EMBODIED LANGUAGE GROUNDING WITH IMPLICIT 3D VISUAL FEATURE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Consider the utterance *"the tomato is to the left of the pot"*. Humans can answer numerous questions about the situation described, as well as reason through counterfactuals and alternatives, 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?", "would the tomato fit inside the pot?"* , and so on. Such reasoning capability remains elusive from current computational models of language understanding. To link language processing with spatial reasoning, we propose associating natural language utterances to a mental workspace of their meaning, encoded as 3-dimensional visual feature representations of the world scenes they describe. We learn such 3-dimensional visual representations—we call them visual imaginations— by predicting images a mobile agent sees while moving around in the 3D world. The input image streams the agent collects are *unprojected* into egomotion-stable 3D scene feature maps of the scene, and projected from novel viewpoints to match the observed RGB image views in an end-to-end differentiable manner. We then train modular neural models to generate such 3D feature representations given language utterances, to localize the objects an utterance mentions in the 3D feature representation inferred from an image, and to predict the desired 3D object locations given a manipulation instruction. We empirically show the proposed models outperform by a large margin existing 2D models in spatial reasoning, referential object detection and instruction following, and generalize better across camera viewpoints and object arrangements.

1 INTRODUCTION

The inspiring experiments of Glenberg and Robertson (20) in 1989 demonstrated that humans can easily judge the plausibility—a.k.a. affordability—of natural language utterances, such as *"he used a newspaper to protect his face from the wind"*, and the implausibility of others, such as *"he used a matchbox to protect his face from the wind"*. They suggest that **humans associate words with actual objects in the environment or prototypes in their imagination that retain perceptual properties—how the objects look—and affordance information** (18)—how the objects can be used. A natural language utterance is then understood through perceptual and motor *simulations* of explicitly and implicitly mentioned nouns and verbs, in some level of abstraction, that encode such affordances, e.g., the matchbox is too small to protect a human face from the wind, while a newspaper is both liftable by a human and can effectively cover a face when held appropriately. Indeed, given the inferred simulation, humans can reason about various aspects of the situation described in the utterance. We can answer questions such as "is the person raising one of his hands while protecting his face?", "how many free hands does he have?", "does the newspaper also protect his feet?", and so on. Could we operationalize these ideas into computational models of vision and language that would permit a machine to carry out similar types of reasoning that humans are capable of?

Existing models of vision and language associate natural language with 2D CNN activations for image captioning (8), visual question answering (2) or language to image generation (11). Their performance has been steadily improving over the years (44), yet, they lack basic common sense (3). For example, they 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

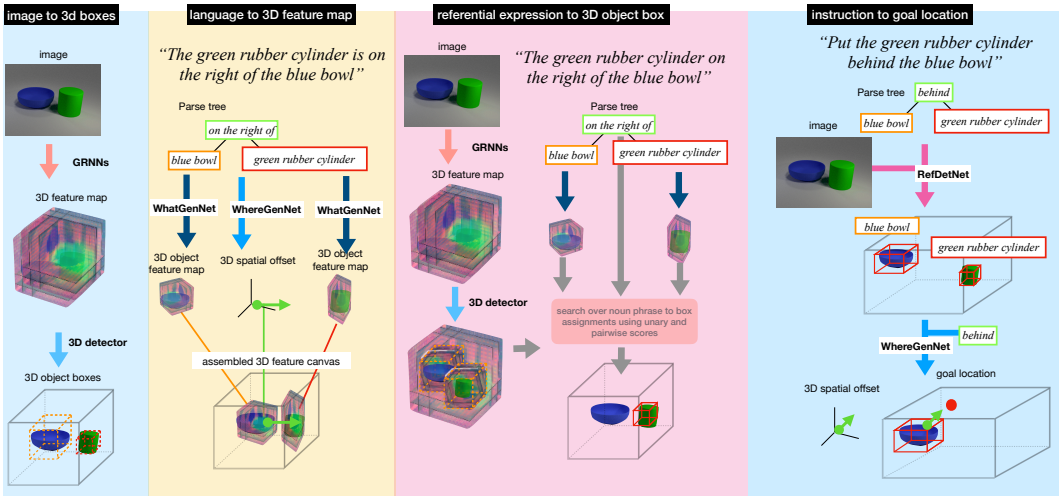

Figure 1: **Embodied language grounding with implicit 3D visual feature representations.** We map RGB images to 3D feature maps of the scene they depict, and 3D object boxes of the objects present (column 1) building upon the method of Tung et al. (50). We map an utterance and its parse tree to object-centric 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 the scene and infer the 3D desired location for the object to be manipulated (column 4). We use predicted location to supply rewards for trajectory optimization of placement policies.

the mug continues to exist if the camera changes viewpoint, and so on. These are simple facts, commonly available to all of us, even to 18 month old infants. We conjecture that this is because existing models ground language on **2D boxes and 2D visual feature representations, which are disconnected from physical scene understanding**: 2D CNN feature activations and 2D object boxes "move" under camera motion, disappear and re-appear arbitrarily during occlusions and dis-occlusions, and change size due to camera zooms; they do not obey intuitive physics constraints or basic spatial common sense constraints. It is further unclear what supervision is necessary for such reasoning ability to emerge in current model architectures.

We propose associating natural language utterances to space-aware 3D visual feature representations of their meaning, akin to abstract visual simulations. The proposed 3D visual feature representations **obey spatial constraints**, such as, object 3D non-intersection, object size constancy, object permanence across camera motion. These 3D visual feature representations **emerge in an unsupervised manner** in neural architectures with geometry-aware 3D representation bottlenecks trained for predicting views a mobile agent sees by moving around in the 3D world (50). After training, **these architectures learn to map video streams or single RGB images to complete 3D feature maps of the scene**, impainting by imagination occluded or missing details of the 2D input. In the inferred 3D feature maps objects have 3D extent, do not 3D-intersect, persist over time through occlusions, maintain their size over time despite camera zooms. We train modular **generative networks to map natural language utterances to the 3D feature map of the scene they describe**, guided by the structure of the utterance's parse tree, as shown in Figure 1, 2nd column. We demonstrate the benefits of associating language to 3D feature representations in three basic language understanding tasks:

(1) Affordability reasoning Our model can classify affordable (plausible) and unaffordable (implausible) spatial expressions, such as *"A to the left of B, B to the left of C, C to the right of A"* is non-affordable, while *"A to the left of B, B to the left of C, C to the left of A"* is affordable, where A, B, C any object mentions. Further, it can process utterances much longer than those seen at training time exploiting 3D non-intersection to reject impossible object arrangements during generation of the 3D object-factorized feature map of the scene. We show our model outperforms modular 2D baselines that map utterances to 2D images—instead of 3D feature maps.

(2) Referential expression detection We train discriminative networks to map the parse tree of a referential spatial expression, e.g., *"the blue sphere behind the yellow cube"*, and an RGB image to 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 2D models.

(3) Instruction following We train conditional generative networks to map a natural language instruction (e.g., "orange inside the wooden bowl") and an image of the scene to the 3D feature map of the desired scene described in the instruction, by identifying the object to be manipulated and generating its desired 3D goal location, as shown in Figure 1 4rth column. We use such 3D goal location in trajectory optimization of placement policies. We show our model successfully executes natural language instructions, while 2D target object locations provided by 2D baselines fail to find a successful placement.

Our datasets and code will be made publicly available upon publication.

2 EMBODIED LANGUAGE GROUNDING

We consider a dataset of 3D scenes annotated with corresponding natural language descriptions and their parse trees, and a reference camera viewpoint. We further assume access at training time to 3D object bounding boxes and correspondences between those and object referents in the natural language parse trees. Lastly, for each 3D scene, we assume we can sample camera viewpoints and observe the corresponding 2D RGB images.

