Embodied Language Grounding with 3D Visual Feature Representations

Mihir Prabhudesai*, Hsiao-Yu Fish Tung*, Syed Ashar Javed*,
Maximilian Sieb†, Adam W. Harley, Katerina Fragkiadaki
{mprabhud,htung,sajaved,msieb,aharley,katef}@cs.cmu.edu
Carnegie Mellon University

Abstract

We propose associating language utterances to 3D visual abstractions of the scene they describe. The 3D visual abstractions are encoded as 3-dimensional visual feature maps. We infer these 3D visual scene feature maps from RGB images of the scene via view prediction: when the generated 3D scene feature map is neurally projected from a camera viewpoint, it should match the corresponding RGB image. We present generative models that condition on the dependency tree of an utterance and generate a corresponding visual 3D feature map as well as reason about its plausibility, and detector models that condition on both the dependency tree of an utterance and a related image and localize the object referents in the 3D feature map inferred from the image. Our model outperforms models of language and vision that associate language with 2D CNN activations or 2D images by a large margin in a variety of tasks, such as, classifying plausibility of utterances, detecting referential expressions, and supplying rewards for trajectory optimization of object placement policies from language instructions. We perform numerous ablations and show the improved performance of our detectors is due to its better generalization across camera viewpoints and lack of object interferences in the inferred 3D feature space, and the improved performance of our generators is due to their ability to spatially reason about objects and their configurations in 3D when mapping from language to scenes.

1. Introduction

Consider the utterance *"the tomato is to the left of the pot"*. Humans can answer numerous questions about the situation described such as, *"is the pot larger than the tomato?"*, *"can we move to a viewpoint from which the tomato is completely hidden behind the pot?"*, *"can we have an object that is both to the left of the tomato and to the right of the pot?"*, and so on. How can we learn computational models that would permit a machine to carry out similar types of reasoning? One possibility is to treat the task as text comprehension (37; 12; 15; 8) and train machine learning models using supervision from utterances accompanied with question answer pairs. However, information needed for answering the questions is not contained in the utterance itself; training a model to carry out predictions in absence of the relevant information would lead to overfitting. Associating utterances with RGB images that depict the scene described in the utterance, and using both images and utterances for answering questions, provides more world context and has been shown to be helpful. Consider though that information about object size, object extent, occlusion relationships, free space and so on, are only indirectly present in an RGB image, while they are readily available given a 3D representation of the scene the image depicts. Though it would take many training examples to learn whether a spoon can be placed in between the tomato and the pot on the table, in 3D this experiment can be imagined easily, simply by considering whether the 3D model of the spoon can fit in the free space between the tomato and the pot. Humans are experts in inverting camera projection and inferring an approximate 3D scene given an RGB image (21). This paper builds upon inverse graphics neural architectures for providing the 3D visual representations to associate language, with the hope to inject spatial reasoning capabilities into architectures for language understanding.

We propose associating language utterances to space-aware 3D visual feature representations of the scene they describe. We infer such 3D scene representations from RGB images of the scene. Though inferring 3D scene representations from RGB images, a.k.a. inverse graphics, is known to be a difficult problem (17; 28; 33), we build upon recent advances in computer vision (34) that consider inferring from images a learnable 3D scene feature representation in place of explicit 3D representations such as meshes, pointclouds or binary voxel occupancies

*Equal contribution
†Work done while in Carnegie Mellon University.

