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 reg. 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

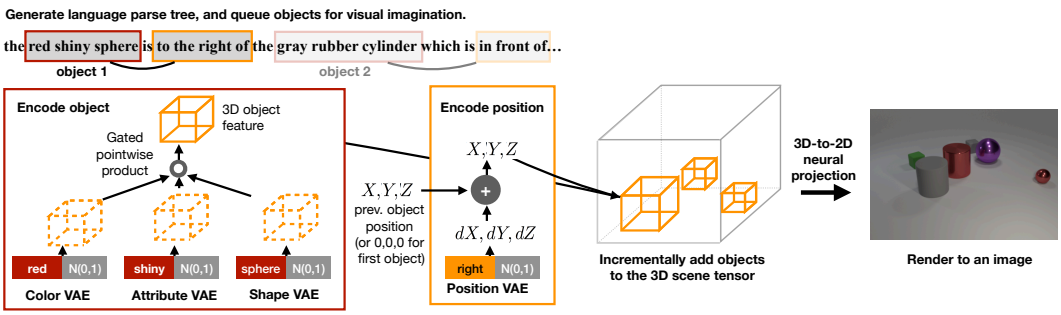

Figure 4: **Mapping natural language to object-centric appearance tensors and cross-object 3D spatial offsets** using conditional variational autoencoders.

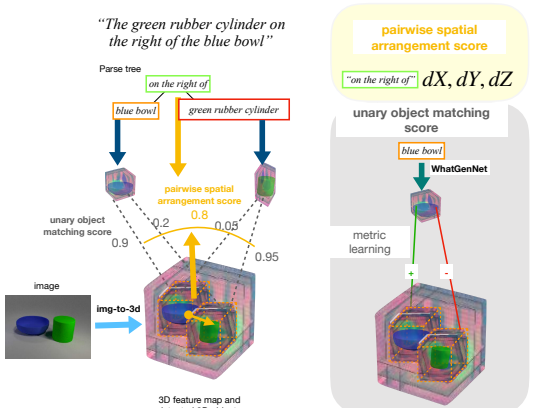

Figure 5: **3D referential object detection** with metric learning between language generated and image generated appearance object 3D feature tensors, and cross object location classifiers.

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.