In Section 2.1, we describe geometry-aware recurrent inverse graphics architectures of Tung et al. (50) that learn to map 2D image streams to 3D visual feature maps via view prediction, without any language supervision. These 3D visual feature maps are at the heart of our approach. We call our model *embodied* since training the 2D image to 3D feature mapping requires supervision from a mobile—or more generally embodied—agent to move around in the 3D world and collect (posed) images. In Section 2.2, we describe generative networks that condition on the parse tree of a natural language utterance and generate an object-factorized 3D feature map of the scene the utterance depicts. In Section 2.3, we describe referential expression detectors that condition on a referential expression and the inferred 3D feature map to localize in 3D the object being referred to. In Section 2.4, we show how our language to 3D map generation and referential detectors can be used for following object placement instructions.

2.1 LEARNING 3D FEATURE REPRESENTATIONS BY GEOMETRY-AWARE VIEW PREDICTION

Mobile agents have access to their egomotion, and can observe sensory outcomes of their motions and interactions. Training sensory representations to predict such outcomes is a useful form of supervision, free of human annotations, often termed *self-supervision* since the "labels" are provided by the embodied agent itself. Our agent is equipped with a neural architecture that takes as input K images and a randomly sampled camera viewpoint V and predicts the RGB image to be seen from that queried viewpoint. It does so using geometry-aware RNNs (GRNNs) of Tung et al. (50). These are modular end-to-end differentiable neural architectures that integrate visual features across video frames by first "lifting" them in 3D in a geometrically consistent manner, and then projecting them from the corresponding query camera viewpoint to generate the desired image. In contrast to popular video sequence models used in the literature (27; 46) whose hidden state is image-centric, GRNNs' hidden state is world-centric: it is a multi-dimensional tensor $\mathbf{M}$ with 3 spatial dimensions (X, Y, Z) and multiple feature dimensions, akin to a 3D map of the scene, which for every (x, y, z) grid location holds 1-dimensional feature vector, as we show in Figure 1 (a). The feature vector describes the semantic and geometric content of the corresponding 3D physical point in the 3D world scene. The map is updated with each new video frame in an **egomotion-stabilized manner**: deep features are transformed to cancel the (estimated) egomotion of the camera before updating the map, so that information from 2D pixels that correspond to the same 3D physical point end-up nearby in the map. For further details, please see the appendix and (50). Upon training, our model 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 knows how to *imagine* the missing or occluded information, we write this mapping as $\mathbf{M} = \text{GRNN}(I, \theta)$ for a single image I.

3D object detection. Given images with annotated 3D object boxes, we train a 3D object detector that takes as input the 3D feature map $\mathbf{M}$ inferred from the image and detects 3D bounding boxes for

the objects present. Our 3D object detector is an adaptation of the state-of-the-art 2D object detector, Mask-RCNN (23), to have 3D input and output instead of 2D, similar to Tung et al. (50). The same 3D detector can be used to detect objects in image streams as opposed to single images, since the visual input is integrated in a geometrically-consistent manner in the 3D map $\mathbf{M}$ as described above.

2.2 LANGUAGE-CONDITIONED 3D SCENE GENERATION

Our model associates natural language utterances to 3D feature maps of the scene they describe. It is a modular neural architecture comprised of a *what* module and a *where* module. The *what* module $\mathrm{A}^{\mathrm{O}}(p, z, \phi)$ is an object-centric appearance stochastic generative network that given a noun phrase p learns to map the one-hot encoding of each adjective and noun and a random vector of sampled Gaussian noise z to a corresponding fixed size 3D feature tensor $o^f \in \mathbb{R}^{w \times h \times d \times c}$ and a size vector $o^s \in \mathbb{R}^3$ that describes the width, height, and depth for the tensor. We resize the 3D feature tensor o^f to have the predicted size o^s. To aggregate outputs from different adjectives and nouns, the network combines 3D feature tensors and size vectors from different words using a gated mixture of experts (45) module—a gated version of point-wise multiplication, as shown in Figure 4. The *where* module $\mathrm{S}^{\mathrm{O}}(s, z, \psi)$ is a stochastic generative network 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 to a relative 3D spatial offset $d\mathbf{X} = (dX, dY, dZ) \in \mathbb{R}^3$ between the corresponding objects.

Our generative network adds one 3D object tensor at a time to a 3-dimensional feature canvas according to their predicted relative 3D locations. **If two generated objects interpenetrate in 3D, we resample object appearances and locations until we find a configuration where objects do not 3D interpenetrate**, or until we reach a maximum number of samples—in which case we infer the utterance is not affordable, i.e., it is impossible to realize. By exploiting the constraint of non 3D intersection of the 3D feature space, our model can both generalize to longer parse trees than seen at training time—by resampling until all spatial constraints are satisfied—as well as infer affordability of utterances, as we validate empirically in Section 3.

We train our generative networks using conditional variational autoencoders, as shown in Figure 4. For the object-centric 3D feature generative model, our inference network conditions on one-hot encoding of the adjectives and the noun, as well as the 3D feature tensor obtained by cropping the 3D feature map $\mathbf{M} = \mathrm{GRNN}(I, \theta)$ using the ground truth 3D object bounding box. For the cross-object 3D spatial object generative network, the corresponding inference network conditions on one-hot encoding of the spatial expression, as well as the 3D related offset, available from 3D object box annotations. Inference networks are used only at training time. Our *what* and *where* decoders take the posterior noise and predict 3D object appearance feature tensors, and cross-object 3D spatial offsets, respectively, for each object, and add those to a 3D feature canvas. The composed 3D feature canvas is projected from various camera viewpoints and is decoded to 2D RGB images using the corresponding neural modules of GRNNs. We train our network so that the language-inferred and image-inferred 3D feature maps are close in feature distance. Specifically, we optimize the following objectives: in 3D, the predicted 3D object appearance feature tensors and cross-object 3D relative spatial offsets minimize their distance against the 3D object feature tensors obtained by cropping the image inferred map $\mathbf{M} = \mathrm{GRNN}(I, \theta)$ using groundtruth 3D object boxes, and the ground-truth cross-object 3D relative offsets, respectively, and in 2D, the decoded predicted image minimizes a reconstruction loss against the groundtruth image view.

2.3 DETECTING REFERENTIAL EXPRESSIONS

We train a language-conditioned 3D object detector that given a spatial expression, e.g., *"the blue cube to the right of the yellow sphere behind the green cylinder"*, localizes the object being referred to in 3D. Our detector combines **two detection scores: one from matching object appearances, and one from matching pairwise object spatial arrangements.** Our object matching score is obtained by computing inner product between normalized the language-generated 3D object appearance features $\mathrm{A}^{\mathrm{O}}(p, \phi)$, obtained by a deterministic alternative of the stochastic network of Section 2.2, and the cropped object 3D feature map $\mathrm{Crop}(\mathbf{M}, b^o)$, and feeding the output in a sigmoid activation layer. During training, we use ground-truth associations of noun phrases p to 3D object boxes in the image b^o for positive examples, and random crops or other objects as negative examples. For cropping, we use ground-truth 3D object boxes b^o_{gt} at training time and detected 3D object box proposals b^o_{pred} from the detector of Section 2.1 at test time. Our pairwise matching score is obtained by a spatial classifier $S(p, b^{o_1}, b^{o_2})$ that takes as input the 3D box coordinates of the the