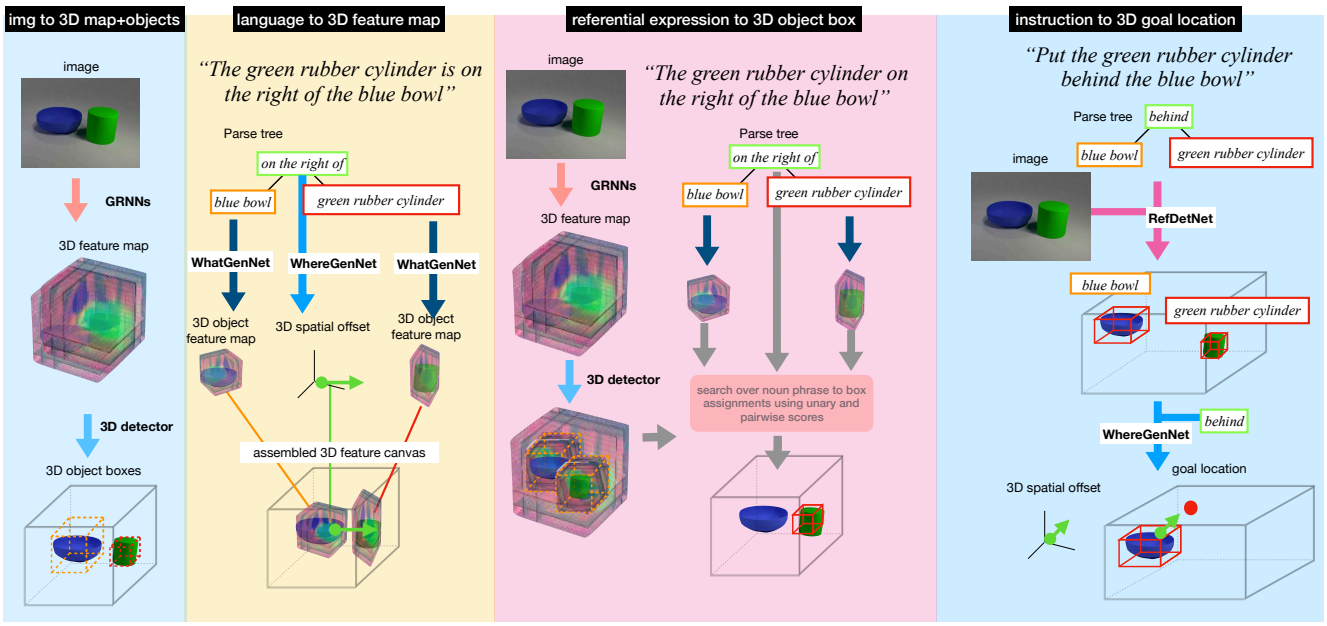

Figure 1: **Embodied language grounding with 3D visual feature representations.** Our model associates utterances with 3D scene feature representations. We map RGB images to 3D scene feature representations and 3D object boxes of the objects present, building upon the method of Tung et al. (34) (column 1). We map an utterance and its dependency tree to object-centric 3D feature maps and cross-object relative 3D offsets using stochastic generative networks (column 2). We map a referential expression to the 3D box of the object referent (column 3). Last, given a placement instruction, we localize the referents in 3D in the scene and infer the desired 3D location for the object to be manipulated (column 4). We use the predicted location to supply rewards for trajectory optimization of placement policies.

pursued in previous inverse graphics research (17; 28; 33). These learnable 3D scene feature maps emerge in a self-supervised manner by optimizing for view prediction in neural architectures with geometry-aware 3D representation bottlenecks (34). After training, these architectures learn to map RGB video streams or single RGB images to complete 3D feature maps of the scene they depict, inpainting details that were occluded or missing from the 2D image input. **The contribution of our work is to use such 3D feature representations for language understanding and spatial reasoning.** We train modular generative networks that condition on the dependency tree of the utterance and predict a 3D feature map of the scene the utterance describes. They do so by predicting the appearance and relative 3D location of objects, and updating a 3D feature workspace, as shown in Figure 1, 2nd column. We further train modular discriminative networks that condition on a referential expression and detect the object being referred to, by scoring object appearances and cross-object spatial arrangements, respectively, as shown in Figure 1, 3rd column. We call our model *embodied* since training the 2D image to 3D feature mapping requires self-supervision by a mobile agent that moves around in the 3D world and collects posed images.

We demonstrate the benefits of associating language to 3D visual feature scene representations in three basic language understanding tasks:

(1) Affordability reasoning. Our model can classify affordable (plausible) and unaffordable (implausible) spatial expressions. For example, *"A to the left of B, B to the left of C, C to the right of A"* describes a plausible configuration, while *"A to the left of B, B to the left of C, C to the left of A"* describes a non-plausible scene configuration, where A, B, C any object mentions. Our model reasons about plausibility of object arrangements in the inferred 3D feature map, where free space and object 3D intersection can easily be learned/evaluated, as opposed to 2D image space.

(2) Referential expression detection. Given a referential spatial expression, e.g., *"the blue sphere behind the yellow cube"*, and an RGB image, our model outputs the 3D object bounding box of the referent in the inferred 3D feature map, as shown in Figure 1 3rd column. Our 3D referential detector generalizes across camera viewpoints better than existing state-of-the-art 2D referential detectors (13) thanks to the view invariant 3D feature representation.

(3) Instruction following. Given an object placement instruction, e.g., *"put the cube behind the book"*, our referential 3D object detector identifies the object to be manipulated, and our generative network predicts its desired 3D

goal location, as shown in Figure 1 4th column. We use the predicted 3D goal location in trajectory optimization of object placement policies. We empirically show that our model successfully executes natural language instructions.

In each task we compare against existing state-of-the-art models: the language-to-image generation model of Deng et al. (6) and the 2D referential object detector of Hu et al. (13), which we adapt to have same input as our model. Our model outperforms the baselines by a large margin in each of the three tasks. We further show strong generalization of natural language learned concepts from the simulation to the real world, thanks to the what-where decomposition employed in our generative and detection networks, where spatial expression detectors only use 3D spatial information, as opposed to object appearance and generalize to drastically different looking scenes without any further annotations. Our model's improved performance is attributed to i) its improved generalization across camera placements thanks to the viewpoint invariant 3D feature representations, and ii) its improved performance on free-space inference and plausible object placement in 3D over 2D. Many physical properties can be trivially evaluated in 3D while they need to be learned through a large number of training examples in 2D, with questionable generalization across viewpoints. 3D object intersection is one such property, which is useful for reasoning about plausible object arrangements.

2. Related Work

Learning and representing common sense world knowledge for language understanding is a major open research question. Researchers have considered grounding natural language on visual cues as a means of injecting visual common sense to natural language understanding (27; 10; 27; 10; 2; 7; 1; 26; 25; 24; 16; 38; 9; 6). For example visual question answering is a task that has attracted a lot of attention and whose performance has been steadily improving over the years (29). Yet, there is vast knowledge regarding basic physics and mechanics that current vision and language models miss, as explained in Vedantam et al. (35). For example, existing models cannot infer whether *"the mug inside the pen"* or *"the pen inside the mug"* is more plausible, whether *"A in front of B, B in front of C, C in front of A"* is realisable, whether the mug continues to exist if the camera changes viewpoint, and so on. It is further unclear what supervision is necessary for such reasoning ability to emerge in current model architectures.

