide an effective medium to describe the *shapes* and the *parts* of different objects, and to express *object differences*. For example, when we see a chair we can decompose it into semantically meaningful parts, like a *back* and a *seat*, and can combine words to create utterances that reflect their geometric and topological *shape-properties* e.g. ‘wide seat with a solid back. Moreover, given a specific communication context, we can craft references that are not merely true, but which are also relevant: i.e. we can refer to the lines found in a chair’s back to *distinguish* it among other similar objects (see Fig. 1).

In this paper we explore this interplay between natural, referential language, and the shape of common objects. While a great deal of recent work has explored visually-grounded language understanding [20, 32, 52, 27, 26, 51], the resulting models have limited capacity to reflect the geometry and topology (i.e. the shape) of the underlying objects. This is because reference in previous studies was possible using properties like *color*, or properties regarding the object and its hosting environment (e.g. its absolute, or relative to other objects, *location*). Indeed, eliciting natural language that refers only to shape properties requires carefully controlling the objects, their presentation, and the linguistic task. To address such challenges, we use pure 3D representations of objects (CAD models), which allow for flexible and *controlled* presentation (i.e. textureless, uniform-color objects, viewed without obstruction in a fixed pose). We further make use of the 3D form to construct a reference game task in which the referred object is similar *shape-wise* to the contrasting objects. The result of this effort is a new multimodal dataset, termed **CiC (Chairs in Context)**, comprised of 4,511 unique chairs from ShapeNet [4] and 78,789 referential utterances. In CiC chairs are organized into 4,054 sets of size 3 (representing contrastive communication contexts) and each utterance is intended to distinguish a chair in context. The visual differences among the grouped objects require a deep understanding of very fine-grained shape properties (especially, for *Hard* contexts, see Section 2); the language that people use to do so is correspondingly complex, exhibiting rich compositionality.

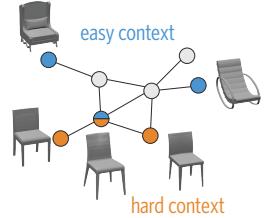
We use CiC to train and analyze a variety of modern neural language understanding (listening) and production

(speaking) models. These models vary in their grounding (pure 3D forms via point-clouds vs. rendered 2D images), the degree of pragmatic reasoning captured (e.g. speakers that reason about a listener or not) and the neural architecture (e.g. with or without word attention, and with context-free or context-aware object encoders). We evaluate these models on the original reference game task with both synthetic and human partners, and with held out utterances and objects, finding strong performance. Since language conveys abstractions, such as object parts, that are shared between object categories, we hypothesized that our models learn *robust* representations that are transferable to objects of unseen classes (e.g. training on chairs while testing on lamps). Indeed, we show that these models have strong generalization capacity to novel object *categories*, as well as to *real-world* colored images drawn from furniture catalogs.

Finally, we explore *how* our models are succeeding on these communication tasks. We demonstrate that the neural listener learns to prioritize the same abstractions in objects (i.e. properties of chair parts) that humans do in solving the communication task, despite *never* being provided with an explicit decomposition of these objects into parts. Similarly, we show that transfer learning to novel object classes is most successful when known part-related words are available. Last, we show that a neural speaker that is *pragmatic*—planing utterances in order to convey the right target object to an imagined listener —produces significantly more informative utterances than a *literal* (listener-unaware) speaker, as measured by human performance in identifying the correct object.

## 2. Dataset and task

**CiC (Chairs in Context)** consists of triplets of chairs coupled with referential utterances that aim to distinguish one chair (the “target”) from the remaining two (the “distractors”). To obtain such utterances, we paired participants from Amazon’s Mechanical Turk (AMT) to play an online reference game [16]. On each round of the game, the two players were shown the same triplet of chairs. The designated target chair was privately highlighted for one player (the “speaker”) who was asked to send a message through a chat box such that their partner (the “listener”) could successfully select it from the context. To ensure speakers used *only* shape-related information, we scrambled the positions of the chairs for each participant independently and used textureless, uniform-color renderings of pre-aligned 3D CAD models, taken from the same viewpoint. To ensure communicative interaction was natural, no constraints were placed on the chat



box: referring expressions from the speaker were occasionally followed by clarification questions from the listener or other discourse.

A key decision in building our dataset concerned the construction of contexts that would reliably elicit *diverse* and potentially *very* fine-grained contrastive language. To achieve diversity we considered all  $\sim 7,000$  chairs from ShapeNet. This object class is geometrically complex, highly diverse, and abundant in the real world. To control the granularity of fine-grained distinctions that were necessary in solving the communication task, we constructed two types of contexts: *Hard* contexts consisted of very similar shape-wise chairs, and *Easy* contexts consisted of less similar chairs. To measure shape-similarity in an unsupervised manner, we used the latent space derived from an Point Cloud-AutoEncoder (PC-AE) [1]. We note, that point-clouds are an intrinsic representation of a 3D object, *oblique* to color or texture. After extracting a 3D point-cloud from the surface of each ShapeNet model we computed the underlying K-nearest-neighbor graph among all models according to their PC-AE embedding distances. For a chair with sufficiently high-in degree on this graph (corresponding intuitively to a canonical chair) we contrasted it with four distractors: the two *closest* to it in latent-space, and two that were sufficiently far (see inset for a demonstration and at the Appendix for additional details). Last, to reduce potential data biases we *counterbalanced* each communication context, by considering every chair of a given context as target, in at least four games.

Before presenting our neural agents, we identify some distinctive properties of our corpus. Human performance on the reference game was high, but listeners made significantly more errors in the Hard triplets (accuracy 94.2% vs. 97.2%,  $z = 13.54, p < 0.001$ ). Also, in Hard triplets longer utterances were used to describe the targets (on average 8.4 words vs. 6.1,  $t = -35, p < 0.001$ ). A wide spectrum of descriptions was elicited, ranging from the more holistic/categorical (e.g. “the rocking chair”) common for Easy triplets, to more complex and fine-grained language, (e.g. “thinner legs but without armrests”) common for Hard triplets. Interestingly, 78% of the utterances used at least one part-related word: “back”, “legs”, “seat,” “arms”, or closely related synonyms e.g. “armrests”.

### 3. Neural listeners

Constructing neural listeners that reason effectively about shape properties is a key contribution of our work. Below we conduct a detailed comparison between three distinct architectures, highlight the effect of different regularization techniques, and investigate the merits of different representations of 3D objects for the listening task, namely, 2D rendered images and 3D surface point clouds. In what follows, we denote the three objects of a commu-

nication context as  $O = \{o_1, o_2, o_3\}$ , the corresponding word-tokenized utterance as  $U = u_1, u_2, \dots$  and as  $t \in O$  the designated target.

Our proposed listener is inspired by [31]. It takes as input a (latent code) vector that captures shape information for each of the objects in  $O$ , and a (latent code) vector for each token of  $U$ , and outputs an object–utterance compatibility score  $\mathcal{L}(o_i, U) \in [0, 1]$  for each input object. At its core lies a multi-modal LSTM [17] that receives as input (“is grounded” with) the vector of a single object, processes the word-sequence  $U$ , and is read out by a final MLP to yield a single number (the compatibility score). This is repeated for each  $o_i$ , *sharing* all network parameters across the objects. The resulting three scores are soft-max normalized and compared to the ground-truth indicator vector of the target under the cross-entropy loss.\*

**Object encoders** We experimented with three object representations to capture the underlying shapes: (a) the bottleneck vector of a pretrained Point Cloud-AutoEncoder (PC-AE), (b) the embedding provided by a convolutional network operating on single-view images of non-textured 3D objects, or (c) a combination of (a) and (b). Specifically, for (a) we use the PC-AE architecture of [1] trained with single-class point clouds extracted from the surfaces of 3D CAD models, while for (b) we use the activations of the penultimate layer of a VGG-16 [38], pre-trained on ImageNet [7], and fine-tuned on an 8-way classification task with images of objects from ShapeNet. For each representation we project the corresponding latent code vector to the input space of the LSTM using a fully connected (FC) layer with  $L_2$ -norm weight regularization. The addition of these projection-like layers improves the training and convergence of our system.

While there are many ways to simultaneously incorporate the two modalities in the LSTM, we found that the best performance resulted when we ground the LSTM with the image code, concatenate the LSTM’s final output (after processing  $U$ ) with the point cloud code, and feed the concatenated result in a shallow MLP to produce the compatibility score. We note that grounding the LSTM with point clouds and using images towards the end of the pipeline, resulted in a significant performance drop ( $\sim 4.8\%$  on average). Also, proper regularization was *critical*: adding dropout at the input layer of the LSTM and  $L_2$  weight regularization and dropout at and before the FC projecting layers improved performance  $\sim 10\%$ . The token codes of each sentence where initialized with the GloVe embedding [35] and fine-tuned for the listening task.

**Incorporating context information** Our proposed baseline listener architecture (*Baseline*, just described) first scores each object *separately* then applies softmax normal-

\*Architecture details, hyper-parameter search strategy, and optimal hyper-parameters for all experiments are described in the Appendix.

ization to yield a score distribution over the three objects. We also consider two alternative architectures that explicitly encode information about the *entire* context while scoring an object. The first alternative (*Early-Context*), is identical to the proposed architecture, except for the codes used to ground the LSTM. Specifically, if  $v_i$  is the image code vector of the  $i$ -th object ( $o_i \in O$ ) resulting from VGG, instead of using  $v_i$  as the grounding vector of  $o_i$ , a shallow convolutional network is introduced. This network, of which the output is the grounding code, receives the signal  $f(v_j, v_k) \parallel g(v_j, v_k) \parallel v_i$ , where  $f, g$  are the symmetric (and norm-normalized), max-pool and mean-pool functions,  $\parallel$  denotes feature-wise concatenation and  $v_j, v_k$  the alternative contrastive objects. Here, we use symmetric functions to induce that object-order is irrelevant for our task. The second alternative architecture (*Combined-Interpretation*) first feeds the image vectors for *all* three objects sequentially to the LSTM as inputs and then proceeds to process the tokens of  $U$  once, to yield the three scores. Similarly to the *Baseline* architecture, point clouds are incorporated in both alternatives via a separate MLP after the LSTM.

**Word attention** We hypothesized that a listener forced to prioritize a few words in each utterance would learn to prioritize words that express properties that distinguish the target from the distractors (and, thus, perform better). To test this hypothesis, we augment the listener models with a standard *bilinear attention mechanism* [37]. Specifically, to estimate the “importance” of each text-token  $u_i$  we compare the output of the LSTM at  $u_i$  (denoted as  $r_i$ ) with the hidden state after the entire utterance has been processed (denoted as  $h$ ). The relative importance of each word is  $a_i \triangleq r_i^T \times W_{\text{att}} \times h$ , where  $W_{\text{att}}$  is a trainable diagonal matrix. The final output of the LSTM uses this attention to combine all latent states:  $\sum_{i=1}^{|U|} r_i \odot \hat{a}_i$ , where  $\hat{a}_i = \frac{\exp(a_i)}{\sum_j^{|U|} \exp(a_j)}$  and  $\odot$  is the point-wise product.

## 4. Listener experiments

We begin our evaluation of the proposed listeners using two reference tasks based on different data splits. In the *language generalization* task, we test on target objects that were seen as targets in at least one context during training but ensure that all utterances in the test split are from unseen speakers. In the more challenging *object generalization* task, we restrict the set of objects that appeared as targets in the test set to be *disjoint* from those in training such that all speakers *and* objects in the test split are new. For each of these tasks, we evaluate choices of input modality and word attention, using [80%, 10%, 10%] of the data, for training, validating and testing purposes.