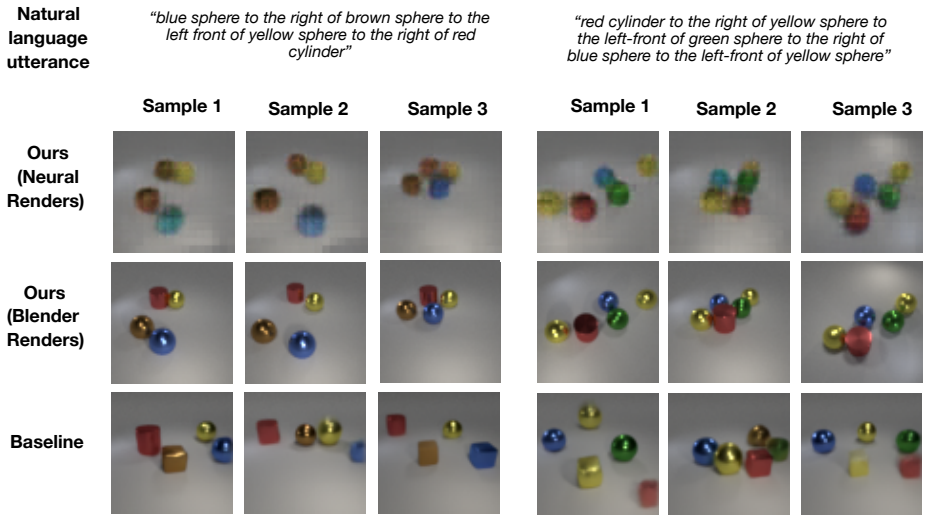

Figure 2: **Language to scene generation** for utterances longer than training utterances for our model (*rows 1,2*) and the 2D baseline model of (11) (*row 3*). Both our model and the baseline are stochastic, and we show three generated scenes per utterance. The baseline changes the shape of the objects arbitrarily (the brown sphere is mapped to a cylinder and the red cylinder to a cube).

pair of objects under consideration, and the word embedding of the spatial utterances p (e.g., *"in front of", "behind"*), and scores whether the object configuration matches the utterance. We use positive examples from our training set and negative examples from competing expressions as well as synthetic 3D object boxes in random locations. Having trained unary and pairwise detectors, and given the parse of an utterance and a set of bottom up object 3D proposals, we exhaustive search over assignments of noun phrases to detected 3D objects in the image. We only keep noun phrase to 3d box assignments if the 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 (28; 9), but operates in 3D appearance features and arrangements, instead of 2D.

2.4 Instruction following

Humans use natural language to program fellow humans e.g., *"please, put the orange inside the wooden bowl"*. We would like to be able to program robotic agents in a similar manner. Most current policy learning methods use manually coded reward functions in simulation or instrumented environments to train policies, as opposed to visual detectors of natural language expressions (49). If visual detectors of *"orange inside the wooden basket"* were available, we would use them to automatically monitor if the agent is succeeding in achieving the desired goal and supply rewards accordingly, as opposed to hard-coding them in the environment.

We use the model proposed in this work to obtain a reliable perceptual reward detector for object placement instructions with the following steps, as shown in Figure 1 4rth column: **(1)** We localize in 3D all objects mentioned in the utterance using the aforementioned referential 3D object detectors. **(2)** We predict the desired goal 3D location $\mathbf{x}_{goal}$ for the object to be manipulated using our stochastic spatial arrangement generative network $\mathrm{S}^{\mathrm{O}}(s, z, \psi))$. **(3)** We supply rewards during trajectory optimization (48) inversely proportional to the Euclidean distance of the current location $\mathbf{x}$ of the object from the desired one $\mathbf{x}_{goal}$. We show in Section 3 that our method successfully trains multiple language-conditioned policies. In comparison, 2D desired goal locations generated by 2D baselines often fail to do so.

3 Experiments

We test the proposed language grounding model in the following tasks: i) Inferring affordability of natural language descriptions, ii) detecting spatial referential expressions, and, iii) following object placement instructions. We consider the CLEVR dataset of Johnson et al. (29) that contains 3D

scenes annotated with natural language descriptions, their parse trees, and the object 3D bounding boxes. The dataset contains Blender generated 3D scenes with geometric objects (Figure 1). 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 3 elevations from cameras places on a viewing sphere of 8 units, to give a total of 36 RGB views. We train GRNNs for view prediction using the RGB image views in the dataset. The annotated 3D bounding boxes are used to train our 3D object detector.

3.1 AFFORDABILITY INFERENCE OF NATURAL LANGUAGE UTTERANCES

We created a dataset of 100 NL utterances, 50 of which are affordable, i.e., describe a realizable object arrangement, e.g., *"a red cube in front of a blue cylinder and in front of a red sphere, the blue cylinder is in front of the red sphere."*, and 50 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.

Our model infers affordability of a NL utterance by generating the described scene in an implicit 3D feature space, as described in Section 2.2. When an object is mentioned multiple times, our model uses the first mention to add it in the 3D feature canvas, and uses pairwise object arrangement classifiers of Section 2.3 to infer if the predicted configuration also satisfies the later constraints. If not, it resamples obejct arrangements until a configuration is found or a maximum number of samples is reached. There are no previous works that attempt this reasoning task. We compare our model against a baseline based on the model of Deng et al. (11) that generates a 2D RGB image conditioned on a NL utterance and its parse tree, the same input as our model. Deng et al. (11) predict *absolute* 2D locations and 2D box sizes for objects and their 2D appearance feature maps, warped in predicted locations, and decoded into an RGB image. Similar to our model, when an object is mentioned multiple times, we use the first mention to add it in the image, and use pairwise object arrangement classifiers over 2D bounding box spatial information—as opposed to 3D—to infer if the predicted configuration also satisfies the later constraints. We show results for affordability prediction in Table 1. In the *fixed camera elevation distribution* setup, we sample camera elevations uniformely from $\{20^o, 40^o, 60^o, 12^o\}$ both during training and test time, while in the *varying camera elevation distribution* setup, we train using images from camera elevation of $20^o, 40^o, 60^o$ and test on elevation of 12^o. In both cases, our model outperforms the baseline, and the gap is larger in case of novel camera elevations. This suggests our model can better generalize across camera viewpoints.

We show language-conditioned generated scenes for our model and the baseline in Figure 2. Both models re-sample an object location when they detect the intersection-over-union of the newly added object to be higher than a cross-validated threshold. We visualize our model's predictions in two ways: i) **neurally rendered** are obtained by feeding the generated 3D assembled canvas to the 3D-to-2D projection neural module of GRNNs, ii) **Blender rendered** are renderings of Blender scenes that contain 3D object models selected by the feature closest to the to the language generated 3D object feature tensors, and arranged based on the predicted 3D spatial offsets. We consider a database of 300 3D object meshes to choose from. To get the object feature tensor for a candidate 3D object model, we render multi-view RGB-D data of this object in Blender, and input them to the GRNN to obtain the corresponding feature map, which we crop using the groundtruth bounding box. Blender renders better convey object appearance because the neurally rendered images are blurry. Despite pixel images being blurry, our model retrieves correct objects that match the natural language descriptions. While the baseline generates realistic RGB images, it arbitrarily changes the shape of the objects, its reasoning capability regarding possible and impossible utterances is worse than our model. More scene generation examples are included in the appendix. Please note that **image generation is not the end-task for this work, rather, it is a task to help learn the mapping from language to the 3D space-aware abstract feature space.** Humans are believed to reason about language meaning using visual abstractions as opposed to pixel-perfect pictures (33). 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.

	Ours	Baseline	Ours - GT 3D boxes	Baseline - GT 2D boxes
Fixed C.E.D.	0.87	0.70	**0.91**	0.79
Varying C.E.D.	0.79	0.25	**0.88**	0.64

Table 3: **F1-Score for detecting spatial referential expressions.** Our model greatly outperforms the baseline both with groundtruth and with predicted region proposals.

	ours	baseline
Fixed C.E.D.	**0.95**	0.80
Varying C.E.D.	**0.83**	0.56

Table 1: **Accuracy for utterance affordability prediction.** C.E.D. stands for camera elevation distribution.

Mean AP	ours RGB+depth	RPN of (37) RGB+depth	ours RGB	RPN of (37) RGB
2D	**0.993**	0.903	0.990	0.925
3D	**0.973**	-	0.969	-

Table 2: **Mean average precision for category agnostic region proposals.** Our 3D RPN outperforms the 2D state-of-the-art RPN of Faster R-CNN (37).