3. Language grounding on 3D visual feature representations

We consider a dataset of 3D static scenes annotated with corresponding language descriptions and their dependency trees, as well as a reference camera viewpoint. We further assume access at training time to 3D object bounding boxes and correspondences between 3D object boxes and noun phrases in the language dependency trees. The language utterances we use describe object spatial arrangements and are programmatically generated, similar to their dependency trees, using the method described in Johnson et al. (14). We infer 3D feature maps of the world scenes from RGB images using Geometry-aware Recurrent Neural Nets (GRNNs) of Tung et al. (34), which we describe for completeness in Section 3.1. GRNNs learn to map 2D image streams to 3D visual feature maps while optimizing for view prediction, without any language supervision. In Section 3.2, we describe our proposed generative networks that condition on the dependency tree of a language utterance and generate an object-factorized 3D feature map of the scene the utterance depicts. In Section 3.3, we describe discriminative networks that condition on the dependency tree of a language utterance and the inferred 3D feature map from the RGB image and localize the object being referred to in 3D. In Section 3.4, we show how our generative and discriminative networks of Sections 3.2 and 3.3 can be used to follow object placement instructions.

3.1. Inverse graphics with Geometry-aware Recurrent Neural Nets (GRNNs)

GRNNs learn to map an RGB or RGB-D (color and depth) image or image sequence that depicts a static 3D world scene to a 3D feature map of the scene in an end-to-end differentiable manner while optimizing for view prediction: the inferred 3D feature maps, when projected from designated camera viewpoints, are neurally decoded to 2D RGB images and the weights of the neural architecture are trained to minimize RGB distance of the predicted image from the corresponding ground-truth RGB image view. We will denote the inferred 3D feature map as $\mathbf{M} \in \mathbb{R}^{W \times H \times D \times C}$—where W, H, D, C stand for width, height, depth and number of feature channels, respectively. Every (x, y, z) grid location in the 3D feature map $\mathbf{M}$ holds a 1-dimensional feature vector that describes the semantic and geometric properties of a corresponding 3D physical location in the 3D world scene. The map is updated with each new video frame while cancelling camera motion, so that information from 2D pixels that correspond to the same 3D physical point end-up nearby in the map. At training time, we assume a mobile agent that moves around in a 3D world scene and sees it from multiple camera viewpoints, in order to provide "labels" for view prediction to GRNNs. Upon training, GRNNs can map an RGB or RGB-D image sequence or single image to a complete 3D feature map of the scene it depicts, i.e., it learns to *imagine* the missing or occluded information; we denote this 2D-to-3D mapping as $\mathbf{M} = \text{GRNN}(I)$ for an input RGB or RGB-D image I.

3D object proposals. Given images with annotated 3D

object boxes, the work of Tung et al. (34) trained GRNNs for 3D object detection by learning a neural module that takes as input the 3D feature map $\mathbf{M}$ inferred from the input image and outputs 3D bounding boxes and binary 3D voxel occupancies (3D segmentations) for the objects present in the map. Their work essentially adapted the state-of-the-art 2D object detector Mask-RCNN (11) to have 3D input and output instead of 2D. We use the same architecture for our category-agnostic 3D region proposal network (3D RPN) in Section 3.3. For further details on GRNNs, please read Tung et al. (34).

3.2. Language-conditioned 3D visual imagination

We train generative networks to map language utterances to 3D feature maps of the scene they describe. They do so using a compositional generation process that conditions on the dependency tree of the utterance (assumed given) and generates one object at a time, predicting its appearance and location using two separate stochastic neural modules, *what* and *where*, as shown in Figure 2.

The *what* generation module $\mathrm{G_A}(p, z; \phi)$ is a stochastic generative network of object-centric appearance that given a noun phrase p learns to map the word embeddings of each adjective and noun and a random vector of sampled Gaussian noise $z \in \mathbb{R}^{50} \sim \mathcal{N}(0, I)$ to a corresponding fixed size 3D feature tensor $\bar{\mathbf{M}}^o \in \mathbb{R}^{w \times h \times d \times c}$ and a size vector $s^o \in \mathbb{R}^3$ that describes the width, height, and depth for the tensor. We resize the 3D feature tensor $\bar{\mathbf{M}}^o$ to have the predicted size s^o and obtain $\mathbf{M}^o = \mathrm{Resize}(\bar{\mathbf{M}}^o, s^o)$. We use a gated mixture of experts (30) layer—a gated version of point-wise multiplication—to aggregate outputs from different adjectives and nouns, as shown in Figure 2.

The *where* generation module $\mathrm{G_S}(s, z, \psi)$ is a stochastic generative network of cross-object 3D offsets that learns to map the one-hot encoding of a spatial expression s, e.g., *"in front of"*, and a random vector of sampled Gaussian noise $z \in \mathbb{R}^{50} \sim \mathcal{N}(0, I)$ to a relative 3D spatial offset $d\mathbf{X}^{(i,j)} = (dX, dY, dZ) \in \mathbb{R}^3$ between the corresponding objects. Let b_i^o denote the 3D spatial coordinates of the corners of a generated object.