*Baseline* listener accuracies are shown in Table 2.<sup>†</sup> Over-

<sup>†</sup>In all results mean accuracies and standard errors across 5 random seeds are reported, to control for the data-split populations and the initial-

all the model achieves good performance. As expected, all listeners have higher accuracy on the language generalization task (3.2% on average). The attention mechanism on words yields a mild performance boost, as long as images are part of the input. Interestingly, images provide a significantly better input than point-clouds when only one modality is used. This may be due to the higher-frequency content of images (we use point-clouds with only 2048 points), or the fact that VGG was pre-trained while the PC-AE was not. However, we find *significant* gains in accuracy (4.1% on average) from exploiting the two object representations *simultaneously*, implying a complementarity among them.

Next, we evaluate how the different approaches in incorporating context information described in Section 3 affect listener performance. We focus on the more challenging object generalization task, using listeners that include attention and both object modalities. We report the findings in Table 1. We find that the *Baseline* and *Early-Context* models perform best overall, outperforming the *Combined-Interpretation* model, which does not share weights across objects. This pattern held for both hard and easy trial types in our dataset. We further explored the small portion (~14%) of our test set that use explicitly contrastive language: superlatives (“skinniest”) and comparatives (“skinier”). Somewhat surprisingly we find that the *Baseline* architecture remains competitive against the architectures with more explicit context information. The *Baseline* model thus achieves high performance and is the most flexible (at test time it can be applied to *arbitrary-sized* contexts); we focus on this architecture in the explorations below.

### 4.1. Exploring learned representations

**Linguistic ablations** Which aspects of a sentence are most critical for our listener’s performance? To inspect the properties of words receiving the most attention, we ran a part-of-speech tagger on our corpus. We found that the highest attention weight is placed on *nouns*, controlling for the length of the utterance. However, adjectives that *modify* nouns received more attention in hard contexts (controlling for the average occurrence in each context), where nouns are often not sufficient to disambiguate (see Fig. 2A). To more systematically evaluate the role of higher-attention tokens in listener performance, we conducted an utterance lesioning experiment. For each utterance in our dataset, we successively replaced words with the <UNK> token according to three schemes: (1) from highest attention to lowest, (2) from lowest attention to highest, and (3) in random order. We then fed these through an equivalent listener trained *without* attention. We found that up to 50% of words can be removed without much performance degradation, but only if these are low attention words (see Fig. 2B). Our word attentive listener thus appears to rely on context-appropriate

ization of the neural-network.

Architecture	Overall	Subpopulations		
		Hard	Easy	Sup-Comp
<i>Combined-Interpretation</i>	$75.9 \pm 0.5\%$	$67.4 \pm 1.0\%$	$83.8 \pm 0.6\%$	$74.4 \pm 1.5\%$
<i>Early-Context</i>	$79.4 \pm 0.8\%$	$\mathbf{70.1} \pm 1.3\%$	$88.1 \pm 0.6\%$	$75.6 \pm 2.2\%$
<i>Baseline</i>	$\mathbf{79.6} \pm 0.8\%$	$69.9 \pm 1.3\%$	$\mathbf{88.8} \pm 0.4\%$	$\mathbf{76.3} \pm 1.3\%$

Table 1: Comparing different ways to include context. The simplest *Baseline* model performs as well as more complex alternatives. Subpopulations are the subset of test data containing: hard trials (shape-wise similar distractors), easy trials, superlatives or comparatives.

	Input Modality	Language Task	Object Task
<b>No Attention</b>	Point Cloud	$67.6 \pm 0.3\%$	$66.4 \pm 0.7\%$
	Image	$81.2 \pm 0.5\%$	$77.4 \pm 0.7\%$
	Both	$83.1 \pm 0.4\%$	$78.9 \pm 1.0\%$
<b>With Attention</b>	Point Cloud	$67.4 \pm 0.3\%$	$65.6 \pm 1.4\%$
	Image	$81.7 \pm 0.5\%$	$77.6 \pm 0.8\%$
	Both	$\mathbf{83.7} \pm 0.3\%$	$\mathbf{79.6} \pm 0.8\%$

Table 2: Performance of the *Baseline* listener architecture using different object representations and with/without word level attention, in two generalization tasks.

content words to successfully disambiguate the referent.

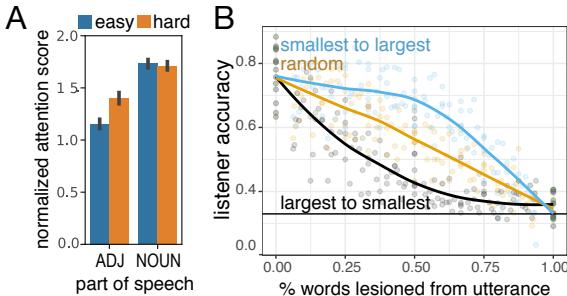


Figure 2: (A) The listener places more attention on adjectives in hard (orange) triplets than easy (blue) ones. The histogram’s heights depict mean attention scores normalized by the length of the underlying utterances; the error bars are bootstrapped 95% confidence intervals. (B) Lesioning highest attention words to lowest worsens performance more than lesioning random words or lesioning lowest attention words.

**Visual ablations** To test the extent to which our listener is relying on the same semantic *parts* of the object as humans, we next conducted a lesion experiment on the visual input. We took the subset of our test set where (1) all chairs had complete part annotations available [50] and (2) the corresponding utterance mentioned a *single* part (17% of our test set). We then created lesioned versions of all three objects

on each trial by removing pixels of images (and/or points when point-clouds are used), corresponding to parts according to two schemes: *removing* a single part or *keeping* a single part. We did this either for the mentioned one, or another part, chosen at random. We report listener accuracies on these lesioned objects in Table 3. We found that removing random parts hurts the accuracy by 10.4% on average, but removing the mentioned part dropped accuracy more than three times as much, nearly to chance. Conversely, keeping only the mentioned part while lesioning the rest of the image merely drops accuracy by 10.6% while keeping a non-mentioned (random) part alone brings accuracy down close to chance. In other words, on trials when participants depended on information about a part to communicate the object to their partner, we found that visual information about that part was both *necessary and sufficient* for the performance of our listener model.

	Single Part Lesioned	Single Part Present
<b>Mentioned Part</b>	$42.8\% \pm 2.3$	$66.8\% \pm 1.4$
<b>Random Part</b>	$67.0\% \pm 2.9$	$38.8\% \pm 2.0$

Table 3: Evaluating the part-awareness of neural listeners by lesioning object *parts*. Results shown are for image-only listeners, with average accuracy of 77.4% when *intact* objects are used. Similar findings regarding point-cloud-based lesioning are provided in the Appendix.

## 5. Neural speakers

**Architecture** We next explore models that learn to generate an utterance that refers to the target and which distinguishes it from the distractors. Similarly to a neural listener the heart of these models is an LSTM which encodes the objects of a communication context, and then decodes an utterance. Specifically, for an *image-based* model, on the first three time steps, the LSTM input is the VGG code of each object. Correspondingly, for a *point-cloud-based* model, the LSTM input is the object codes extracted from a PC-AE. During training and after the objects are encoded, the remaining input to the LSTM is the ‘current’ utterance token,

while the output of the LSTM is compared with the ‘next’ utterance token, under the cross-entropy loss [46]. The target object is always presented last, eliminating the need to represent the index of the target separately. To find the best model hyper-parameters (e.g.  $L_2$ -weights, dropout-rate and # of LSTM neurons) and the optimal stopping epoch, we sample synthetic utterances from the model during training and use a pretrained *listener* to select the result with the highest listener accuracy. We found this approach to produce models and parameters that yield better quality utterances than evaluating with listening-unaware metrics like BLEU [34].

**Variations** The above (*literal*) speakers can learn to generate language that discriminates targets from distractors. To test the degree to which distractor objects are used for generation, we experiment with *context-unaware* speakers that are provided the encoding of the target object *only*, and are otherwise identical to the above models. Motivated by the recursive social reasoning characteristic of human pragmatic language use (as formalized in the Rational Speech Act framework [13]), we create *pragmatic* speakers that choose utterances according to their capacity to be discriminative, as judged by a pretrained “internal” listener. In this case, we sample utterances from the (*literal*) speakers, but score (i.e. re-rank) them with:

$$\beta \log(P_L(t|U, O)) + \frac{(1 - \beta)}{|U|^\alpha} \log(P_S(U|O, t)), \quad (1)$$

where  $P_L$  is the listener’s probability to predict the target ( $t$ ) and  $P_S$  is the likelihood of the *literal* speaker to generate  $U$ . The parameter  $\alpha$  controls a length-penalty term to discourage short sentences [48], while  $\beta$  controls the relative importance of the speaker’s vs. the listener’s opinions.

## 6. Speaker experiments

Qualitatively, our speakers produce good object descriptions, see Fig. 3 for examples, with the pragmatic speakers yielding more discriminating utterances.<sup>‡</sup> To quantitatively evaluate the speakers we measure their success in reference games with two different kinds of partners: with an independently-trained listener model and with human listeners. To conduct a *fair* study when we used a neural listener, we split the training data in half. The evaluating listener was trained using one half, while the scoring (or “internal”) listener used by the pragmatic speaker was trained on the remaining half. For our human evaluation, we used the *literal* and *pragmatic* variants to generate referring expressions on the test set (we use all training data to train the internal listeners here). We then showed these referring expressions to participants recruited

on AMT and asked them to select the object from context that the speaker was referring to. We collected approximately 2.2 responses for each triplet (we use 1200 unique triplets from the *object-generalization* test-split, annotated separately by each speaker model). The synthetic utterances used were the highest scoring ones (Eq. 1) for each model with optimal (per-validation)  $\alpha$  and a  $\beta = 1.0$ . We note that while the *point-based* speakers operate *solely* with point-cloud representations, we present their produced utterances to AMT participants accompanied by CAD rendered images, to keep the human-side presentation identical across experiments.

Table 4: Evaluating neural speakers operating with Point Cloud or Image object representations.

Speaker Architecture	Modality	Neural Listener	Human Listener
Context Unaware	Point Cloud	$59.1 \pm 2.0\%$	-
	Image	$64.0 \pm 1.7\%$	-
Literal	Point Cloud	$71.5 \pm 1.3\%$	66.2
	Image	$76.6 \pm 1.0\%$	68.3
Pragmatic	Point Cloud	$90.3 \pm 1.3\%$	69.4
	Image	<b><math>92.2 \pm 0.5\%</math></b>	<b>78.7</b>

We found (see Table 4) that our *pragmatic* speakers perform best with both synthetic and human partners. While their success with the synthetic listener model may be unsurprising, given the architectural similarity of the internal listener and the evaluating listener, *human* listeners were 10.4 percentage points better at picking out the target on utterances produced by the *pragmatic* vs. *literal* speaker for the best-performing (*image-based*) variant. We also found an asymmetry between the listening and speaking tasks: while context-unaware listeners achieved high performance, we found that context-unaware speakers fare significantly worse than context-aware ones. Last, we note that both literal and pragmatic speakers produce *succinct* descriptions (average sentence length 4.21 vs. 4.97) but the pragmatic speakers use a much richer vocabulary (14% more unique nouns and 33% more unique adjectives, after controlling for average length discrepancy).