3.2 Detecting referential spatial expressions

For each annotated scene, we consider the first mentioned object as the one being referred to, that needs to be detected. We use the same dataset and train/test split of scenes as before. In this task, we compare our model with a variant of the modular 2D referential object detector of Hu et al. (28) that also takes as input the parse tree of the expression. Same as our model, we train the noun phrase detector for the baseline using metric learning our region proposal extracted features, and the pairwise spatial expression classifier to map width, height and x,y coordinates of the two 2D bounding boxes and the embedding of the spatial expression, e.g., *"in front of"* obtained with a bi-LSTM, to a score reflecting whether the two boxes respect the corresponding arrangement. Note that our pairwise spatial expression classifier use 3D box information instead, and thus we expect it can better 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 compare language generated features to bottom-up region extracted ones. We compare RPN performance in Table 2. An object is successfully detected when the predicted box has a intersection over union (IoU) at least 0.5 with the groundtruth box. For our model, we project the detected 3D boxes to 2D to compute 2D mean average precision (mean AP). Both our model and the baseline use a single RGB image as input as well as its depth, which our model using during the 2D-to-3D unprojection operation and the 2D RPN concatenates across channels the depth map with the RGB input image. Our 3D RPN that takes the GRNN map $\mathbf{M}$ as input better delineates the objects under heavy occlusions than the 2D RPN of faster-RCNN. Moreover, our RPN improves with more views of the scene available (simply accumulating information in the memory map $\mathbf{M}$), while it is unclear how our 2D baseline RPN can take advantage of more views.

We show quantitative results for referential expression detection in Table 3 with groundtruth as well as RPN predicted boxes, and qualitative results in Figure 3. An object is detected successfully when the corresponding detected 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, because the 3D representations of our model do not suffer from projection artifacts, they better generalize across camera viewpoints and object arrangements.

3.3 Manipulation instruction following

We test our model in its ability to map NL instructions to desired object goal configurations in 3D and supply costs for trajectory optimization of object placements of the form: $\mathcal{C}^{3D}(\mathbf{x}_t) = \|\mathbf{x}_t^{3D} - \mathbf{x}_{goal}^{3D}\|_2^2$. We compare against the 2D generative baseline of (11) that generates object locations in 2D, and thus supply costs of the form: $\mathcal{C}^{2D}(\mathbf{x}_t) = \|\mathbf{x}_t^{2D} - \mathbf{x}_{goal}^{2D}\|_2^2$.

Simulation setup We use the PyBullet Physics simulator (1) with similar setup as our CLEVR scenes. We use a 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 by the robot right above

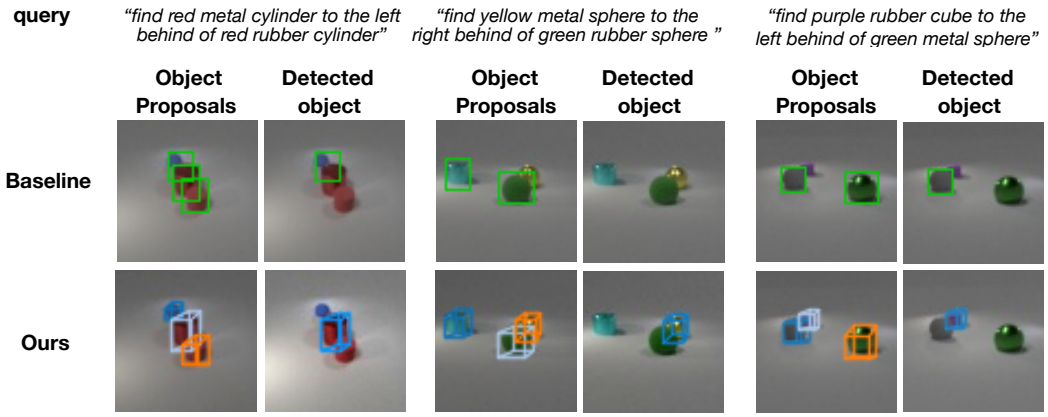

Figure 3: **Detecting referential spatial expressions.** Given a scene and a referential expression, our model localizes the object being referred to in 3D, while our baseline in 2D.

the bowl. The instruction described the desired location of the cube. The generated spatial offsets are then used to provide a cost function to learn a policy that guides the robot end-effector to the goal location. We use LQR-based trajectory optimization (48) to minimize: $\mathcal{C}(\mathbf{x}_t) = \|\mathbf{x}_t - \mathbf{x}_{goal}\|_2^2$. $\mathbf{x}_t$ denotes the state at time step t where the state is defined as a concatenated feature vector of the robot's 3-dimensional end-effector position and the relative spatial location of the cube to the bowl. The actions $\mathbf{u}$ are defined as the translational changes in the robot's end-effector 3D position. We fix the end-effector to always point downwards, and we assume the cube to be grasped at the beginning of an episode.

We show in Table 4 success rates for different spatial expressions, where we define success as placing a cube within a bounded cone around the orientation normal of a given expression. Success is defined as placing the cube within a 22.5 degree cone around the orientation normal and below the maximum height of the bowl. Goal locations provided in 2D do much worse than target object locations in 3D supplied by our model in guiding policy search. This is because 2D distances suffer from foreshortening and reflect planning distance worse than 3D ones. Videos of the learnt language-conditioned placement policies can be seen here : https://sites.google.com/view/embodiedlanguagegrounding/home

Language Exp.	left	left-behind	left-front	right	right-behind	right-front	inside
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 4: **Success rates for reaching desired goals specified by different language expressions.**

Limitations The proposed model has two important limitations. First, it has been tested on a restricted domain of spatial object arrangements. Second, it relies on strong language supervision (parse trees), 3D object bounding boxes, and correspondences between object referents in the parse tree and 3D object boxes. Extending this language domain to actions and verbs, as well as relaxing such supervision by employing curricula (35), are direct avenues for future work.

4 Conclusion

We proposed models that learn to associate natural language utterances with compositional 3D feature representations of objects and scenes. We showed the benefits of the proposed models in affordability inference, 3D referential detection, and following object placement instructions. We believe our model is a first step towards injecting basic spatial common sense into language understanding, not by reading large amounts of text, but rather, linking language to visual simulations and exploiting the rich constraints for the 3D space. Going beyond basic spatial common sense would require learning dynamics, physics and mechanics of the grounding 3D feature space. This is the avenue of our future work.

REFERENCES

[1] http://bulletphysics.org/wordpress/. 7

[2] 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. 1, 14

[3] 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. 1, 14

[4] L. Aziz-Zadeh, C. Fiebach, S. Narayanan, J. Feldman, E. Dodge, and R. Ivry. Modulation of the ffa and ppa by language related to faces and places. *Social neuroscience*, 3:229–38, 02 2008. 16

[5] B. Bergen. Mental simulation in spatial language processing. 01 2005. 14

[6] B. Bergen. Experimental methods for simulation semantics. *Methods in cognitive linguistics*, pages 277–301, 2007. 15

[7] B. Bergen. Embodiment, simulation and meaning. *The Routledge Handbook of Semantics*, pages 142–157, 01 2015. 14

[8] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015. 1

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

[10] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Iccv*, volume 3, pages 1403–1410, 2003. 11

[11] 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. 1, 5, 6, 7, 12, 14, 17

[12] 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. 14

[13] B. Dhingra, H. Liu, W. W. Cohen, and R. Salakhutdinov. Gated-attention readers for text comprehension. *CoRR*, abs/1606.01549, 2016. 13

[14] 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. 14

[15] 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. 14

[16] J. Feldman and S. Narayanan. Embodied meaning in a neural theory of language. *Brain and language*, 89:385–92, 06 2004. 14

[17] J. A. Feldman. *From Molecule to Metaphor: A Neural Theory of Language*. MIT Press, Cambridge, MA, 2006. 14

[18] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979. 1

[19] A. Glenberg and M. Kaschak. Grounding language in action. *Psychonomic Bulletin and Review*, 9(3):558–565, 9 2002. 15

[20] A. Glenberg and D. Robertson. Symbol grounding and meaning: A comparison of high-dimensional and embodied theories of meaning. *Journal of Memory and Language*, 2000. 1

[21] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2017. 16

[22] S. Harnad. The symbol grounding problem. *Phys. D*, 42(1-3):335–346, June 1990. 13