Our complete generative network conditions on the dependency parse tree of the utterance and adds one 3D object tensor $\mathbf{M}_i^o, i = 1...K$ at a time to a 3-dimensional feature canvas according to their predicted 3D locations, where K is the number of noun phrases in the dependency tree: $\mathbf{M}^g = \sum_{i=1}^{K} \mathrm{DRAW}(\mathbf{M}_i^o, \mathbf{X}_i^o)$, where DRAW denotes the operation of adding a 3D feature tensor to a 3D location. The 3D location $\mathbf{X}^1$ of the first object is chosen arbitrarily, and the locations of the rest of the object are based on the predicted cross-object offsets: $\mathbf{X}_2^o = \mathbf{X}_1^o + d\mathbf{X}^{(2,1)}$. If two added objects intersect in 3D, i.e., the intersection over union of the 3D object bounding boxes is above a cross-validated threshold of 0.1, $\mathrm{IoU}(b_i^o, b_j^o) > 0.1$, we re-

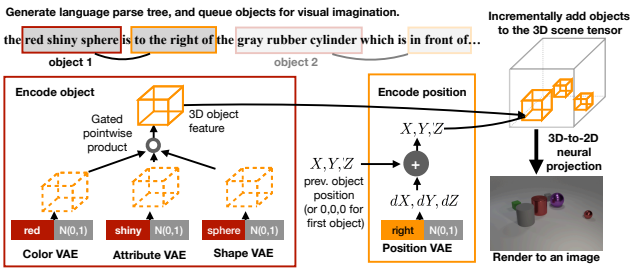

Figure 2: **Mapping language utterances to object-centric appearance tensors and cross-object 3D spatial offsets** using conditional *what-where* generative networks.

sample object locations until we find a scene configuration where objects do not 3D intersect, or until we reach a maximum number of samples—in which case we infer that the utterance is impossible to realize. By exploiting the constraint of non 3D intersection in the 3D feature space, our model can both generalize to longer parse trees than those seen at training time—by re-sampling until all spatial constraints are satisfied—as well as infer the plausibility of utterances, as we validate empirically in Section 4.2. In 3D, non-physically plausible object intersection is easy to distinguish from physically plausible object occlusion, something that is not easy to infer with 2D object coordinates, as we show empirically in Section 4.2. Given the 3D coordinates of two 3D bounding boxes, our model detects whether there exists 3D object interpenetration by simply thresholding the computed 3D intersection over union.

We train our stochastic generative networks using conditional variational autoencoders. We detail the inference networks in Section 1 of the supplementary file.

3.3. Detecting referential expressions in 3D

We train discriminative networks to map spatial referential expressions, e.g., *"the blue cube to the right of the yellow sphere behind the green cylinder"*, and related RGBD images, to the 3D bounding box of the objects the expressions refer to. Our model uses a compositional detection module conditioned on the dependency tree of the referential expression (assumed given). The compositional detection module has two main components: (1) an object appearance matching function that predicts a 3D appearance detector template for each noun phrase and uses the template to compute an object appearance matching score, and (2) a 3D spatial classifier for each spatial expression that computes a spatial compatibility score. We detail these components below. The compositional structure of our detector is necessary to handle referential expressions of arbitrary length. Our detector is comprised of a *what* detection module and a *where* detection module, as shown in Figure 3. The *what* module $\mathrm{D_A}(p; \xi)$ is a neural network

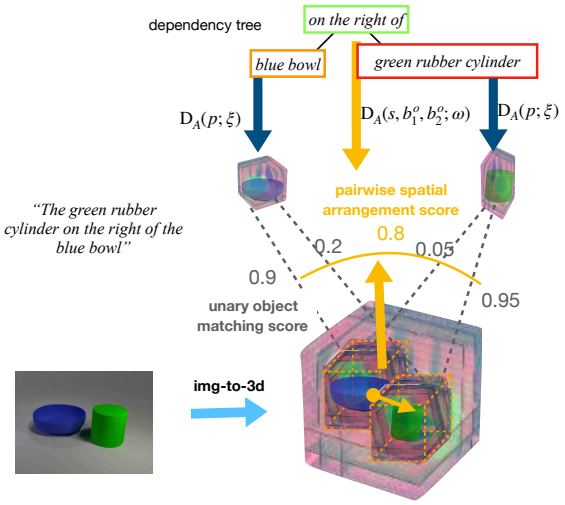

Figure 3: **3D referential object detection.** We score possible assignments of noun phrases to the detected 3D objects based on their appearance and pairwise spatial relations.

that given a noun phrase p learns to map the word embeddings of each adjective and noun to a corresponding fixed-size 3D feature tensor $f = \mathrm{D_A}(p;\xi) \in \mathbb{R}^{W\times H\times D\times C}$, we used $W = H = D = 16$ and $C = 32$. Our *what* detection module is essentially a deterministic alternative of the *what* generative stochastic network of Section 3.2. The object appearance score is obtained by computing the inner product between the detection template $\mathrm{D_A}(p;\xi)$ and the cropped object 3D feature map $\mathbf{F} = \mathrm{CropAndResize}(\mathbf{M}, b^o)$, where $\mathbf{M} = \mathrm{GRNN}(I)$ and b^o the 3D box of the object. We feed the output of the inner product to a sigmoid activation layer.

The *where* detection module $\mathrm{D_S}(s, b_1^o, b_2^o; \omega)$ takes as input the 3D box coordinates of the hypothesized pair of objects under consideration, and the one-hot encoding of the spatial utterance s (e.g., *"in front of", "behind"*), and scores whether the two-object configuration matches the spatial expression.