## 7. Out-of-distribution transfer learning

Language is abstract and compositional. These properties make language use generalizable to new situations (e.g. using concrete language in novel scientific domains) and robust to low-level perceptual variation (e.g. lighting). In our final set of experiments we examine the degree to which our neural listeners and speakers learn representations that are correspondingly *robust*: that capture associations between the visual and the linguistic domains permit generalization out of the training domain.

<sup>‡</sup>The project’s webpage contains additional qualitative results.

	distractors		target	distractors		target	distractors		target
listener scores	0.29	0.20	0.51	0.00	0.14	0.86	0.19	0.24	0.57
<b>pragmatic</b> speaker	<b>it has rollers on the feet</b>		<b>square back, straight legs</b>		<b>thin-est seat</b>				
listener scores	0.55	0.16	0.29	0.05	0.85	0.10	0.19	0.32	0.49
<b>literal</b> speaker	<b>the one with the circle on the bottom</b>		<b>the one with the thick-est legs</b>		<b>the chair with the thin-est legs</b>				

Figure 3: *Pragmatic* vs. *literal* speakers in *unseen* (‘hard’) contexts. The pragmatic generations successfully discern the target, even in cases where the literal ones fail. The two left-most examples are based on image-based speakers/listeners, the right-most with point-cloud-based. The utterances are color coded according to the attention placed by an evaluating neural listener whose classification scores are shown above each corresponding utterance.

**Understanding out-of-class reference** To test the generalization of listeners to novel stimuli, we collected referring expressions in communication contexts made of objects in ShapeNet drawn from new classes: beds, lamps, sofas and tables. These classes are distinct from chairs, but share some parts and properties, making transfer possible for a sufficiently compositional model. For each of these classes we created 200 contexts made of random triplets of objects; we collected 2 referring expressions for each target in each context (from participants on AMT). Examples of visual stimuli and collected utterances are shown in Fig. 4 (bottom-row). To this data, we applied an (image-only, with/without-attention) listener trained on the CiC (i.e. chairs) data. We avoid using point-clouds since unlike VGG which was finetuned with multiple ShapeNet classes, the PC-AE was pre-trained on a single-class.

As shown in Table 5, the average accuracy is well above chance in all transfer categories (56% on average). Moreover, constraining the evaluation to utterances that contain *only* words that are in the CiC training vocabulary (75% of all utterances, column: *known*) only slightly improves the results. This is likely because utterances with unknown words still contain enough known vocabulary for the model to determine meaning. We further dissect the *known* population into utterances that contain part-related words (*with-part*) and their complement (*without-part*). For the training domain of chairs without-part utterances yield slightly higher accuracy. However the useful subcategories that support this performance (e.g. “recliner”) do not support transfer to new categories. Indeed, we observe that for transfer classes (except sofa) the listener performs better when part-related words are present. Furthermore, the performance gap between the two populations appears to become larger as the perceptual distance between the transfer and training domains increases (compare sofas to lamps).

Table 5: Transfer-learning of neural listeners in novel object *classes*, and in different subpopulations of utterances. The subpopulations are: *entire*: all utterances, *known*: with all tokens in the chair training-vocabulary, *with-part*: subset of *known* that contain at least one part-related word, *without-part* subset of *known* and complement of *with-part*. For reference the test-chair statistics are shown (first row), but are not included in the reported average (last row). The accuracies are *averages* of five listeners trained on different data splits. Further details are provided in the Appendix.

Class	Population			
	entire	known	with part	without part
chair	77.4	77.8	77.0	<b>80.5</b>
bed	56.4	55.8	<b>63.8</b>	51.5
lamp	50.1	51.9	<b>60.3</b>	47.1
sofa	53.6	55.0	<b>55.1</b>	54.7
table	63.7	65.5	<b>68.3</b>	62.7
average	56.0	57.1	<b>61.9</b>	54.9

**Describing real images** Transfer from synthetic data to real data is often difficult for modern machine learning models, that are attuned to subtle statistics of the data. We explored the ability of our models to transfer to real chair images (rather than the training images which were rendered without color or texture from CAD models) by curating a modest-sized (300) collection of chair images from online furniture catalogs. These images were taken from a *similar* view-point to that of the training renderings and have rich color and texture content. We applied the (image-only) *pragmatic* speaker to these images, after subtracting the average ImageNet RGB values (i.e. before passing the images to VGG). Examples of the speaker’s productions are shown in Figure 4. For each chair, we randomly selected

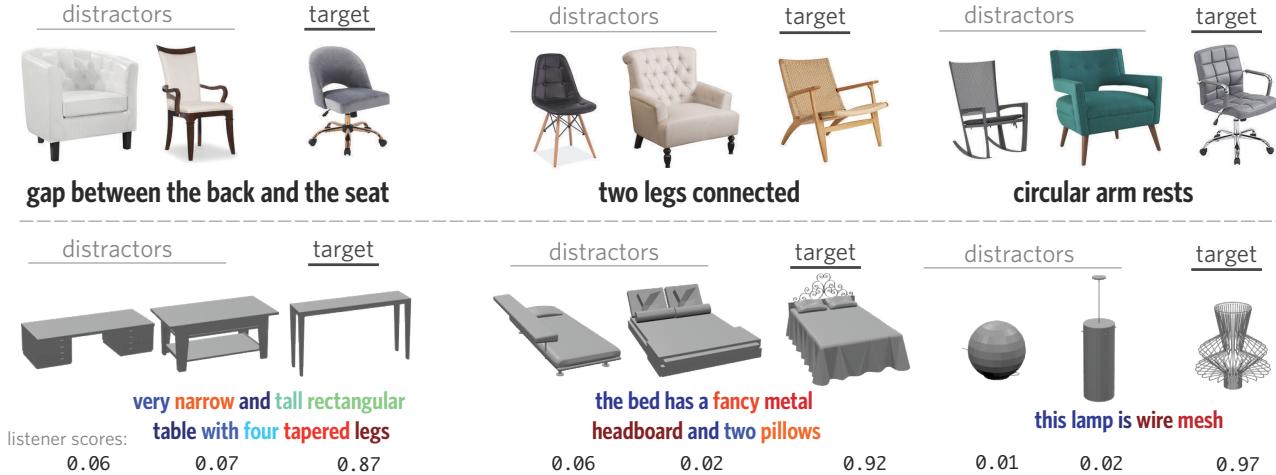


Figure 4: Examples of *out-of-distribution* neural speaking and listening. **Top row:** model generations for *real-world* catalogue images. The speaker successfully describes fine grained shape differences on images with rich color and texture content, not present in the training data. **Bottom row:** results of applying a word-attentive listener on renderings of CAD objects from *unseen* classes with human-produced utterances. The listener can detect the (often localized) visual cues that humans refer to, despite the large visual discrepancy of these objects from training-domain of chairs. (The utterances are color coded according to the listener’s attention.)

two distractors and asked 2 AMT participants to guess the target given the utterance produced by our speaker. Human listeners correctly guessed the target chair 70.1% of the time. Our speaker appears to transfer successfully to real images, which contain color, texture, pose variation, and likely other differences from our training data.

## 8. Related work

**Image labeling and captioning** Our work builds on recent progress in the development of vision models that involve some amount of language data, including object categorization [38, 54] and image captioning [19, 45, 49]. Unlike object categorization, which pre-specifies a fixed set of class labels to which all images must project, our systems use open-ended, referential language. Similarly to other recent works in image captioning [29, 32, 52, 43, 27, 26, 51], instead of captioning a single image (or entity therein), in isolation, our systems learn how to communicate across diverse communications contexts.

**Reference games** In our work we use reference games [20] in order to operationalize the demand to be relevant in context. The basic arrangement of such games can be traced back to the language games explored by Wittgenstein [47] and Lewis [25]. For decades, such games have been a valuable tool in cognitive science to quantitatively measure inferences about language use and the behavioral consequences of those inferences [36, 23, 5, 42]. Recently, these approaches have also been adopted as a benchmark for discriminative or context-aware NLP [33, 2, 40, 44, 31, 6, 24].

**Rational speech acts framework** Our models draw on recent formalization of human language use in the Rational Speech Acts (RSA) framework [13]. At the core of RSA is the Gricean proposal [15] that speakers are agents who select utterances that are parsimonious yet informative about the state of the world. RSA formalizes this notion of informativity as the expected reduction in the uncertainty of an (internally simulated) listener, as our pragmatic speaker does. The literal listener in RSA uses semantics that measure compatibility between an utterance and a situation, as our baseline listener does. Previous work has shown that RSA models account for context sensitivity in speakers and listeners [14, 31, 53, 11]. Our results add evidence for the effectiveness of this approach in complex domains.

## 9. Conclusion

In this paper, we explored models of natural language grounded in the shape of common objects. The geometry and topology of objects can be complex and the language we have for referring to them is correspondingly abstract and compositional. This makes the shape of objects an ideal domain for exploring grounded language learning, while making language an especially intriguing source of evidence for shape variations. We introduced the Chairs-in-Context corpus of highly descriptive referring expressions for shapes in context. Using this data we explored a variety of neural listener and speaker models, finding that the best variants exhibited strong performance. These models draw on both 2D and 3D object representations and appear

to reflect human-like part decomposition, though they were never explicitly trained with object parts. Finally, we found that the learned models are surprisingly robust, transferring to real images and to new classes of objects. Future work will be required to understand the transfer abilities of these models and how this depends on the compositional structure they have learned.

**Acknowledgements** The authors wish to acknowledge the support of a Sony Stanford Graduate Fellowship, a NSF grant CHS-1528025, a Vannevar Bush Faculty Fellowship and gifts from Autodesk and Amazon Web Services for Machine Learning Research.

## References

- [1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. J. Guibas. Learning representations and generative models for 3d point clouds. *Proceedings of the 35th International Conference on Machine Learning*, 2018. [3](#), [11](#)
- [2] J. Andreas and D. Klein. Reasoning about pragmatics with neural listeners and speakers. *CoRR*, 2016. [8](#)
- [3] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. [11](#)
- [4] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. [2](#), [11](#)
- [5] H. H. Clark and D. Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22(1):1–39, 1986. [8](#)
- [6] R. Cohn-Gordon, N. Goodman, and C. Potts. Pragmatically informative image captioning with character-level reference. *CoRR*, abs/1804.05417, 2018. [8](#)
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. [3](#)
- [8] A. Dubrovina, F. Xia, P. Achlioptas, M. Shahal, and G. J. Leonidas. Composite shape modeling via latent space factorization. *CoRR*, abs/1901.02968, 2019. [2](#)
- [9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010. [2](#)
- [10] A. M. Fischler and E. A. Robert. The representation and matching of pictorial structures. *IEEE Trans. on Computers*, 1973. [2](#)
- [11] D. Fried, J. Andreas, and D. Klein. Unified pragmatic models for generating and following instructions. *CoRR*, abs/1711.04987, 2017. [8](#)
- [12] E. Gibson, R. Futrell, J. Jara-Ettinger, K. Mahowald, L. Bergen, S. Ratnasingam, M. Gibson, S. T. Piantadosi, and B. R. Conway. Color naming across languages reflects color use. *Proceedings of the National Academy of Sciences*, 114(40):10785–10790, 2017. [2](#)
- [13] N. D. Goodman and M. C. Frank. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, 20(11):818 – 829, 2016. [6](#), [8](#)
- [14] C. Graf, J. Degen, R. X. D. Hawkins, and N. D. Goodman. Animal, dog, or dalmatian? level of abstraction in nominal referring expressions. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society*, 2016. [8](#)
- [15] H. P. Grice. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics*, pages 43–58. Academic Press, New York, 1975. [8](#)
- [16] R. X. D. Hawkins. Conducting real-time multiplayer experiments on the web. *Behavior Research Methods*, 47(4):966–976, 2015. [2](#)
- [17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. [3](#)
- [18] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. [11](#)
- [19] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. [8](#), [13](#)
- [20] S. Kazemzadeh, V. Ordonez, M. Mark, and B. L. Tamara. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. [2](#), [8](#)
- [21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. [12](#)
- [22] S. Kirby, M. Tamariz, H. Cornish, and K. Smith. Compression and communication in the cultural evolution of linguistic structure. *Cognition*, 141:87–102, 2015. [2](#)
- [23] R. M. Krauss and S. Weinheimer. Changes in reference phrases as a function of frequency of usage in social interaction: A preliminary study. *Psychonomic Science*, 1964. [8](#)
- [24] A. Lazaridou, K. M. Hermann, K. Tuyls, and S. Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. *CoRR*, abs/1804.03984, 2018. [8](#)
- [25] D. Lewis. *Convention: A philosophical study*. Harvard University Press, 1969. [8](#)
- [26] J. Lu, J. Yang, D. Batra, and D. Parikh. Neural baby talk. *CVPR*, 2018. [2](#), [8](#)
- [27] R. Luo and G. Shakhnarovich. Comprehension-guided referring expressions. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. [2](#), [8](#)
- [28] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013. [11](#)
- [29] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and M. Kevin. Generation and comprehension of unambiguous object descriptions. *CoRR*, abs/1511.02283, 2016. [8](#)
- [30] T. Miyato, D. M. Andrew, and G. Ian. Adversarial training methods for semi-supervised text classification. *International Conference on Learning*, 2017. [11](#)
- [31] W. Monroe, R. X. Hawkins, N. D. Goodman, and C. Potts. Colors in context: A pragmatic neural model for grounded language understanding. *CoRR*, abs/1703.10186, 2017. [3](#), [8](#)
- [32] K. V. Nagaraja, I. V. Morariu, and D. S. Larry. Modeling context between objects for referring expression understanding. *ECCV*, 2016. [2](#), [8](#)