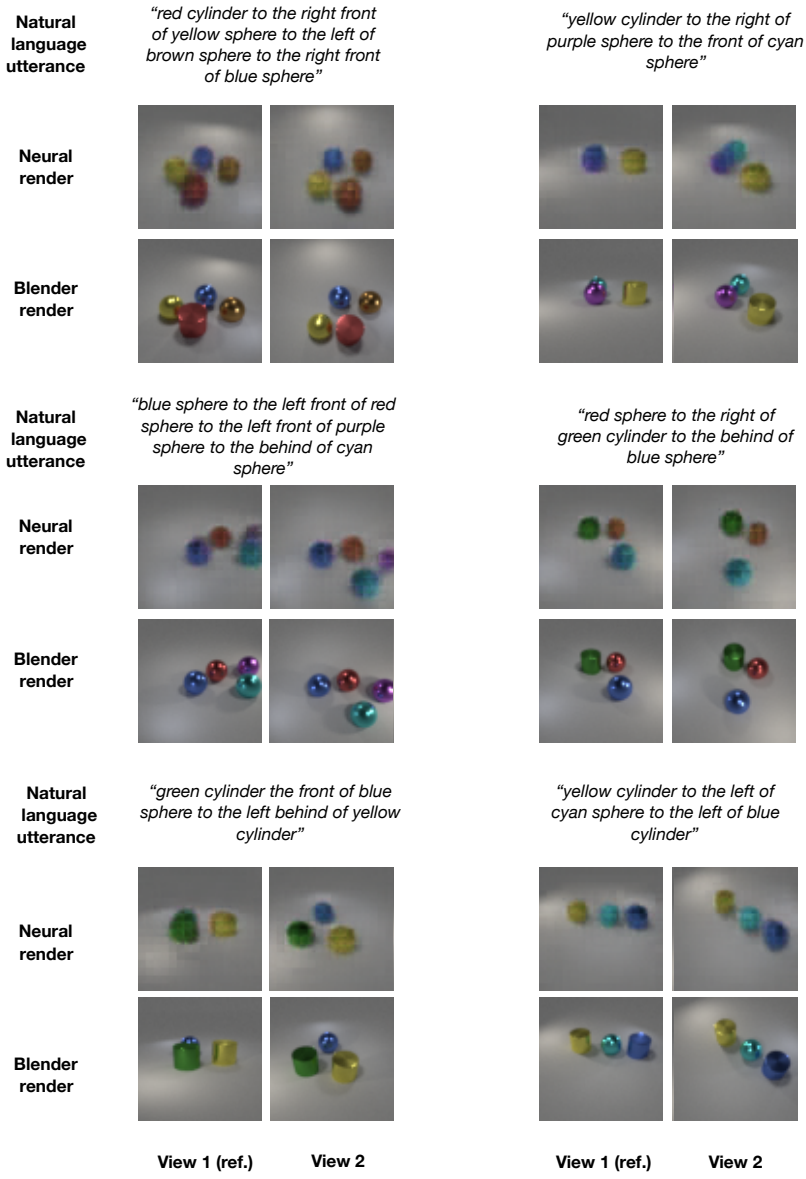

Figure 6: **Natural language conditioned neural and blender scene renderings generated by the proposed model.** We visualize each scene from two nearby views, a unique ability of our model, due to its 3-dimensional generation space.

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 (52; 25; 30; 13; 43; 53; 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