We train both the *what* and *where* detection modules in a supervised way. During training, we use ground-truth associations of noun phrases p to 3D object boxes in the image for positive examples, and random crops or other objects as negative examples. For cropping, we use ground-truth 3D object boxes at training time and detected 3D object box proposals from the 3D region proposal network (RPN) of Section 3.1 at test time.

Having trained our *what* and *where* detector modules, and given the dependency parse tree of an utterance and a set of bottom up 3D object proposals, we exhaustively search over assignments of noun phrases to detected 3D objects in the scene. We only keep noun phrase to 3D box assignments if their unary matching score is above a cross-validated threshold of 0.4. Then, we simply pick the assignment of noun phrases to 3D boxes with the highest product of unary and pairwise scores. Our 3D referential detector resembles previous 2D referential detectors (13; 4), but operates in 3D appearance features and spatial arrangements, instead of 2D.

3.4. Instruction following

Humans use natural language to program fellow humans e.g., *"put the orange inside the wooden bowl, please"*. Programming robotic agents in a similar manner is desirable since it would allow non-experts to also program robots. While most current policy learning methods use manually coded reward functions in simulation or instrumented environments to train policies, here we propose to use visual detectors of natural language expressions (32), such as *"orange inside the wooden basket,"* to automatically monitor an agent's progress towards achieving the desired goal and supply rewards accordingly.

We use the language-conditioned generative and detection models proposed in Section 3.2 and 3.3 to obtain a reliable perceptual reward detector for object placement instructions with the following steps, as shown in Figure 1 4th column: **(1)** We localize in 3D all objects mentioned in the instruction using the aforementioned 3D referential detectors. **(2)** We predict the desired 3D goal location for the object to be manipulated $\mathbf{x}^o_{goal}$ using our stochastic spatial arrangement generative network $\mathrm{G_S}(s, z; \psi))$. **(3)** We compute per time step costs being proportional to the Euclidean distance of the current 3D location of the object $\mathbf{x}^o_t$ and end-effector 3D location $\mathbf{x}^e_t$ assumed known from forward dynamics, and the desired 3D goal object location $\mathbf{x}^o_{goal}$ and end-effector 3D location $\mathbf{x}^e_{goal}$: $\mathcal{C}_t = \|\mathbf{x}_t - \mathbf{x}_{goal}\|^2_2$, where $\mathbf{x}_t = [\mathbf{x}^o_t; \mathbf{x}^e_t]$ is the concatenation of object and end-effector state at time step t and $\mathbf{x}_{goal} = [\mathbf{x}^o_{goal}; \mathbf{x}^e_{goal}]$. We formulate this as a reinforcement learning problem, where at each time step the cost is given by $c_t = \|\mathbf{x}_t - \mathbf{x}_{goal}\|_2$. We use i-LQR (iterative Linear Quadratic Regulator) (31) to minimize the cost function $\sum_{t=1}^{T} \mathcal{C}_t$. I-LQR learns a time-dependent policy $\pi_t(\mathbf{u}|\mathbf{x}; \theta) = \mathcal{N}(\mathbf{K}_t\mathbf{x}_t + \mathbf{k}_t, \mathbf{\Sigma}_t)$, where the time-dependent control gains are learned by model-based updates, where the dynamical model $p(\mathbf{x}_t|, \mathbf{x}_{t-1}, \mathbf{u}_t)$ of the *a priori* unknown dynamics is learned during training time. The actions $\mathbf{u}$ are defined as the changes in the robot end-effector's 3D position, and orientation about the vertical axis, giving a 4-dimensional action space.

We show in Section 4.4 that our method successfully trains multiple language-conditioned policies. In comparison, 2D desired goal locations generated by 2D baselines (32) often fail to do so.

4. Experiments

We test the proposed language grounding model in the following tasks: **(i)** Generating scenes based on language

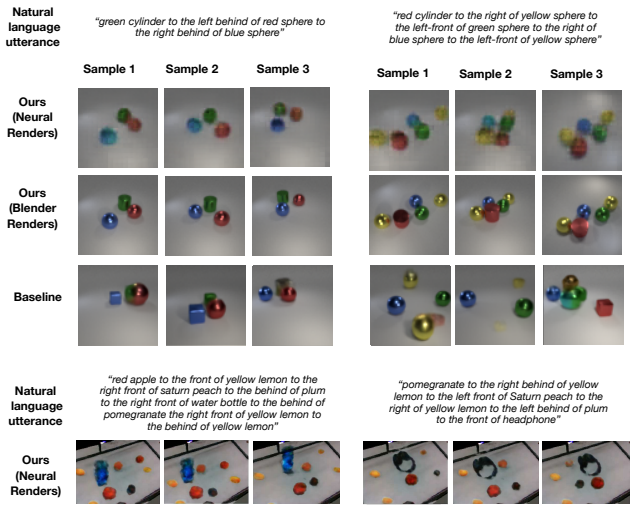

Figure 4: **Language to scene generation** (Rows 1,2,4) and **language to image generation** (Row 3) from our model and the model of Deng et al (6) for utterances longer than those encountered at training time on CLEVR and our real-world dataset. Both our model and the baseline are stochastic, and we sample three generated scenes per utterance.