- [33] M. Paetz, D. N. Racca, and D. DeVault. A multimodal corpus of rapid dialogue games. In *Language Resources and Evaluation Conference (LREC)*, 2014. 8
- [34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 6
- [35] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 3
- [36] S. Rosenberg and B. D. Cohen. Speakers' and listeners' processes in a word-communication task. *Science*, 1964. 8
- [37] S. Shen and H. Lee. Neural attention models for sequence classification: Analysis and application to key term extraction and dialogue act detection. *CoRR*, abs/1604.00077, 2016. 4
- [38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 8
- [39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1), 2014. 11
- [40] J.-C. Su, C. Wu, H. Jiang, and S. Maji. Reasoning about fine-grained attribute phrases using reference games. *CoRR*, abs/1708.08874, 2017. 8
- [41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. 11
- [42] K. van Deemter. *Computational models of referring: a study in cognitive science*. MIT Press, 2016. 8
- [43] R. Vedanta, S. Bengio, K. Murphy, D. Parikh, and G. Chechik. Context-aware captions from context-agnostic supervision. *CoRR*, abs/1701.02870, 2017. 8
- [44] R. Vedantam, S. Bengio, K. Murphy, D. Parikh, and G. Chechik. Context-aware captions from context-agnostic supervision. In *Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2017. 8
- [45] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2015. 8
- [46] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1989. 6
- [47] L. Wittgenstein. *Philosophical investigations*. Macmillan, 1953. 8
- [48] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. S. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. 6
- [49] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2016. 8
- [50] L. Yi, H. Su, X. Guo, and L. J. Guibas. Syncspeccnn: Synchronized spectral CNN for 3d shape segmentation. *CoRR*, abs/1612.00606, 2016. 5
- [51] L. Yu, Z. Lin, X. Shen, Y. Jimei, X. Lu, M. Bansal, and L. T. Berg. Mattnet: Modular attention network for referring expression comprehension. *CVPR*, 2018. 2, 8
- [52] L. Yu, P. Poirson, S. Yang, C. A. Berg, and L. T. Berg. Modeling context in referring expressions. *ECCV*, 2016. 2, 8
- [53] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speaker-listener-reinforcer model for referring expressions. *CoRR*, abs/1612.09542, 2017. 8
- [54] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *European conference on computer vision*, pages 834–849. Springer, 2014. 8
- [55] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. *CVPR*, 2010. 2

## A. Appendix

### A.1. CiC details

To build the triplets comprising the communication contexts of *CiC*, we exploited the *latent* (bottleneck-derived) vector space of a Point-Cloud based AutoEncoder (PC-AE) [1], trained with chair-only objects of ShapeNet [4]. Concretely, we used a PC-AE with small bottleneck (64D) to promote meaningful euclidean distances and after embedding all  $\sim 7000$  ShapeNet chairs in the resulting space, we computed their underlying 2-(euclidean)-nearest-neighbor graph. On this graph, we selected the 1K chairs with the highest in-degree to ‘seed’ the triplet generation. For each of the 1K (seed) chairs, we considered it together with its two nearest neighbors from the *entire* shape collection to form a *Hard* triplet. Also, we considered it together with the two chairs that were closest to it but which were also more distant from it than the median of all pairwise distances, to form an *Easy* triplet. The above procedure gives rise to 2000 communication contexts when target vs. distractor information is ignored. However, to counterbalance the dataset while annotating these contexts in AMT, we ensured that *each* chair of a context was considered as a distractor and as a target, and that each resulting combination was annotated by at least 4 humans. Last, we note that when building the Hard triplets, we applied a manually tuned distance-threshold, to reject triplets that contained objects that were ‘too’ close: we found that about  $\sim 3\%$  of chairs had a geometric duplicate that could vary only wrt. its texture.

### A.2. Image and point-cloud pre-training

For the listeners and speakers we trained a PC-AE under the Chamfer loss [1] with a 128D bottleneck and point clouds with 2048 points extracted from 3D CAD models, uniformly area-wise. We also fine-tuned a VGG-16 pre-trained on ImageNet on a 8-way classification, with 36,632 rendered images of textureless 3D CAD models, taken from a single view-point. Concretely, we used images of the 8 largest object classes of Shape-Net (car, airplane, vessel, sofa, chair, table, lamp, rifle) and a uniformly random i.i.d. split of [90%, 5%, 5%] for train/test/val purposes. We fine-tuned the network for 30 epochs. During the first 15 epochs we optimized *only* the weights of the last (fc8) layer and during the last 15 epochs the weights of all layers. The attained test classification accuracy was 96.9%. Last, to embed an image for the downstream listening/speaking tasks, we used the 4096D output activations of the penultimate (fc7) fully-connected layer.

### A.3. Pre-processing utterances

We preprocessed the collected human utterances by i) lowercasing, ii) tokenizing by splitting off punctuation, iii) tokenizing by splitting superlative or comparative adjectives

ending in -er, -est to their stem word, e.g. ‘thinner:’ → ['thin', 'er'] and, iv) replacing tokens that appear once or not at all in a training split with a special symbol marking an unknown token (<UNK>). Furthermore, we ignored the utterances comprised by more than 33 tokens (99th percentile) and those for which the human listener in the underlying trial did not guess correctly the target. Last, we concatenated listener and speaker utterances from the same trial (in their order of formulation) by adding in the end of each but the last utterance a special symbol marking a dialogue: (<DIA>), e.g. ['the', 'thin', 'chair', <DIA>, 'yes'].

### A.4. Listeners details

For the listeners we used a uni-directional LSTM cell with 100 hidden units, the output of which was passed into a 3-layer MLP with [100, 50, 3] neurons that predicted the triplet’s classification logits. To the output of each hidden layer of the MLP, batch normalization [18] and a ReLU [28] non-linearity was applied. The listeners’ word-embedding was initialized with a 100D GloVe embedding pre-trained on the 6B Wikipedia 2014 corpus, and which was further fine-tuned during training. The PC-AE (128D) and VGG (4096D) latent vectors, that encoded each object, were passed as *input* to the LSTM when only one geometric modality was used. When the two modalities used together, the PC-AE codes were concatenated with the *output* of the LSTM, and the concatenated result was processed by the final MLP. In either case, we first re-embedded these geometric codes (100D) with 2 separate/single FC-ReLU layers (referred as ‘projection’ layers in the Main Paper Section 3). An overview of the proposed listener reflecting the overall design choices is given in Fig.5. We used dropout with 0.5 keep probability *before* the ‘projection’ layers with a drop-out mask that was the same for the objects of a given triplet. Separate dropout with 0.5 keep probability was applied in all input vectors of the LSTM (i.e. on the language tokens or the grounding geometric codes). Last, the ground-truth indicator vectors of each triplet were label-smoothed [41] by assigning 0.933 probability mass to the target and 0.0333 to the distractors (i.e. smoothing of 0.9).

**Discussion** Label smoothing yielded a mild performance boost of  $\sim 2\%$  across all ablated listener architectures, in accordance with previous work [41]. We note that we did not manage to improve the best attained accuracies by applying layer normalization [3] in the LSTM, or adversarial regularization [30] on the word-embedding. Dropout [39] was by far the most effective form of regularization for our listeners ( $\sim[8\text{-}9\%]$ ), following by  $L_2$  weight-regularization of the projection layers ( $\sim[2\text{-}3\%]$ ). Finally, using a separate MLP to process the PC-AE codes, was slightly better than feeding them directly in the LSTM (after the tokens of each utterance were processed). However, grounding the

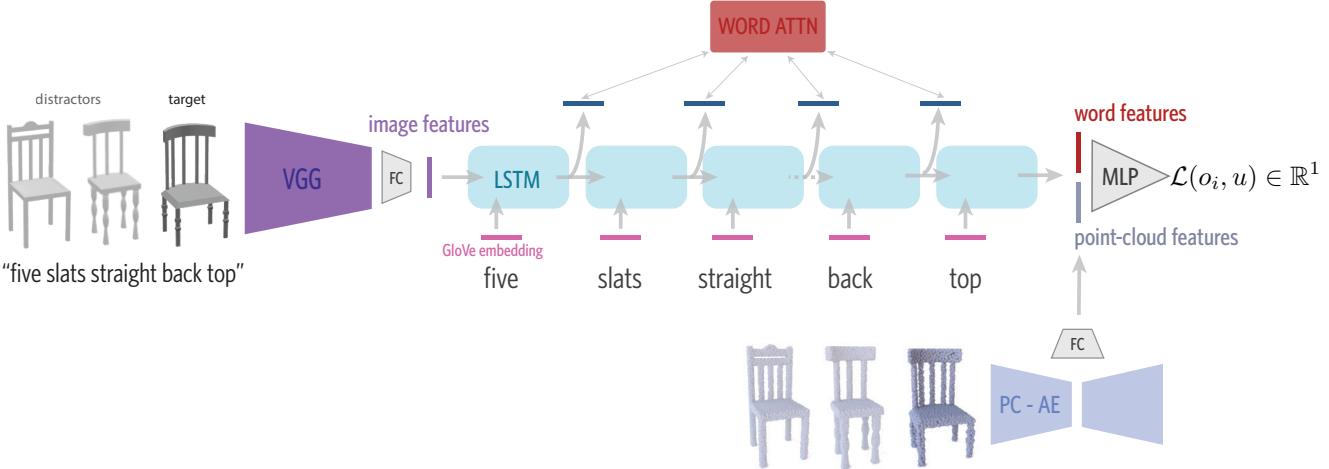


Figure 5: *Baseline* listener architecture combining 2D images, 3D point-clouds and linguistic utterances.