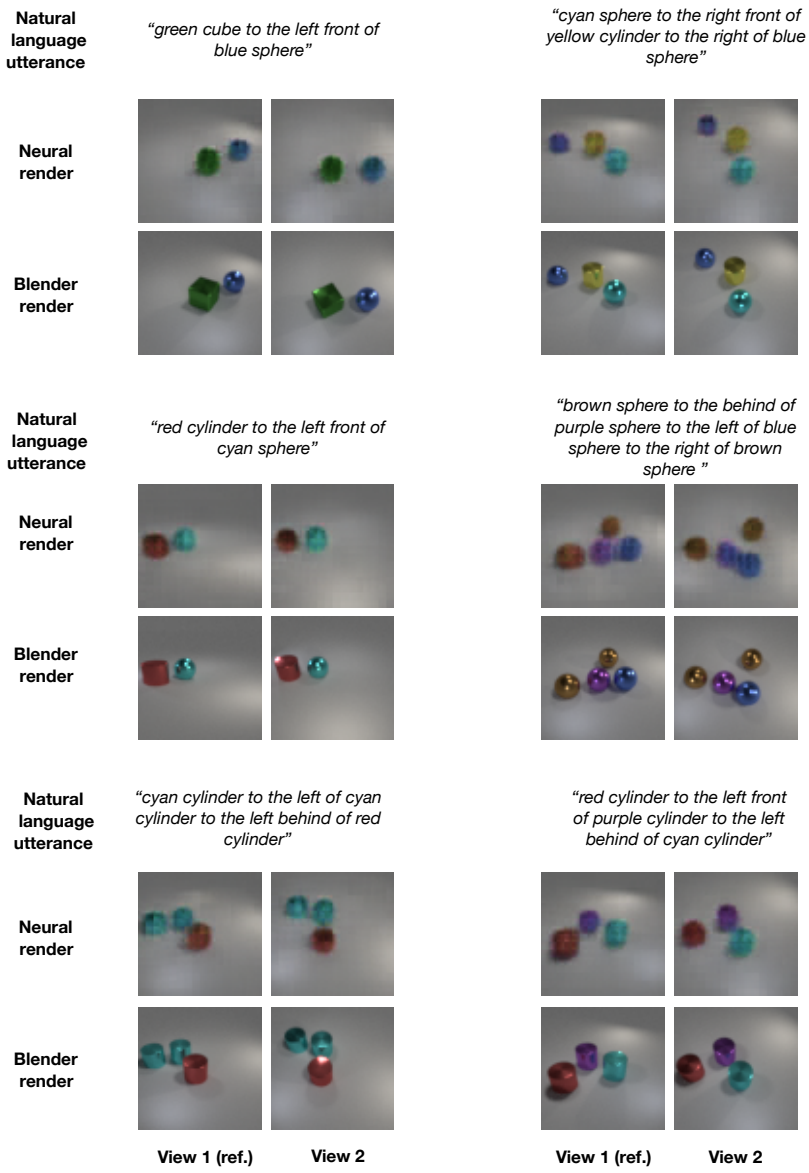

Figure 7: **(Additional) Natural language conditioned neural and blender scene renderings generated by the proposed model.**

common sense knowledge to natural language (41; 15; 41; 15; 2; 12; 1; 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; 6) formally states that processing words and sentences leads to perceptual and motor simulations of explicitly and implicitly mentioned aspects of linguistic content,