utterances **(ii)** classifying utterances based on whether they describe possible or impossible scenes, **(iii)** detecting spatial referential expressions, and, **(iv)** following object placement instructions. We consider two datasets: (i) The CLEVR dataset of Johnson et al. (14) that contains 3D scenes annotated with natural language descriptions, their dependency parse trees, and the object 3D bounding boxes. The dataset contains Blender generated 3D scenes with geometric objects. Each object can take a number of colors, materials, shapes and sizes. Each scene is accompanied with a description of the object spatial arrangements, as well as its parse tree. Each scene is rendered from 12 azimuths and 4 elevation angles, namely, $\{12^o, 20^o, 40^o, 60^o\}$. We train GRNNs for view prediction using the RGB image views in the training sets. The annotated 3D bounding boxes are used to train our 3D object detector. We generate 800 scenes for training, and 400 for testing. The language is generated randomly with a **maximum of 2 objects** for the training scenes. (ii) A dataset we collected in the real world. We built a camera dome comprised of 8 cameras placed in a hemisphere above a table surface. We move vegetables around and collect multiview images. We automatically annotate the scene with 3D object boxes by doing 3D pointcloud subtraction at training time, We use the obtained 3D boxes to train our 3D object detector. At test time, we detect objects from a single view using our trained 3D detector. We further provide category labels for the vegetable present in single-object scenes to facilitate the association of labels to object 3D bounding boxes. More elaborate multiple instance learning techniques could be used to handle the general case of weakly annotated multi-object scenes (20). We leave this for future work. We show extensive qualitative results on our real world dataset as an evidence that our model can effectively generalize to real world data if allowed multiview embodied supervision and weak category object labels.

4.1. Language conditioned scene generation

We show language-conditioned generated scenes for our model and the baseline model of Deng et al. (6) in Figure 4 for utterances longer than those encountered at training time. The model of Deng et al. (6) generates a 2D RGB image directly (without an intermediate 3D representation) conditioned on a language utterance and its dependency tree. For each object mentioned in the utterance, the model of Deng et al. (6) predicts the absolute 2D location, 2D box size and a 2D appearance feature map for the object, and then it warps and places the 2D appearance feature map on a canvas according to the predicted location and object sizes. The canvas with 2D features is neurally decoded into an RGB image. We visualize our own model's predictions in two ways: i) **neural renders** are obtained by feeding the generated 3D assembled canvas to the 3D-to-2D neural projection module of GRNNs, ii) **Blender renders** are renderings of Blender scenes that contain 3D meshes selected by nearest neighbor to the language generated object 3D feature tensors, and arranged based on the predicted 3D spatial offsets.

Our model re-samples an object location when it detects that the newly added object penetrates the existing objects, with a 3D intersection-over-union (IOU) score higher than a cross-validated threshold of 0.1. The model of Deng et al. (6) is trained to handle occluded objects. Notice in Figure 4 that it generates weird configurations as the number of objects increase. We tried imposing constraints of object placement using 2D IoU threshold in our baseline, but ran into the problem that we could not find plausible configurations for strict IoU thresholds, and we would obtain non-sensical configurations for low IoU thresholds, we include the results in the supplementary file. Note that 2D IoU cannot discriminate between physically plausible object occlusions and physically implausible object intersection. Reasoning about 3D object non intersection is indeed much easier in 3D space.

Sections 2 and 3 of the supplementary file include more scene generation examples, where predictions of our model are decoded from multiple camera viewpoints, more comparisons against the baseline, and more details on the Blender rendering visualization. Please note that **image generation is not the end-task for this work; instead, it is a task to help learn the mapping from language to the 3D space-aware feature space.** We opt for a model that

has reasoning capabilities over the generated entities, as opposed to generating pixel-accurate images that we cannot reason on.

4.2. Affordability inference of natural language utterances

We test our model and baselines in their ability to classify language utterances as describing sensical or nonsensical object configurations. We created a test set of 92 NL utterances, 46 of which are affordable, i.e., describe a realizable object arrangement, e.g., *"a red cube is in front of a blue cylinder and in front of a red sphere, the blue cylinder is in front of the red sphere."*, and 46 are unaffordable, i.e., describe a non-realistic object arrangement, e.g., *"a red cube is behind a cyan sphere and in front of a red cylinder, the cyan sphere is left behind the red cylinder"*. In each utterance, an object is mentioned multiple times. The utterance is unaffordable when these mentions are contradictory. Answering correctly requires spatial reasoning over possible object configurations. **Both our model and the baselines have been trained only on plausible utterances and scenes. We use our dataset only for evaluation.** This setup is similar to violation of expectation (23): the model detects violations while it has only been trained on plausible versions of the world.

Our model infers affordability of a language utterance by generating the 3D feature map of the described scene, as detailed in Section 3.2. When an object is mentioned multiple times in an utterance, our model uses the first mention to add it in the 3D feature canvas, and uses the pairwise object spatial classifier $\mathrm{D_S}$ of Section 3.3 to infer if the predicted configuration also satisfies the later constraints. If not, our model re-samples object arrangements until a configuration is found or a maximum number of samples is reached.

We compare our model against a baseline based on the model of Deng et al. (6). Similar to our model, when an object is mentioned multiple times, we use the first mention to add it in the 2D image canvas, and use pairwise object spatial classifiers we train over 2D bounding box spatial information—as opposed to 3D—to infer if the predicted configuration also satisfies the later constraints. Note that there are no previous works that attempt this language affordability inference task, and our baseline essentially performs similar operations as our model but in a 2D space.