Hyper Parameters \ Architecture	<i>Baseline</i>	<i>Early-Context</i>	<i>Combined-Interpretation</i>
Learning rate	0.0005	0.001	0.001
Label-smoothing	0.9	0.9	0.9
$L_2$ regularization	0.3	0.05	0.09
LSTM-input-dropout	0.5	0.7	0.45

Table 6: Optimal hyper-parameters for ablated neural listener architectures, using both geometric modalities and word-attention and various degrees of context. Dropout numbers reflect the *keep* probability.

LSTM with the PC-AE codes, and using the VGG codes in the end of the pipeline (either via pre-MLP concatenation or by feeding the latter in the LSTM) deteriorate *significantly* all attained results.

**Context Ablations** We ablated three architectures that used simultaneously images and point-clouds, word attention and different degrees of context (See Main Paper Section 3). The optimal Hyper-Parameters (HP) for each architecture are shown in Table 6. We did a grid search over the space of HP associated with each architecture *separately*. To circumvent the exponential growth of this space, we search it into two phases. First, we optimized the learning rate (in the regime of [0.0001, 0.0005, 0.001, 0.002, 0.004, 0.005]) in conjunction with the drop-out (keep probability) applied at the LSTM’s *input*, in the range [0.4-0.7] with increments of 0.05. Given the acquired optimal values, we searched for the optimal  $L_2$  weight-regularization (in the range of [0.005, 0.01, 0.05, 0.1, 0.3, 0.9]) applied at the two projection layers, and label-smoothing ([0.8, 0.9, 1.0]). For these experiments we used a single random seed to control for the data splits with the *object-generalization task*. We note that for the *Early-Context* listener, using a single 1D convolutional layer to extract the grounding vec-

tor of each object, appeared to produce better results than using a single FC layer (or deeper alternatives). This single convolutional layer we used, converted the input signal  $[f(v_j, v_k) || g(v_j, v_k) || v_i] \in \mathbb{R}^{100 \times 3}$  to a  $\mathbb{R}^{100 \times 1}$  LSTM-grounding vector for each object  $v_i$ , with an  $8 \times 3 \times 1$  kernel and stride 1.

**Training** We trained the *Baseline* and the *Combined-Interpretation* for 500 epochs and the *Early-Context* for 350. This was sufficient, as more training increased overfitting without improving the attained test/val accuracies. We halved the learning every 50 epochs, if the validation error was not improved in any of them. Namely, every 5 epochs we evaluated the model on the validation split in order to select the epoch/weights with the best accuracy. Because the *Combined-Interpretation* is sensitive in the input order of the object codes, we randomly permute them during training. We use the ADAM [21] ( $\beta_1 = 0.9$ ) optimizer for all experiments.

## A.5. Speaker details

**Image-based speaker** To find good model parameters for an image-based speaker, we considered a hyper-parameter search on a *literal* variant. Similarly, to what we did in the

ablations of listener variants we conducted a two-stage grid search given a single random seed and the *object generalization* task. At the first stage, we searched models varying: a) the hidden neurons of the LSTM (100 or 200), b) the initial learning rate ([0.0005, 0.001, 0.003]), c) the dropout keep probability applied on the word-embeddings ([0.8, 0.9, 1.0]) and d) the dropout keep probability applied at the LSTM’s output ([0.8, 0.9, 1.0]). The two best performing models were further optimized by considering  $L_2$ -weight regularization applied at the FC-projection layer (with values in [0, 0.005, 0.01]) and the dropout keep-probability applied before the FC-projection layer ([0.5, 0.7, 0.9 1.0]). The resulting optimal parameters are reported in Table 7.

**Point-cloud-based speaker** For the point-based speaker, we did a similar but more constrained hyper-parameter search as we did for the image-based speaker, by also considering its *literal* variant. Here, we fixed the drop-out applied the word-embeddings and to the LSTM’s output (0.8 and 0.9 keep-probability respectively) and ablated the remaining hyper-parameters as we did for the image-based speaker. We found the same configuration of parameters (Table 7) to be optimal for point-based models as well. Exception to this was the the dropout applied to the PC-AE codes before the FC-projection (no dropout at all was best in this case). Also, the point-based speakers needed more training to converge than the image-based ones (maximally 400 epochs vs. 300).

**Model selection** To do model selection for a training speaker, we used a pre-trained listener (with the same train/test/val splits) which evaluated the synthetic utterances produced by the speaker during training. To this purpose the speaker generated 1 utterance for each unique triplet in the validation set via greedy (arg-max) sampling every 10 epochs of training and the listener reported the accuracy of predicting the target given the synthetic utterance. In the end of training (300 epochs for image-based speakers vs. 400 for point-based ones), the epoch/model with the highest accuracy was selected.

**Other details** We initially used GloVe to provide our speakers pre-trained word embeddings, as in the listener, but found that it was sufficient to train the word embedding from uniformly random initialized weights (we used the range [-0.1, 0.1]). We also initialized the bias terms of the linear word-encoding layer with the log probability of the frequency of each word in the training data [19], which provided faster convergence. We train with SGD and ADAM ( $\beta_1 = 0.9$ ) and apply norm-wise gradient clipping with a cut-off threshold of 5.0. The training utterances have a maximal length of 33 tokens (99th percentile of the dataset). For

any speaker we sampled utterances of the maximum training length. For the *pragmatic* speaker we sample and score 50 utterances per triplet at test time (following Eq. 1 of Main Paper).

**Point-cloud & image-based speaker** In *preliminary* experiments, we attempted to incorporate both geometric modalities: point-clouds and images in a speaker network, similarly to what we did for the best-performing listener. While, this resulted in a (*literal*) speaker model that could achieve higher neural-listener evaluation-accuracy than when either modality was used in isolation, we did not observe any improvement against the image-based speaker in AMT *human*-listener experiments.

We attempted three ways of ‘mixing’ the two modalities in a speaker. Namely, for each object of a communication context: a) providing the LSTM with the *concatenation* of its projected VGG code and its projected PC-AE code, b) same as a) but instead of concatenation, using the *sum* operator, c) first providing its PC-AE projected code followed at the *next time* step by its VGG one. We compared these approaches by using the optimal hyper-parameters for an image-based speaker and only vary the amount of dropout applied to the point-cloud before the projection layer ([1.0 0.8, 0.6] keep probability). In all cases, avoiding dropout was best. The final results for a single random-seed and the object-generalization task are reported in Table 8. We note that while the optimal speaker that used two modalities performed slightly better than the image-based speaker, per neural-listener evaluation, it did not improve the attained performance in preliminary experiments with of human listeners in AMT.

## A.6. Further quantitative results

### A.6.1 Listeners: context incorporation

In Table 9 we complement the results presented in the Main Paper at Table 1, by including two more sub-populations (‘Negative’ and ‘Split’). In Table 10, we repeat this study for listeners trained and tested on the *language generalization* task. ‘Negative’ is a subpopulation of utterances that contain at least one word of negative content e.g. ‘not’, ‘but’ etc. and is comprised by  $\sim 15.0\%$  of all test utterances. ‘Split’ is smaller subpopulation ( $\sim 3.2\%$  of test data) that includes language the explicitly contrasts the target with the distractors e.g. ‘from the two that have thin legs, the one...’. We used an ad hoc set of search queries to find such utterances among the test set and found that the *Early-Context* architecture does perform noticeably better on these utterances. However, given the low occurrence of such cases, the resulting effects were not significant and we decided the gains of *Early-Context* architecture were not worth the increase in model complexity and rigidity with respect to con-

LSTM Size	Learning rate	$L_2$ -reg.	Word-Dropout	Image-Dropout	LSTM-out Dropout
200	0.003	0.005	0.8	0.5	0.9

Table 7: Optimal hyper parameters for *literal* image-based neural-speaker. The dropout numbers reflect keep probabilities and the Image-Dropout refers to the dropout applied at the VGG-image codes, before the FC-projection layer.

Approach	Listener's Accuracy
Concat (100D)	$65.1 \pm 0.51\%$
Concat (200D)	$78.2 \pm 0.95\%$
Sum	$77.9 \pm 0.38\%$
Serial	<b><math>79.0 \pm 0.32\%</math></b>

Table 8: Ablating approaches for incorporating simultaneously point-clouds with images in a *literal* neural-speakers. *Sum*: Summing the two latent codes for each object. *Concat*: Concatenating the codes. *Serial*: Feeding them one after the other in the LSTM. Concatenation naturally doubles the input-dimensions of the LSTM (Concat 200D). To keep them the same as with all other experiments (100D) we also tested reducing the VGG/PC-AE projection layers to 50 dimensions for each modality (Concat 100D). Results are averages of 5 samples of utterances for a fixed test dataset.

text size.

### A.6.2 Listeners: part-lesion

We complement Table 3 of the Main Paper, with a similar study (Table 11) where we ablate our neural listeners with regards to their sensitivity in referential utterances based on object parts, when *both* geometric modalities are used. We have observed that the PC-AE attempts to reconstruct (de-code) noisy but complete models, even when the input is a partial, which could explain the gains seen in Table 11 compared to Table 3 when lesioning parts.

### A.6.3 Speakers: length penalty and listener awareness

To find the optimal length-penalty value ( $\alpha$ , Main Paper Eq.1) for image-based *literal* and a *context-unaware* speaker variants, we used our best-performing listener to simultaneously score and evaluate the utterances produced by the speakers for different values of  $\alpha$  (Fig. 6a). The best performing length penalty for a context-unaware speaker is 0.7, and for a literal 0.6. Given the optimal  $\alpha$  values, for these models we show the effect of using different degrees of listener-awareness ( $\beta$ ) in Fig. 6b. It is interesting to observe that even the context-unaware speaker can generate utterances that an evaluating listener can find them very discriminative, as long as it allows to rank them.

In Fig. 7 we demonstrate the effect that the relative (training) size of the evaluating listener vs. the 'internal'

listener used by a *pragmatic* speaker has for the evaluating accuracy, for two values of  $\beta$ . In either case we observe a slow decline in evaluating accuracy as the training size for the evaluating listener increases (from 0.5 to 0.9) and consequently the training size for the 'internal' listener decreases (from 0.5 to 0.1).

### A.6.4 Understanding out-of-class reference

We complement the Table 5 with the standard-deviations of the underlying accuracies in Table 13. We also report simple statistics regarding the underlying transfer classes in Table 12. We note that the transfer learning accuracies acquired by listeners operating with both point-clouds and images for these experiments were significantly lower ( $\sim 7\%$  on average). We hypothesize that this is due to the fact that our (chair-trained) listener models that utilize point-clouds, rely on a pre-trained *single*-class PC-AE, unlike the pre-trained VGG (image encoder) which was fine-tuned with multiple ShapeNet classes. Also, for these experiments, [ $\sim 1\% \sim 7\%$ ] (depending on the transfer class) of the tokens were not in the chair-vocabulary, and we chose to ignore them i.e. treat them as white-space. Last, per Table 12 in *all* transfer classes the *with-part* population contains quite larger utterances than the *without-part* (9.3 vs. 5.5 on average) and that even in the case of lamps, arguably the most dissimilar category from chairs,  $20 + 37 = 57\%$  of the collected utterances are in the *known* population.