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

[24] J. F. Henriques and A. Vedaldi. MapNet: An allocentric spatial memory for mapping environments. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 16

[25] 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. 13

[26] D. Hewlett, A. Lacoste, L. Jones, I. Polosukhin, A. Fandrianto, J. Han, M. Kelcey, and D. Berthelot. Wikireading: A novel large-scale language understanding task over wikipedia. *CoRR*, abs/1608.03542, 2016. 13

[27] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. 3

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

[29] 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. 5

[30] R. Kadlec, M. Schmid, O. Bajgar, and J. Kleindienst. Text understanding with the attention sum reader network. *CoRR*, abs/1603.01547, 2016. 13

[31] A. Kar, C. Häne, and J. Malik. Learning a multi-view stereo machine. *CoRR*, abs/1708.05375, 2017. 16

[32] 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. 14

[33] S. M. Kosslyn. *Image and Brain: The Resolution of the Imagery Debate*. MIT Press, Cambridge, MA, USA, 1994. 6

[34] R. M Willems, I. Toni, P. Hagoort, and D. Casasanto. Neural dissociations between action verb understanding and motor imagery. *Journal of cognitive neuroscience*, 22:2387–400, 11 2009. 16

[35] 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. 8

[36] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016. 13

[37] 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

[38] 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. 14

[39] 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. 14

[40] 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. 14

[41] 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. 14

[42] A. P. Saygin, S. Mccullough, M. Alac, and K. Emmorey. Modulation of bold response in motion-sensitive lateral temporal cortex by real and fictive motion sentences. *Journal of cognitive neuroscience*, 22:2480–90, 11 2009. 16

[43] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016. 13

[44] M. Shah, X. Chen, M. Rohrbach, and D. Parikh. Cycle-consistency for robust visual question answering. *CoRR*, abs/1902.05660, 2019. 1

[45] 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

[46] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 802–810, Cambridge, MA, USA, 2015. MIT Press. 3

[47] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhöfer. DeepVoxels: Learning persistent 3d feature embeddings. *arXiv preprint arXiv:1812.01024*, 2018. 16

[48] 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, 8

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

[50] H.-Y. F. Tung, R. Cheng, and K. Fragkiadaki. Learning spatial common sense with geometry-aware recurrent networks. *arXiv:1901.00003*, 2018. 2, 3, 4, 11, 16

[51] 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. 14

[52] J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698, 2015. 13

[53] J. Weston, S. Chopra, and A. Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014. 13

[54] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. *CoRR*, abs/1511.02274, 2015. 14

A APPENDIX

B OVERVIEW OF GEOMETRY-AWARE RECURRENT NEURAL NETWORKS (GRNNS) (50)

GRNNs are RNNs that have a 4D latent state $\mathbf{M}^{(t)} \in \mathbb{R}^{w \times h \times d \times c}$, which has spatial resolution $w \times h \times d$ (width, height, and depth) and feature dimensionality c (channels). At each time step, they estimate the rigid transformation between the current camera viewpoint and the coordinate system of the latent map $\mathbf{M}^{(t)}$, then rotate and translate the features extracted from the current input view $I^{(t)}$ and depth map $D^{(t)}$ to align them with the coordinate system of the latent map, and convolutionally update the latent map using a standard convolutional 3D GRU, 3D LSTM or plain feature averaging. We refer to the memory state as the model's imagination to emphasize that most of grid points in $\mathbf{M}^{(t)}$ will not be observed by any sensor, and so the feature content is "imagined" by the model.

Below we present the individual modules of GRNNs in detail which allow the model to differentiably go back and forth between 2D pixel observation space and 3D imagination space.

2D-to-3D unprojection This module converts the input RGB image $I^{(t)} \in \mathbb{R}^{w \times h \times 3}$ and depth map $D^{(t)} \in \mathbb{R}^{w \times h}$ into a 4D tensor $[\mathbf{U}^{(t)}, \mathbf{O}^{(t)}] \in \mathbb{R}^{w \times h \times d \times 4}$, by filling the 3D imagination grid $\mathbf{U}^{(t)} \in \mathbb{R}^{w \times h \times d \times 4}$ with samples from the 2D image pixel grid using perspective (un)projection, and mapping our depth map to a binary occupancy voxel grid $\mathbf{O}^{(t)} \in \mathbb{R}^{w \times h \times d \times 1}$, by assigning each voxel a value of 1 or 0, depending on whether or not a point lands in the voxel.

Latent map update This module aggregates egomotion-stabilized (registered) feature tensors into the memory tensor $\mathbf{M}^{(t)}$. We denote registered tensors with a subscript λ. We treat the first camera position as the reference system thus $\mathbf{U}^{(0)} = \mathbf{U}^{(0)}_{\text{reg}}$ (and $\mathbf{O}^{(0)} = \mathbf{O}^{(0)}_{\text{reg}}$). We first pass the registered tensors $[\mathbf{U}^{(t)}_{\text{reg}}, \mathbf{O}^{(t)}_{\text{reg}}]$ through a series of 3D convolution layers, producing a 3D feature tensor for the timestep, denoted $\mathbf{F}^{(t)}_{\text{reg}} \in \mathbb{R}^{w \times h \times d \times c}$. On the first timestep, we set $\mathbf{M}^{(0)} = \mathbf{F}^{(0)}_{\text{reg}}$. On later timesteps, our memory update is computed using a running average operation.

Egomotion estimation This module computes the relative 3D rotation and translation between the current camera pose (from timestep t) and the reference pose (from timestep 0) of the latent 3D map (as opposed to consecutive camera poses). This allows us to register all observations to a common coordinate system, while avoiding incremental drift (10). We assume egomotion (relative rotation and translation between camera views) available in this work.

3D-to-2D projection This module "renders" 2D feature maps given a desired viewpoint $V^{(t)}$ by projecting the 3D feature state $\mathbf{M}^{(t)}$. We first orient the state map by resampling the 3D feature map $\mathbf{M}^{(t)}$ into a view-aligned version $\mathbf{M}^{(t)}_{\text{view}}$. Finally, we pass the perspective-transformed tensor through a series of 2D convolutional layers and an LSTM residual decoder, converting it to an RGB image.

C MODEL ARCHITECTURES FOR LANGUAGE CONDITIONED 3D SCENE GENERATION AND 3D REFERENTIAL OBJECT DETECTION

In Figure 4, we show the pipeline for scene generation from language. In Figure 5, we show the pipeline for object detection using metric learning.

D ADDITIONAL EXPERIMENTS

Scene generation conditioned on natural language We show in Figures 6-7 more neural and Blender rendering of scenes predicted from our model, conditioning on parse trees of natural language utterances. We remind the reader that a Blender rendering is computed by using the cross-object relative 3D offsets predicted by our model, and using the generated object 3D feature tensors to retrieve the closest matching meshes from a training set. Our training set is comprised of 100 objects with known 3D bounding boxes, and for each we compute a 3D feature tensor by using the

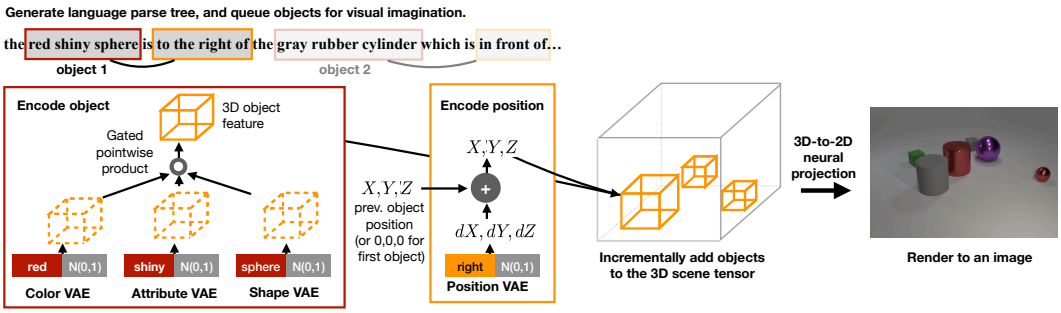

Figure 4: **Mapping natural language to object-centric appearance tensors and cross-object 3D spatial offsets** using conditional variational autoencoders.