We consider a sentence to be affordable if the spatial classifier predicts a score above 0.5 for the later constraint. **Our model achieved an affordability classification accuracy of** 95% **while the baseline achieved** 79%. This suggests that reasoning in 3D as opposed to 2D makes it easier to determine the affordability of object configurations.

mAP	ours RGB-D	(22) RGB-D	ours RGB	(22) RGB
2D	**0.993**	0.903	0.990	0.925
3D	**0.973**	-	0.969	-

Table 1: **Mean average precision for category agnostic region proposals on Clevr dataset.** Our 3D RPN outperforms the 2D state-of-the-art RPN of Faster R-CNN (22).

4.3. Detecting spatial referential expressions

To evaluate our model's ability to detect spatial referential expressions, we use the same dataset and train/test split of scenes as in the previous section. For each annotated scene, we consider the first mentioned object as the one being referred to, that needs to be detected. In this task, we compare our model with a variant of the modular 2D referential object detector of Hu et al. (13) that also takes as input the dependency parse tree of the expression. We train the object appearance detector for the baseline the same way as we train our model using positive and negative examples, but the inner product is on 2D feature space as opposed to 3D. We also train a pairwise spatial expression classifier to map width, height and x,y coordinates of the two 2D bounding boxes and the one-hot encoding of the spatial expression, e.g., *"in front of"*, to a score reflecting whether the two boxes respect the corresponding arrangement. Note that our pairwise spatial expression classifier uses 3D box information instead, which helps it to generalize across camera placements.

Our referential detectors are upper bounded by the performance of the Region Proposal Networks (RPNs) in 3D for our model and in 2D for the baseline, since we use language-generated object feature tensors to compare with object features extracted from 2D and 3D bounding box proposals. We evaluate RPN performance in Table 1. An object is successfully detected when the predicted box has an intersection over union (IoU) of at least 0.5 with the groundtruth bounding box. For our model, we project the detected 3D boxes to 2D and compute 2D mean average precision (mAP). Both our model and the baseline use a single RGB image as input as well as a corresponding depth map, which our model uses during the 2D-to-3D unprojection operation and the 2D RPN concatenates with the RGB input image. Our 3D RPN that takes the GRNN map **M** as input better delineates the objects under heavy occlusions than the 2D RPN.

We show quantitative results for referential expression detection in Table 2 with groundtruth as well as RPN predicted boxes, and qualitative results in Figure 5. In the *"in-domain view"* scenario, we test on camera viewpoints that have been seen at training time, in the *"out-of-domain view"* scenario, we test on **novel camera viewpoints.** An object is detected successfully when the corresponding de-

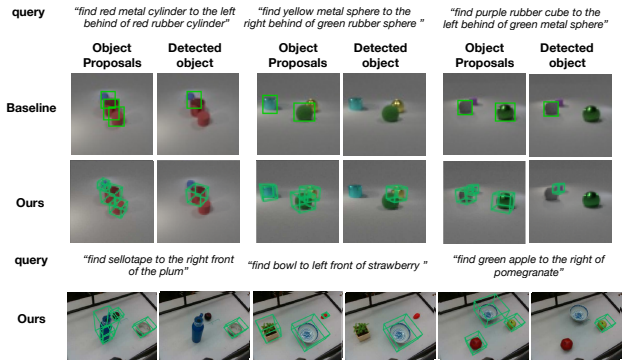

Figure 5: **Detecting referential spatial expressions.** On Clevr and our real world dataset, we show given a scene and a referential expression, our model localizes the object being referred to in 3D, while our baseline in 2D.

tected bounding box has an IoU of 0.5 with the groundtruth box (in 3D for our model and in 2D for the baseline). Our model greatly outperforms the baseline for two reasons: a) it better detects objects in the scene despite heavy occlusions, and, b) even with groundtruth boxes, our model generalizes better across camera viewpoints and object arrangements because the 3D representations of our model do not suffer from projection artifacts.

	Ours	(13)	Ours - GT 3D boxes	(13) - GT 2D boxes
in-domain view	0.87	0.70	**0.91**	0.79
out-of-domain view	0.79	0.25	**0.88**	0.64

Table 2: **F1-Score for detecting referential expressions.** Our model greatly outperforms the baseline with both groundtruth and predicted region proposals, especially for novel camera views on the CLEVR dataset.

4.4. Manipulation instruction following

We use the PyBullet Physics simulator (5) with simulated KUKA robot arm as our robotic platform. We use a *cube* and a *bowl*, using the same starting configuration for each scene, where the cube is held right above the bowl. We fix the end-effector to always point downwards.

We compare our model against the 2D generative baseline of (6) that generates object locations in 2D, and thus supply costs of the form: $\mathcal{C}^{2D}(\mathbf{x}_t) = \|\mathbf{x}_t^{2D} - \mathbf{x}_{goal}^{2D}\|_2^2$. We show in Table 3 success rates for different spatial expressions, where we define success as placing the object in the set of locations implied by the instruction. Goal locations provided in 2D do much worse in guiding policy search than target object locations in 3D supplied by our model. This is because 2D distances suffer from foreshortening and reflect planning distance poorly. This is not surprising: in fact, the robotics control literature almost always considers desired locations of objects to be achieved to be in 3D (18; 19). In our work, we link language instructions with such 3D inference using inverse graphics computer vision architectures for 2D to 3D lifting in a learnable 3D feature space. Videos of the learnt language-conditioned placement policies can be found here: https://mihirp1998.github.io/project_pages/emblang/

Language Exp.	left	left-behind	left-front	right	right-behind	right-front	in
Baseline	4/5	1/5	3/5	0/5	2/5	0/5	1/5
Ours	**5/5**	**3/5**	**5/5**	**5/5**	**5/5**	**3/5**	**5/5**

Table 3: **Success rates for executing instructions regarding object placement.** Policies learnt using costs over 3D configurations much outperform those learnt with costs over 2D configurations.