Architecture	Overall	Subpopulations				
		Hard	Easy	Sup-Comp	Negative	Split
<i>Combined-Interpretation</i>	$75.9 \pm 0.5\%$	$67.4 \pm 1.0\%$	$83.8 \pm 0.6\%$	$74.4 \pm 1.5\%$	$77.3 \pm 1.5\%$	$65.8 \pm 5.2\%$
<i>Early-Context</i>	$79.4 \pm 0.8\%$	<b><math>70.1 \pm 1.3\%</math></b>	$88.1 \pm 0.6\%$	$75.6 \pm 2.2\%$	<b><math>78.9 \pm 1.4\%</math></b>	<b><math>67.4 \pm 3.6\%</math></b>
<i>Baseline</i>	<b><math>79.6 \pm 0.8\%</math></b>	$69.9 \pm 1.3\%$	<b><math>88.8 \pm 0.4\%</math></b>	<b><math>76.3 \pm 1.3\%</math></b>	$77.5 \pm 1.2\%$	$62.5 \pm 3.7\%$

Table 9: Comparing the effect of context inspection for listening on various (test) subpopulations of the *object generalization* task. The listeners use images, point-clouds and word-attention. Reporting averages of five random seeds controlling the split populations and the network’s initialization.

Architecture	Overall	Subpopulations				
		Hard	Easy	Sup-Comp	Negative	Split
<i>Combined-Interpretation</i>	$78.4 \pm 0.2\%$	$71.5 \pm 0.6\%$	$85.2 \pm 0.3\%$	$75.8 \pm 0.9\%$	$77.6 \pm 0.8\%$	$61.8 \pm 3.0\%$
<i>Early-Context</i>	<b><math>84.4 \pm 0.5\%</math></b>	<b><math>78.5 \pm 0.8\%</math></b>	$90.2 \pm 0.7\%$	<b><math>80.9 \pm 0.6\%</math></b>	<b><math>82.6 \pm 1.1\%</math></b>	<b><math>68.9 \pm 2.3\%</math></b>
<i>Baseline</i>	$83.7 \pm 0.2\%$	$77.0 \pm 0.8\%$	<b><math>90.3 \pm 0.3\%</math></b>	$80.8 \pm 0.8\%$	$80.5 \pm 1.0\%$	$64.6 \pm 3.7\%$

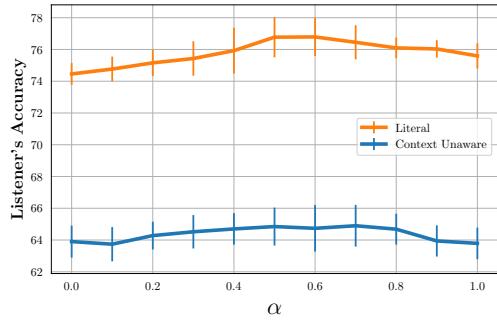
Table 10: Comparing the effect of context inspection for listening on various (test) subpopulations of the *language generalization* task. The listeners use images, point-clouds and word-attention. Reporting averages of five random seeds controlling the split populations and the network’s initialization.

	Single Part Lesioned	Single Part Present
<b>Mentioned Part</b>	$44.9\% \pm 1.2$	$67.2\% \pm 1.1$
<b>Random Part</b>	$68.9\% \pm 1.3$	$42.3\% \pm 1.3$

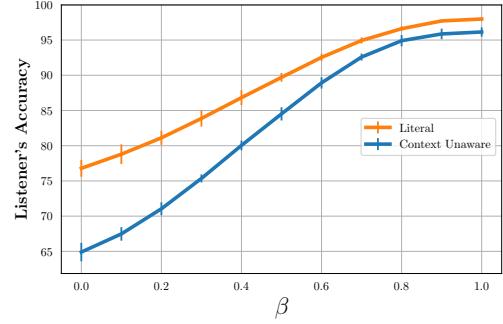
Table 11: Evaluating the part-awareness of neural listeners by lesioning object *parts*. Results shown are for listeners using **both** point-clouds and images, with average accuracy of 78.8% when *intact* objects are used.

Class	Population		
	entire	with part	without part
chair	7.1	8.0 (77%)	4.7 (21%)
bed	6.4	7.0 (26%)	5.3 (48%)
lamp	7.3	11.0 (20%)	5.9 (37%)
sofa	10.1	11.0 (72%)	5.9 (15%)
table	6.6	8.0 (40%)	4.9 (42%)
average	7.6	9.3 (39.5%)	5.5 (35.5%)

Table 12: Average length of utterances for various transfer classes (complementing Table 5, Main Paper). Between parentheses is reported the percentage of the entire population that is captured by its specific sub-population. The average (last row) is wrt. the transfer classes only; the chair-category is displayed for reference.

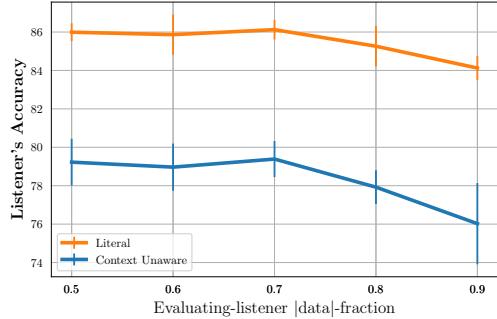


(a) Effect of the length-penalty.

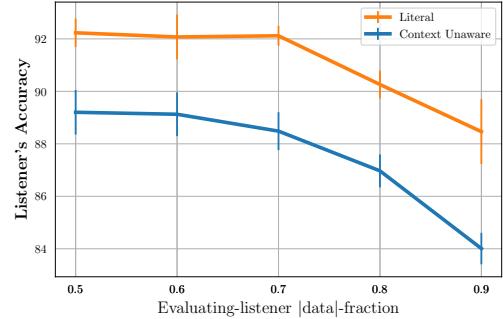


(b) Effect of increasing listener's opinion ( $\beta$ ).

Figure 6: Left: Measuring the effect of using different length-penalty ( $\alpha$ ) values to select the top-1 scoring utterance for context-unaware and pragmatic speakers for contexts of the *object generalization* validation split (left). Right, measuring the the effect of various  $\beta$ -values used in turning the context-unaware and literal speakers ( $\beta = 0.0$ ) to *pragmatic* speakers, under the optimal  $\alpha$  of the left figure. In both plots, the y-axis reflects the performance of a listener who is used to rank *and* evaluate the utterances. Averages are with respect to 5 random seeds controlling the data splits and the initializations of the neural-networks.



(a) Speakers using a modest  $\beta = 0.5$  value.



(b) Speakers using the most aggressive  $\beta = 1.0$  value.

Figure 7: Effect of partitioning the training data for the evaluating and ‘internal’ listeners. Here, we turn context-unaware and literal speakers into pragmatic ones under two  $\beta$  values. The x-axis shows the fraction ( $f$ ) of the training data that was used to train the *evaluating* listener (the remaining  $100 - f\%$  is used to train the *internal* listener) of the resulting pragmatic speaker. On the y-axis we display the performance of the evaluating listener for the top-scoring model-generated utterance.

Class Population \	bed	chair	lamp	sofa	table
entire	$56.4 \pm 2.0\%$	$77.4 \pm 0.9\%$	$50.1 \pm 1.3\%$	$53.6 \pm 2.0\%$	$63.7 \pm 1.2\%$
known	$55.8 \pm 1.5\%$	$77.8 \pm 0.8\%$	$51.9 \pm 1.8\%$	$55.0 \pm 2.0\%$	$65.5 \pm 0.9\%$
with part	<b><math>63.8 \pm 4.2\%</math></b>	$77.0 \pm 0.8\%$	<b><math>60.3 \pm 4.4\%</math></b>	<b><math>55.1 \pm 2.5\%</math></b>	<b><math>68.3 \pm 2.6\%</math></b>
without part	$51.5 \pm 3.0\%$	<b><math>80.5 \pm 1.2\%</math></b>	$47.1 \pm 2.8\%$	$54.7 \pm 5.5\%$	$62.7 \pm 0.9\%$

Table 13: Transfer-learning of neural listeners in novel object *classes*: average accuracies *with* standard deviations (complementing Table 13, Main Paper). The sub-populations denote *entire*: all collected utterances, *known*: utterances containing *only* chair-training-vocabulary words, *with-part*: subset of *known*, with utterances containing at least one part-related word, *without-part* subset of *known* and complement of *with-part*. For reference the test-chair statistics are shown (first row) but not included in the reported average (last row).

Figure 8: **Examples of attention weights on human utterances.** The listener’s LSTM appears to learn attention weights that emphasize the more informative words disambiguating the referent. For these results the *Baseline* listener is used and the attention-scores are extracted when the target object is grounding the LSTM.