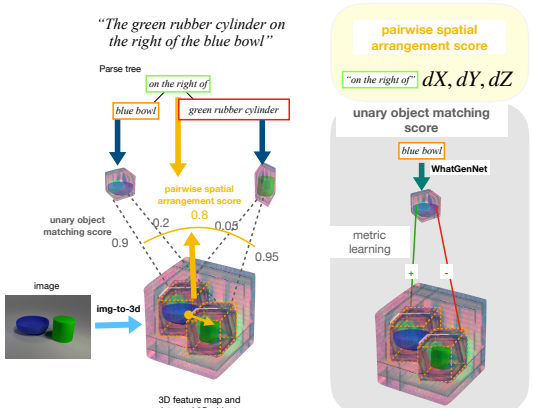

Figure 5: **3D referential object detection** with metric learning between language generated and image generated appearance object 3D feature tensors, and cross object location classifiers.

2D-to-3D unprojection module described above, and cropping the corresponding sub-tensor based on the 3D bounding box coordinates of the object. Despite our neural rendering being blurry, we show the features of our generative networks achieve correct nearest neighbor retrieval.

Scene generation conditional on natural language and visual context In Figures 8-9 we show examples of scene generation from our model when conditioned on both natural language and the visual context of the agent. In this case, some objects mentioned in the natural language utterance are present in the agent's environment, and some are not. Our model uses a 3D object detector to localize objects in the scene, and the learnt 2D-to-3D unprojection neural module to compute a 3D feature tensor for each, by cropping the scene tensor around each object. Then, it compares the object tensors generated from natural language to those generated from the image, and if a feature distance is below a threshold, it grounds the object reference in the parse tree of the utterance to object present in the environment of the agent. If such binding occurs, as is the case for the "green cube" in the top left example, then our model uses the image-generated tensors of the binded objects, instead of the natural language generated ones, to complete the imagination. In this way, our model grounds natural language to both perception and imagination.

Affordability inference based on 3D non-intersection Objects do not intersect in 3D. Our model has a 3D feature generation space and can detect when this basic principle is violated. The baseline model of (11) directly generates 2D images described in the utterances (conditioned on their parse tree) without an intermediate 3D feature space. Thus, it performs such affordability checks in 2D. However, in 2D, objects frequently occlude one another, while they still correspond to an affordable scene. We show in Figure 10 intersection over union scores computed in 3D by our model and in 2D by the baseline. While for our model such scores correlate with affordabilty of the scene (e.g.,

the scenes in 1st, third, and forth columns in the first row are clearly non-affordable as objects interpenetrate) the same score from the baseline is not an indicator of affordability, e.g., the last column in the last row of the figure can in fact be a perfectly valid scene, despite the large IoU score.

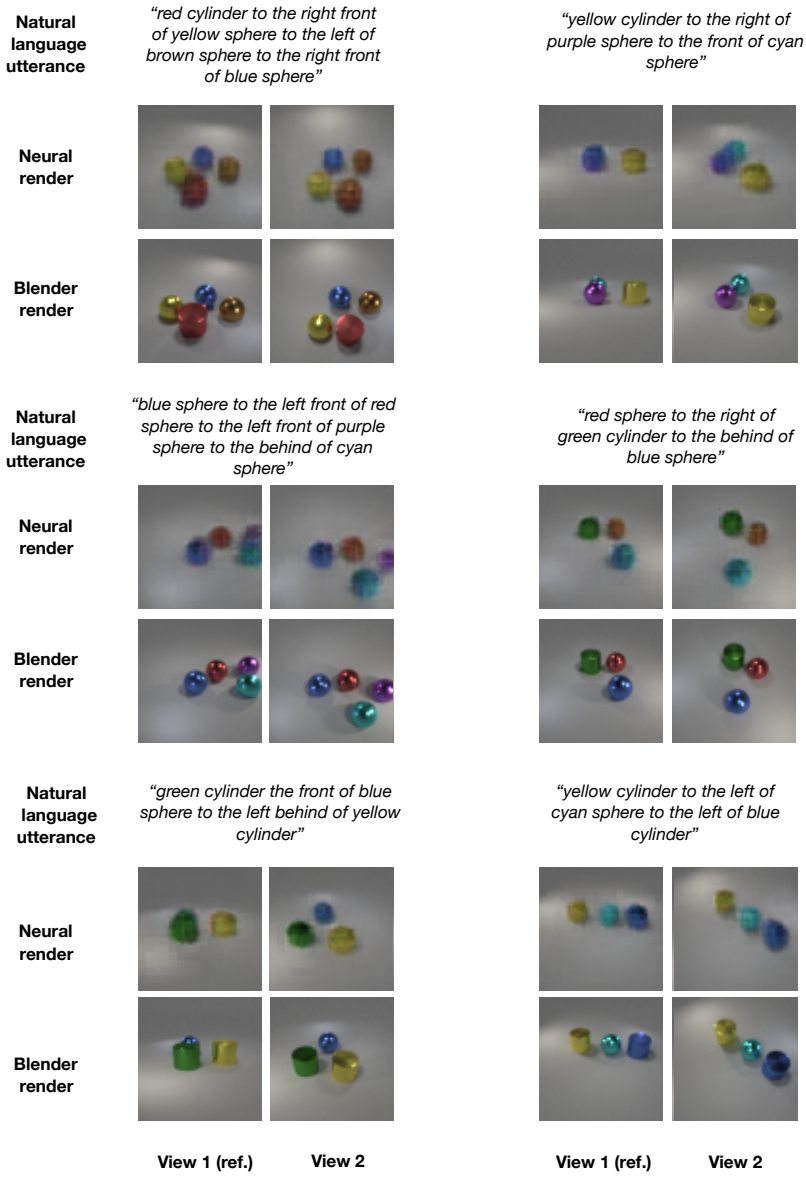

Figure 6: **Natural language conditioned neural and blender scene renderings generated by the proposed model.** We visualize each scene from two nearby views, a unique ability of our model, due to its 3-dimensional generation space.

D.1 Additonal Related Work

Common sense and language understanding The symbol grounding problem (22) states that abstract language symbols do not obtain meaning when grounded in terms of other abstract symbols. For example, the task of reading a passage of text and answering questions about it (53; 25; 30; 13; 43; 52; 26; 36) requires common sense about the world which is not contained in the passage itself. Learning and representing this common sense knowledge is a major research question. Researchers have considered grounding natural language on visual cues as a means of injecting