5. Discussion - Future Work

We proposed models that associate language utterances with compositional 3D feature representations of the objects and scenes the utterances describe, and exploit the rich constrains of the 3D space for spatial reasoning. We showed our model can effectively imagine object spatial configurations conditioned on language utterances, can reason about affordability of spatial arrangements, detect objects in them, and train policies for following object placement instructions. We further showed our model generalizes to real world data without real world examples of scenes annotated with spatial descriptions, rather, only single category labels. The language utterances we use are programmatically generated (14). One way to extend our framework to handle truly natural language is by paraphrasing such programmatically generated utterances (3) to create paired examples of natural language utterances and parse trees, then train a dependency parser (36) to generate dependency parse trees as input for our model using natural language as input. Going beyond basic spatial arrangements would require learning dynamics, physics and mechanics of the grounding 3D feature space. These are clear avenues for future work.

6. Acknowledgements

We would like to thank Shreshta Shetty and Gaurav Pathak for helping set up the table dome and Xian Zhou for help with the real robot placement experiments. This work was partially funded by Sony AI and AiDTR grant. Hsiao-Yu Tung is funded by Yahoo InMind Fellowship and Siemens FutureMaker Fellowship.

References

[1] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2016. 3

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 3

[3] J. Berant and P. Liang. Semantic parsing via paraphrasing. In *ACL (1)*, pages 1415–1425. The Association for Computer Linguistics, 2014. 8

[4] V. Cirik, T. Berg-Kirkpatrick, and L.-P. Morency. Using syntax to ground referring expressions in natural images. In *AAAI*, 2018. 5

[5] E. Coumans. Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses*, SIGGRAPH '15, New York, NY, USA, 2015. ACM. 8

[6] Z. Deng, J. Chen, Y. FU, and G. Mori. Probabilistic neural programmed networks for scene generation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4028–4038. Curran Associates, Inc., 2018. 3, 6, 7, 8

[7] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell. Language models for image captioning: The quirks and what works. *arXiv preprint arXiv:1505.01809*, 2015. 3

[8] B. Dhingra, H. Liu, W. W. Cohen, and R. Salakhutdinov. Gated-attention readers for text comprehension. *CoRR*, abs/1606.01549, 2016. 1

[9] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[10] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015. 3

[11] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. 4

[12] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc., 2015. 1

[13] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. 11 2016. 2, 3, 5, 7, 8

[14] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890, 2016. 3, 6, 8

[15] R. Kadlec, M. Schmid, O. Bajgar, and J. Kleindienst. Text understanding with the attention sum reader network. *CoRR*, abs/1603.01547, 2016. 1

[16] 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. 3

[17] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum. Deep convolutional inverse graphics network. *CoRR*, abs/1503.03167, 2015. 1, 2

[18] V. Kumar, A. Gupta, E. Todorov, and S. Levine. Learning dexterous manipulation policies from experience and imitation. *CoRR*, abs/1611.05095, 2016. 8

[19] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17(1):1334–1373, Jan. 2016. 8

[20] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *International Conference on Learning Representations*, 2019. 6

[21] B. A. Olshausen. Perception as an inference problem. 2013. 1

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. 7

[23] R. Riochet, M. Y. Castro, M. Bernard, A. Lerer, R. Fergus, V. Izard, and E. Dupoux. Intphys: A benchmark for visual intuitive physics reasoning. 2019. 7

[24] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[25] A. Rohrbach, M. Rohrbach, S. Tang, S. J. Oh, and B. Schiele. Generating descriptions with grounded and co-referenced people. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[26] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012. 3

[27] M. Rohrbach, Q. Wei, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 3

[28] L. Romaszko, C. K. I. Williams, P. Moreno, and P. Kohli. Vision-as-inverse-graphics: Obtaining a rich 3d explanation of a scene from a single image. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 1, 2

[29] M. Shah, X. Chen, M. Rohrbach, and D. Parikh. Cycle-consistency for robust visual question answering. *CoRR*, abs/1902.05660, 2019. 3

[30] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. 2017. 4

[31] Y. Tassa, N. Mansard, and E. Todorov. Control-limited differential dynamic programming. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1168–1175, 2014. 5

[32] F. Tung and K. Fragkiadaki. Reward learning using natural language. *CVPR*, 2018. 5

[33] H. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. *CoRR*, abs/1705.11166, 2017. 1, 2

[34] H.-Y. F. Tung, R. Cheng, and K. Fragkiadaki. Learning spatial common sense with geometry-aware recurrent networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 3, 4

[35] R. Vedantam, X. Lin, T. Batra, C. L. Zitnick, and D. Parikh. Learning common sense through visual abstraction. In *International Conference on Computer Vision (ICCV)*, 2015. 3

[36] D. Weiss, C. Alberti, M. Collins, and S. Petrov. Structured training for neural network transition-based parsing. *CoRR*, abs/1506.06158, 2015. 8

[37] J. Weston, S. Chopra, and A. Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014. 1

[38] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. *CoRR*, abs/1511.02274, 2015. 3