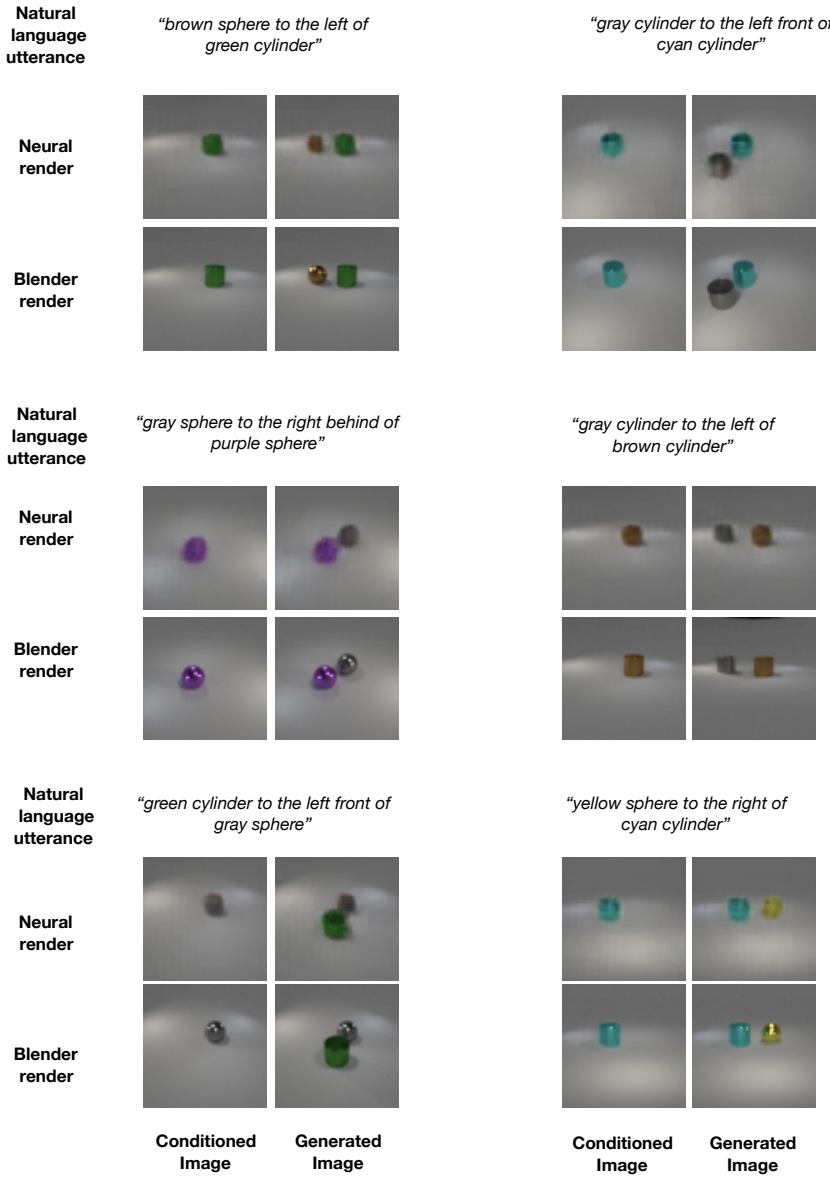

Figure 8: **Neural and blender scene renderings generated by the proposed model, conditioned on natural language *and* the visual scene.** Our model uses a 3D object detector to localize objects in the scene, and the learnt 2D-to-3D unprojection neural module to compute a 3D feature tensor for each, 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 (20; 4), and MRI activity is increased in the brain's vision system or motor areas when human subjects are shown

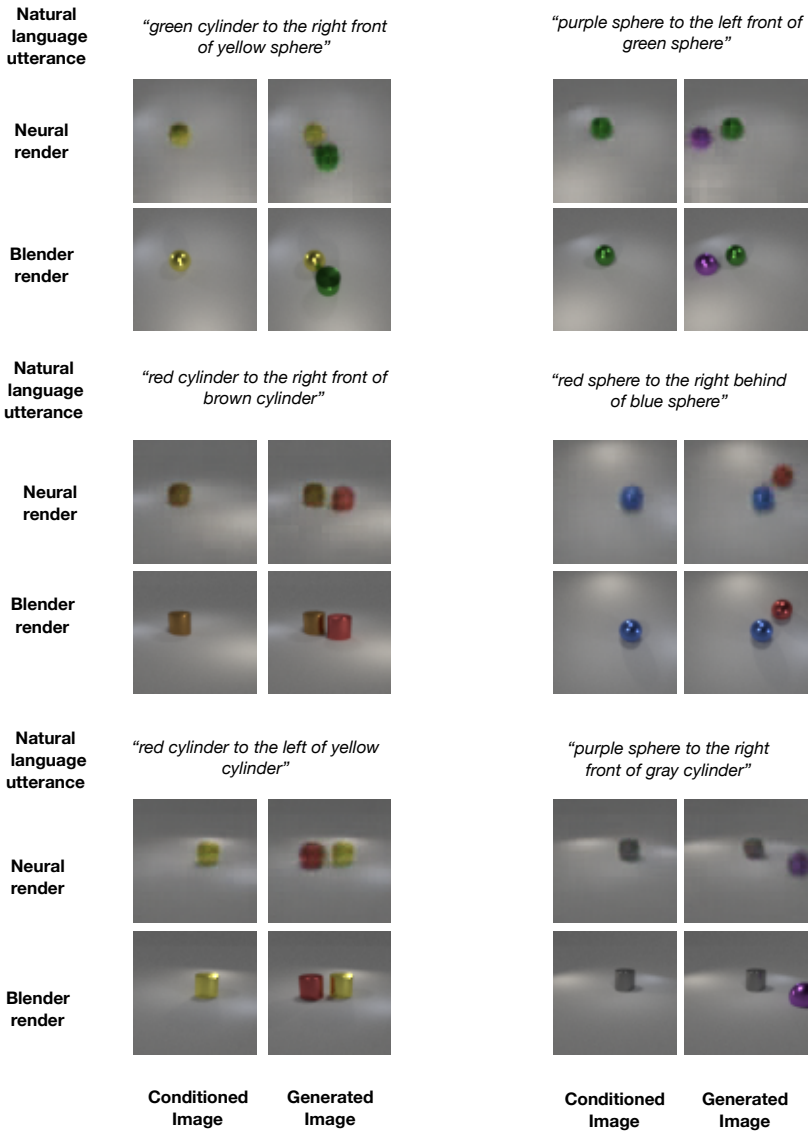

Figure 9: **(Additional) Neural and blender scene renderings generated by the proposed model, conditioned on natural language *and* the visual scene.**

vision- or motor-related linguistic concepts, respectively (3; 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.