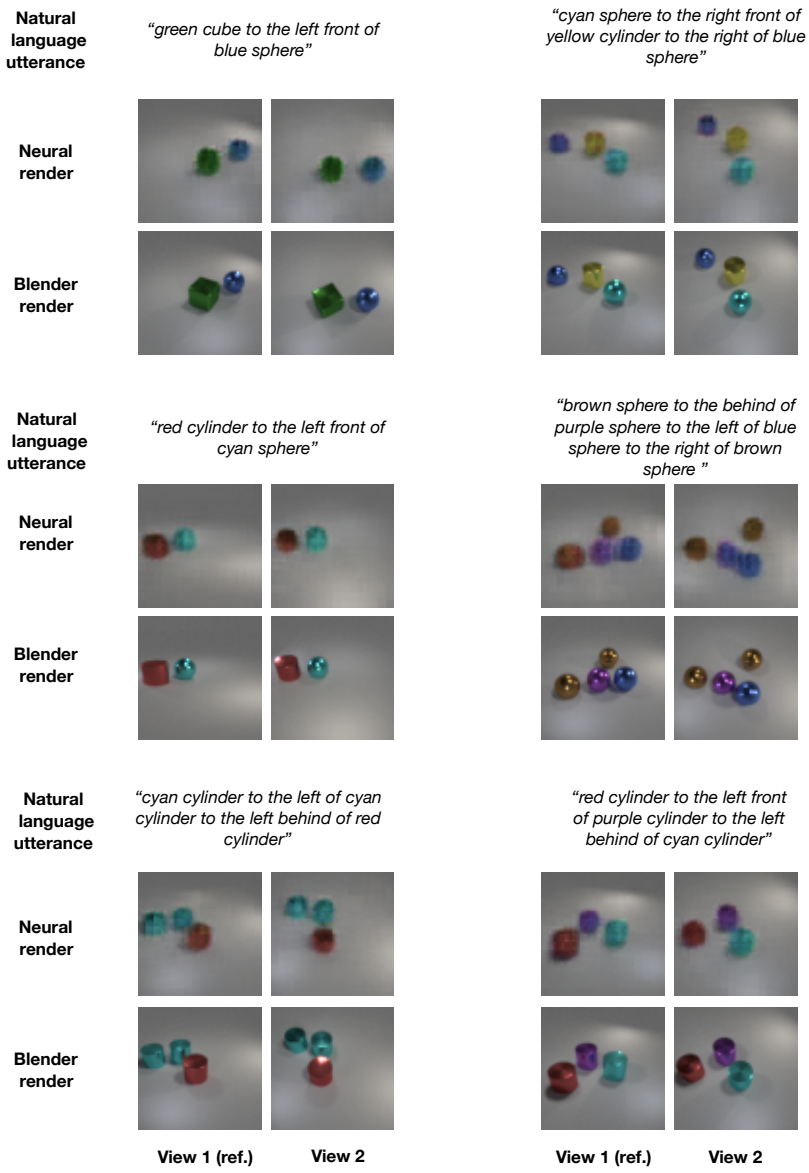

Figure 7: **(Additional) Natural language conditioned neural and blender scene renderings generated by the proposed model.**

common sense knowledge to natural language (41; 15; 41; 15; 3; 12; 2; 40; 39; 38; 32; 54; 14; 11). Yet, there is vast knowledge that current vision and language models miss, regarding basic Physics and Mechanics which are too tedious or obvious to label in datasets, as explained in (51), e.g., to name a few, inanimate objects cannot move on their own, objects that are not supported fall towards the ground, etc. In this paper we point out that 2D boxes and 2D image features cannot be used to reason about affordability of language meaning since even very basic facts, such as object permanence, do not hold in a 2D space. We instead propose associating language to 3D visual feature representations, and show the superior reasoning capabilities that stem out of such 3D grounding space.

Simulation semantics (16; 17; 5; 7) formally states that processing words and sentences leads to perceptual and motor simulations of explicitly and implicitly mentioned aspects of linguistic content,

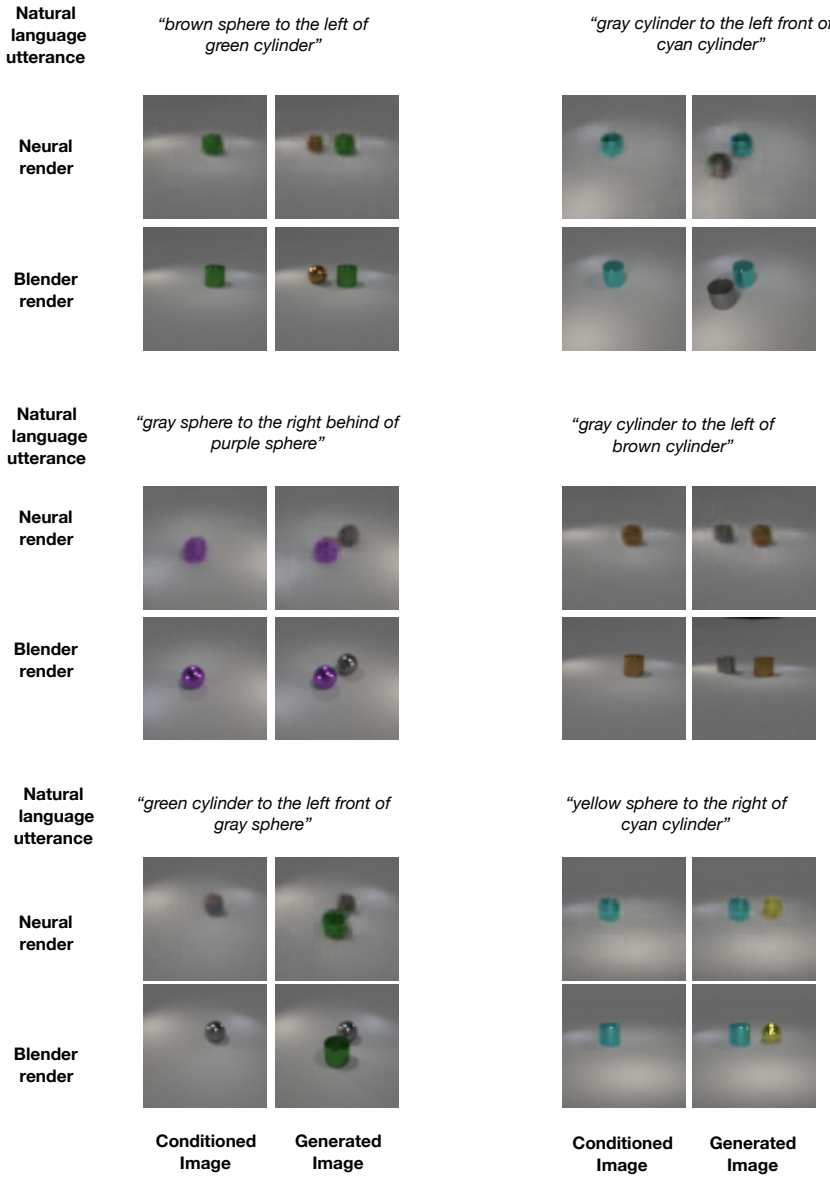

Figure 8: **Neural and blender scene renderings generated by the proposed model, conditioned on natural language *and* the visual scene.** Our model uses a 3D object detector to localize objects in the scene, and the learnt 2D-to-3D unprojection neural module to compute a 3D feature tensor for each, by cropping accordingly the scene tensor. Then, it compares the natural language conditioned generated object tensors to those obtained from the image, and grounds objects references in the parse tree of the utterance to objects presents in the environment of the agent, if the feature distance is below a threshold. If such binding occurs, as is the case for the "green cube" in top left, then, our model used the image-generated tensors of the binded objects, instead of the natural language generated ones, to complete the imagination. In this way, our model grounds natural language to both perception and imagination.

such as verbs and nouns. Currently, it has extensive empirical support: reaction times for visual or motor operations are shorter when human subjects are shown a related sentence (19; 6), and MRI activity is increased in the brain's vision system or motor areas when human subjects are shown