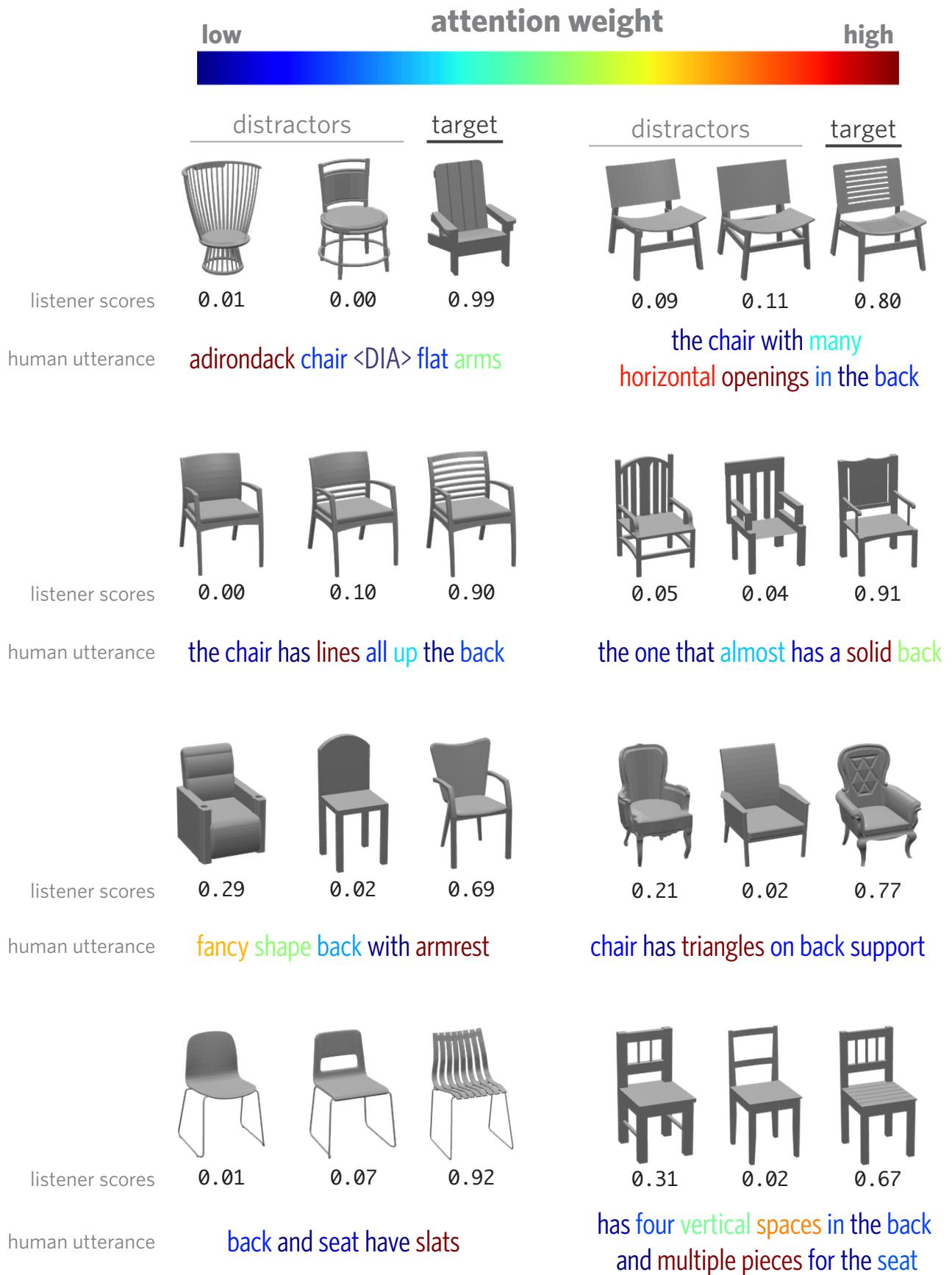


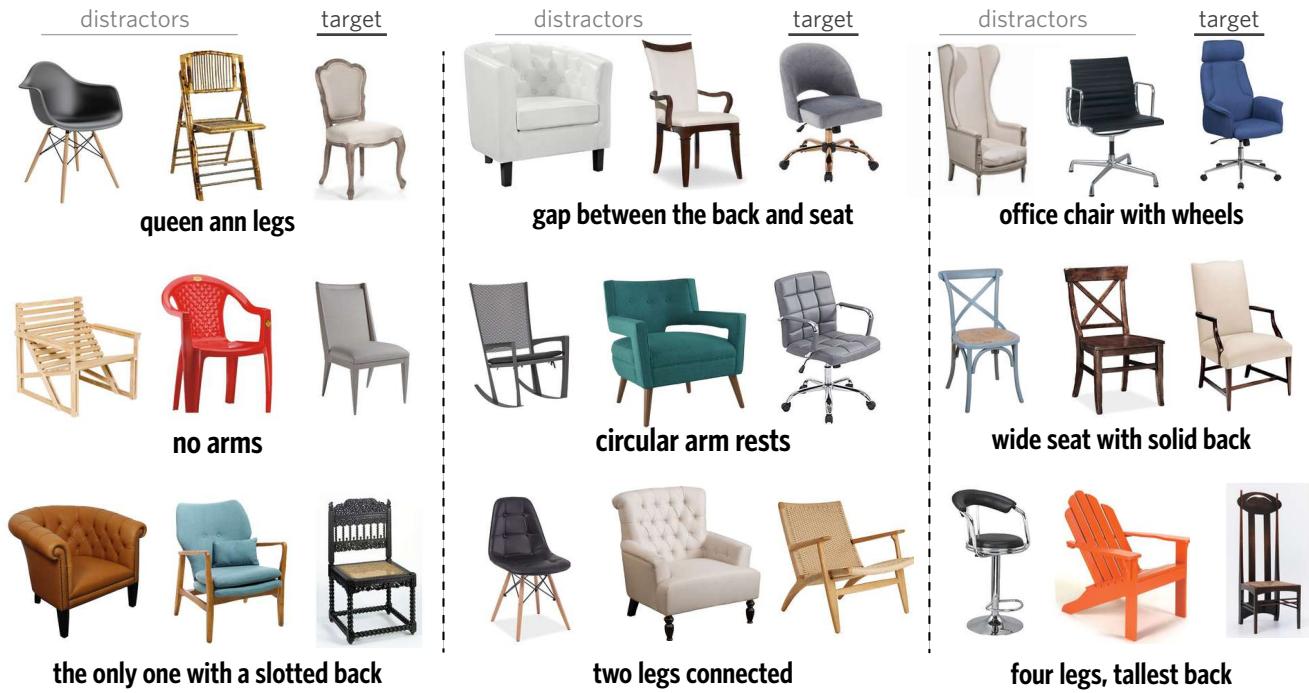
Figure 9: **Examples of lesioning all but the mentioned part.** Here, we show the response of a *Baseline* listener tested with visual representations of entire objects (left column, three chairs) vs. its response when it receives **only** the visual features corresponding to the referred semantic-part (right column). The corresponding utterance is shown left-most of each row. In these examples the listener assigns higher confidence to the actual target when the isolated parts are considered instead of the entire objects, implying that further performance gains can occur with an explicit part-aware visual attention mechanism.

Human Utterance	distractors			target	distractors			target
solid, square <b>backing</b> <DIA> hole in back? <DIA> no								
	listener scores	0.48	0.01	0.51	0.32	0.08	0.60	
sleek rounded <b>arms</b> , expensive								
	listener scores	0.30	0.11	0.59	0.14	0.05	0.81	
the <b>seat</b> of the chair has a curve								
	listener scores	0.04	0.84	0.12	0.07	0.30	0.63	
the one with the fattest <b>legs</b>								
	listener scores	0.38	0.43	0.19	0.07	0.13	0.80	

Figure 10: **Pragmatic vs. literal speakers for two modalities.** More examples of pragmatic vs. literal generations in Hard contexts. Top-row includes examples from image-based speakers. Bottom-row from point-based ones.

	distractors	target	distractors	target	distractors	target
<b>image-based speakers</b>						
<b>pragmatic speaker</b>	<b>square arms</b>	<b>knobby legs</b>	<b>no arm rests</b>			
literal speaker	with the tall-est back and seat	the one with the thick-est legs	the one with high-est back			
<b>point-cloud based speakers</b>						
<b>pragmatic speaker</b>	<b>most square back</b>	<b>thick-est legs</b>	<b>tall-est back</b>			
literal speaker	thin-est seat	square rack at bottom of chair	has arms			

(a) **Model generations with real images.** The top-scoring utterance of a pragmatic model is displayed under each context.



(b) **Human-utterance comprehension with unseen object classes.** The human utterance is color-coded according to the attention placed by a chair-trained listener who also evaluates the object-utterance compatibility (scores shown under its context).

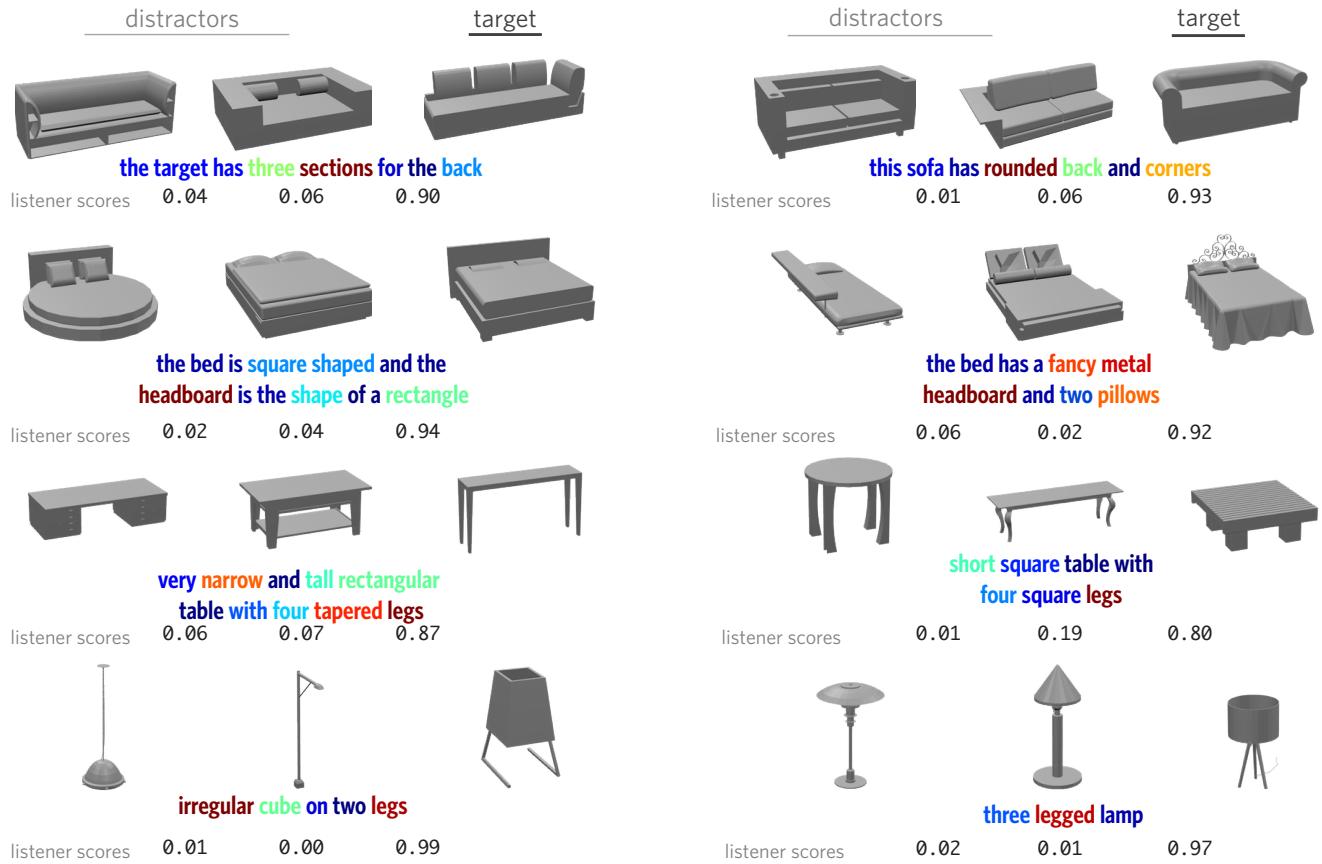
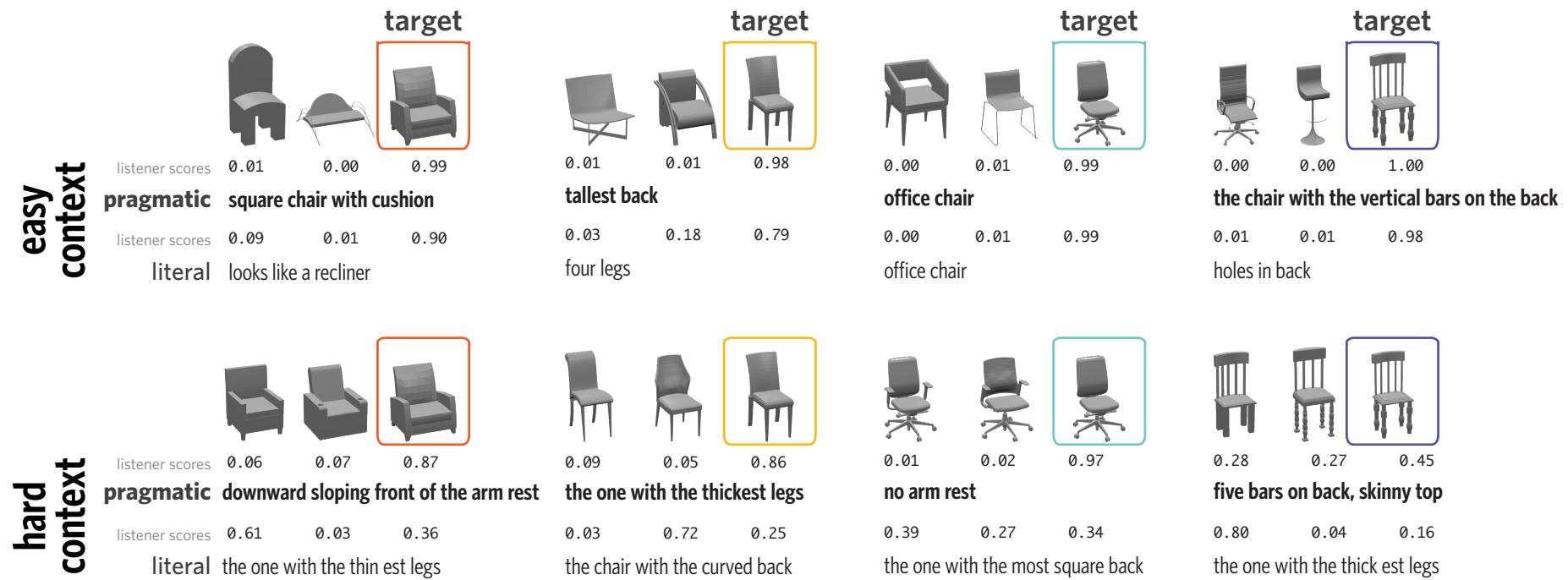
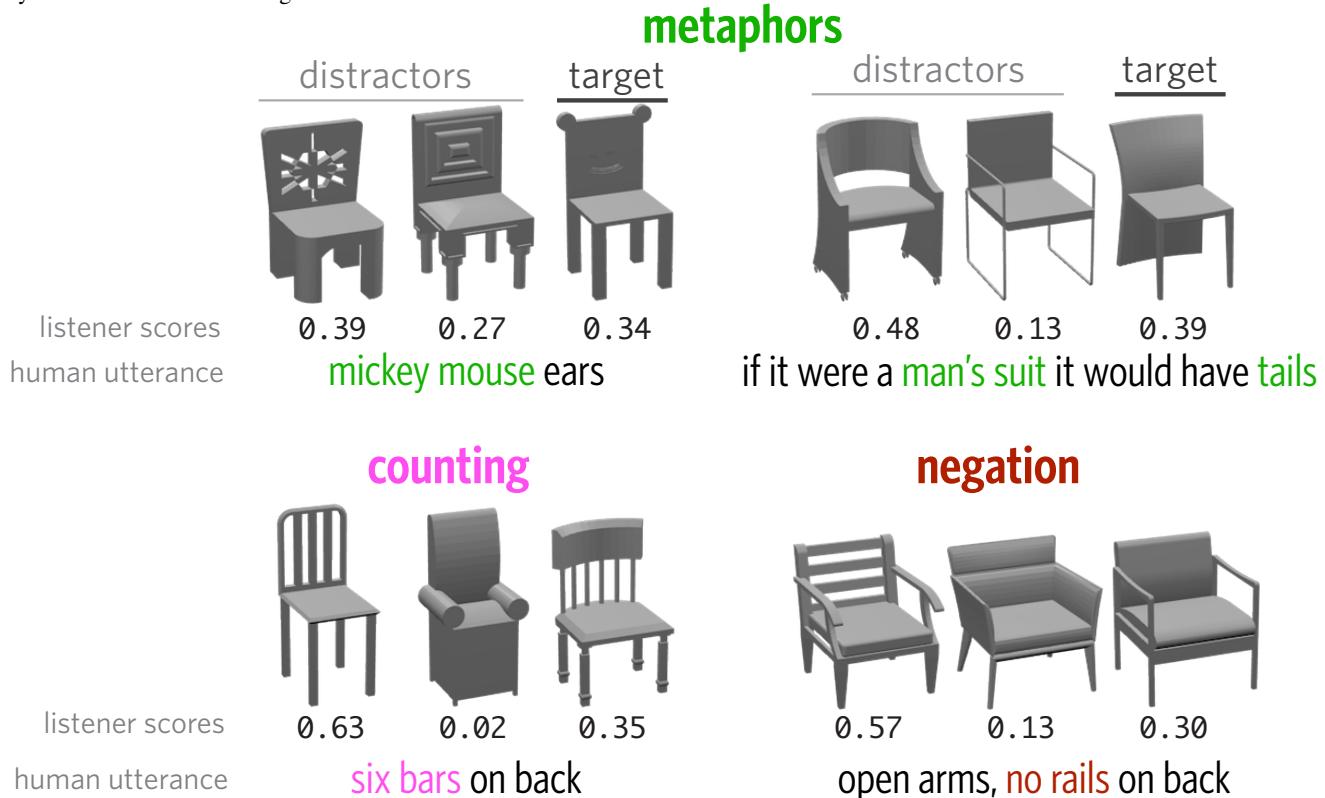