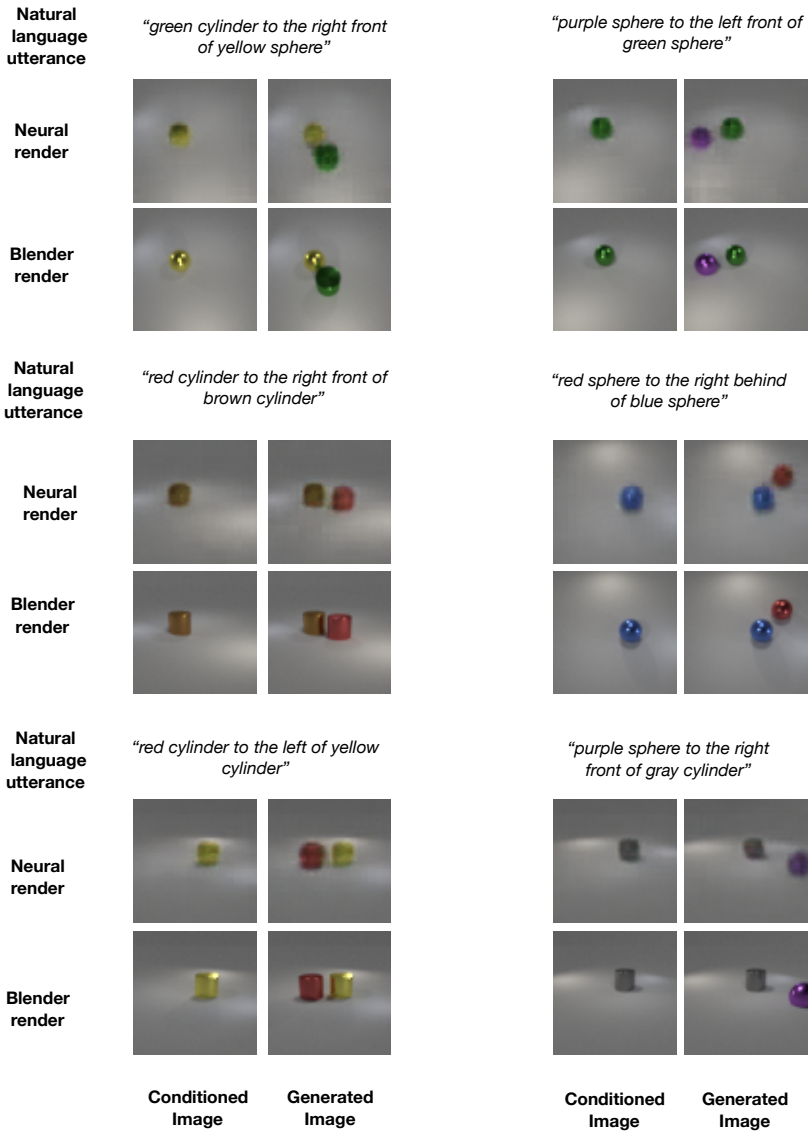

Figure 9: **(Additional) Neural and blender scene renderings generated by the proposed model, conditioned on natural language *and* the visual scene.**

vision- or motor-related linguistic concepts, respectively (4; 34; 42). This paper proposes an initial computational model for the simulation semantics hypothesis for the language domain of object spatial arrangements.

3D representations and feature learning Many recent works have attempted various forms of geometrically-consistent temporal integration of visual information (21; 24; 31; 47), in place of geometry-unaware vanilla LSTM or GRU models. Our work builds upon geometry-aware RNNs (GRNNs) of Tung et al. (50) that learn to integrate images sampled from a viewing sphere into a latent 3D feature memory tensor, in an egomotion-stabilized manner, guided by view prediction: projecting the 3D map from sampled viewpoints and decoding it into corresponding RGB images. To the best of our knowledge this is the first work that associates language with implicit 3D feature representations of objects and scenes.

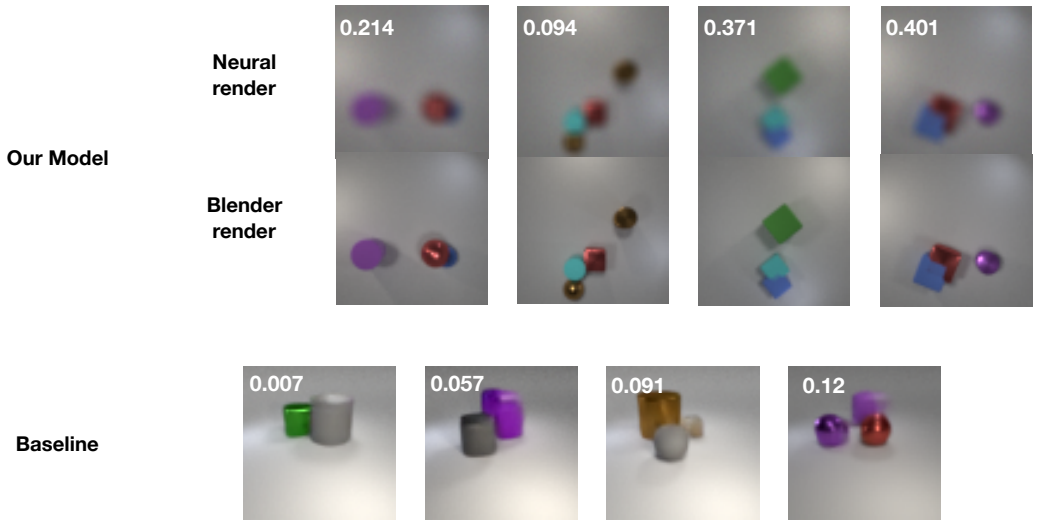

Figure 10: **Affordability prediction comparison of our model with the baseline work of (11).** In the top 2 rows, we show the Neural and Blender renderings of our model. Since we reason about the scene in 3D, our model allows checks for expression affordability by computing the 3D intersection-over-union (IoU) scores. In contrast, the bottom row shows the baseline model which operates in 2D latent space and hence cannot differentiate between 2D occlusions and overlapping objects in 3D.

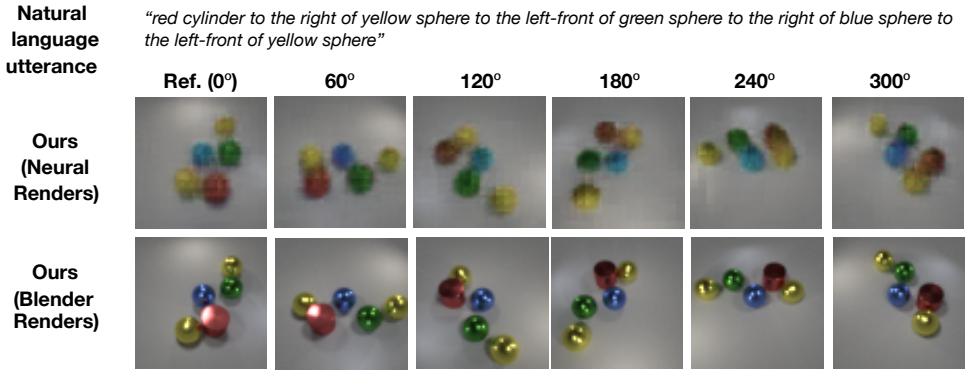

Figure 11: **Consistent scene generation** . We render the generated 3D feature canvas from various viewpoints in the first row using the neural GRNN decoder, and compare against the different viewpoint projected Blender rendered scenes. Indeed, our model correctly predicts occlusions and visibilities of objects from various viewpoints, and can generalize across different number of objects. 2D baselines do not have such imagination capability.

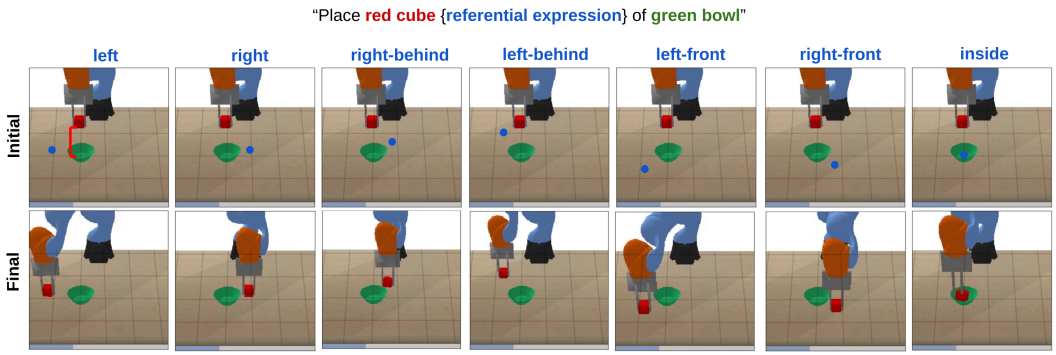

Figure 12: **Language-guided placement policy learning.** Displayed are the final configurations of the learned policy using different language expressions. *Top:* Goals generated with our method. *Bottom:* Goals generated with baseline method. Note that certain baseline configurations that seem correct from the given viewpoint are wrong in terms of depth since the baseline only generates goals on image-level (2D) rather than 3D.