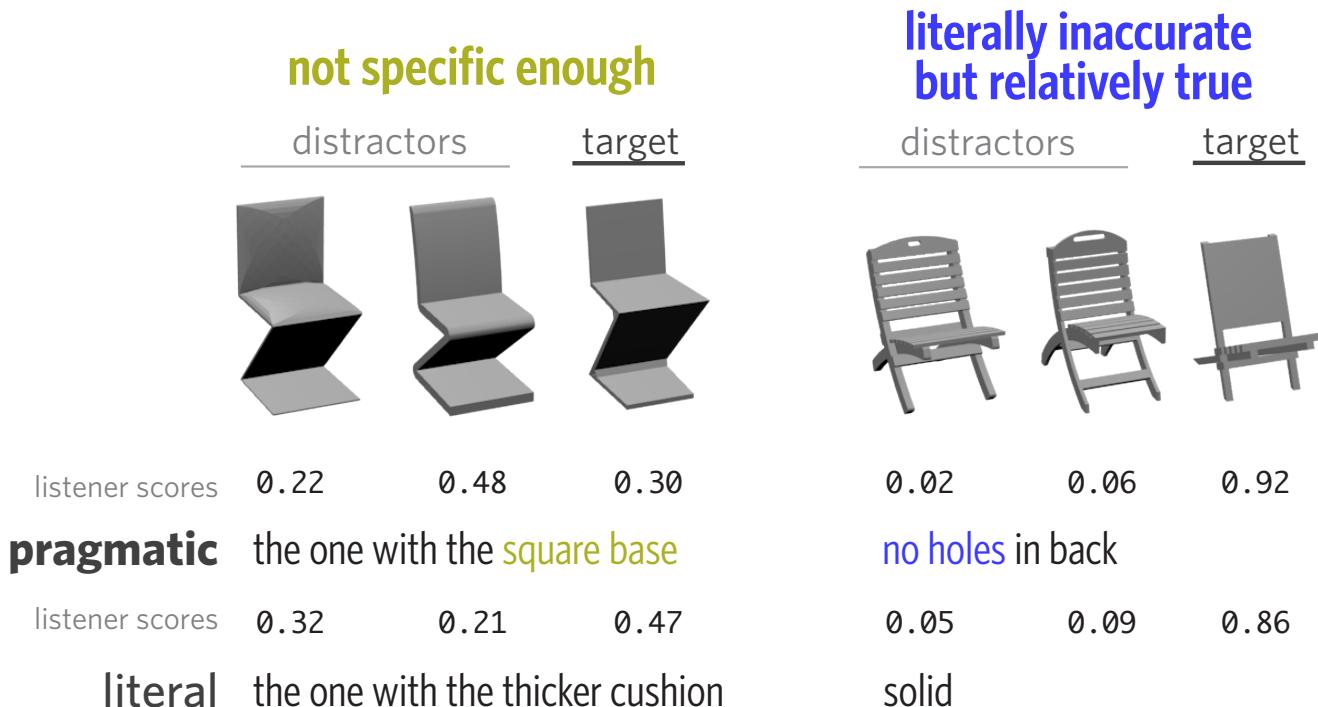
Figure 12: **Effect of context on production:** Synthetic utterances generated by a *literal* and *pragmatic* image-based speaker. The top and bottom rows show utterances produced for the same target in a Easy and Hard context, respectively. The *Baseline* (with point-clouds and images and attention) listener is used to predict the target and its confidence is displayed above each utterance. While both speaker models produce similarly effective utterances in Easy contexts, the literal speaker fails to produce effective utterances in Hard contexts.



(a) **Neural-listener failure cases.** Our top-performing listener model appears to struggle to interpret referential language that relies on metaphors, precisely counting parts, or (to a less degree) negations. All examples are drawn from the test set and were correctly classified by human listeners in the original task.



(b) **Neural-speaker failure cases.** Sometimes even the *pragmatic* speaker produces insufficiently specific utterances that mention only undiagnostic features, or produces utterances that are literally false of the target (e.g. there technically *is* a hole in the back) while still succeeding in distinguishing the objects.



## A.7. Miscellaneous

	<b>word</b>	office	sofa	regular	folding	wooden	stool	wheels	metal	normal	rocking
<b>Easy</b>	<b>pmi</b>	-1.70	-0.94	-0.88	-0.84	-0.83	-0.79	-0.78	-0.71	-0.67	-0.66
<b>Hard</b>	<b>word</b>	alike	identical	thickness	texture	darker	skinnier	thicker	perfect	similar	larger
	<b>pmi</b>	0.69	0.67	0.67	0.66	0.65	0.64	0.63	0.62	0.62	0.61

Table 14: Most distinctive words in each context type according to point-wise mutual information (excluding tokens that appeared fewer than 30 times in the dataset). Lower numbers are more distinctive of Easy and higher numbers are more distinctive of Hard.

Each game consisted of 69 trials (unique triplets) and participants swapped speaker and listener roles with the conclusion of each trial. The game’s interface is depicted in Figure 14. Participants were allowed to play multiple games, but most participants in our dataset played exactly one game (81% of participants). The most distinctive words in each triplet type (as measured by point-wise mutual information) are shown in Table 14.

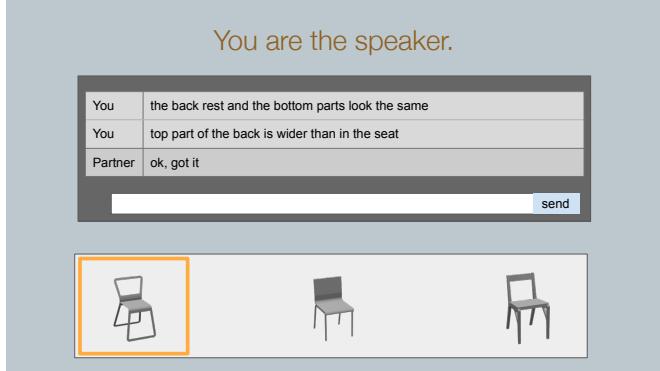


Figure 14: Reference game interface. Communication was natural without any system constraints being imposed.

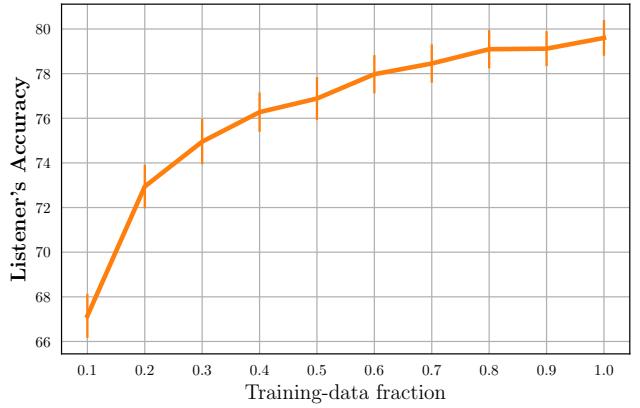


Figure 15: Listener’s accuracy for different sizes of training data, under the *object* generalization task. The original split includes [80%, 10%, 10%] for training/test/val purposes, thus the maximum size of training data is 0.8 of the entire dataset corresponding to the value (fraction) 1.0 in the x-axis. The listener model uses the *Baseline* architecture with word attention, images and point-clouds and its accuracy is measured on the original (10%) test split. Results are averages of 5 random seeds controlling the original data split and the neural-net’s initialization.