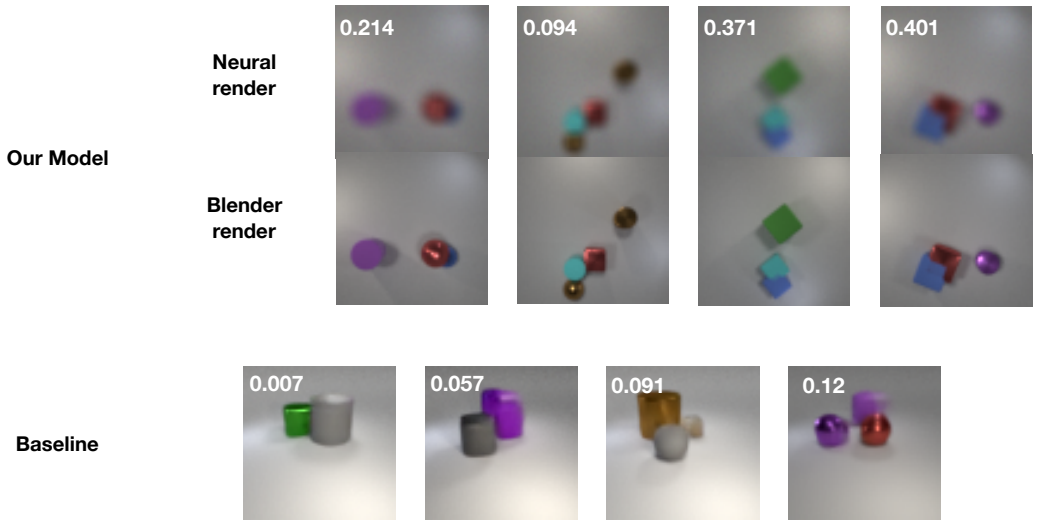

Figure 10: **Affordability prediction comparison of our model with the baseline work of (11).** In the top 2 rows, we show the Neural and Blender renderings of our model. Since we reason about the scene in 3D, our model allows checks for expression affordability by computing the 3D intersection-over-union (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.

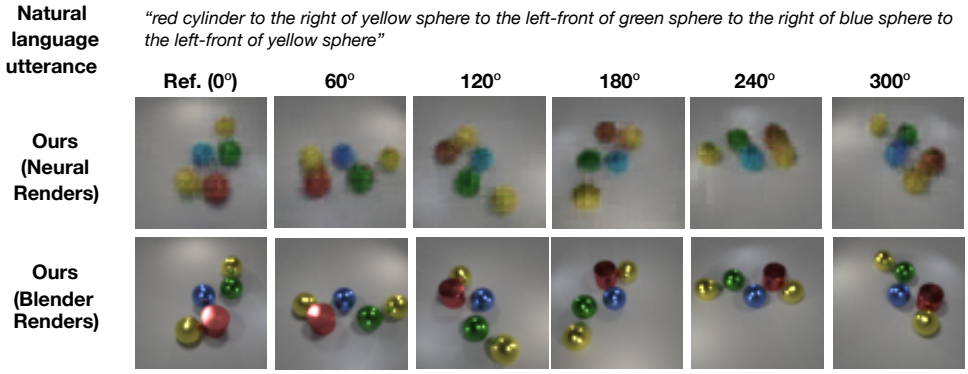

Figure 11: **Consistent scene generation** . We render the generated 3D feature canvas from various viewpoints in the first row using the neural GRNN decoder, and compare against the different viewpoint projected Blender rendered scenes. Indeed, our model correctly predicts occlusions and visibilities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

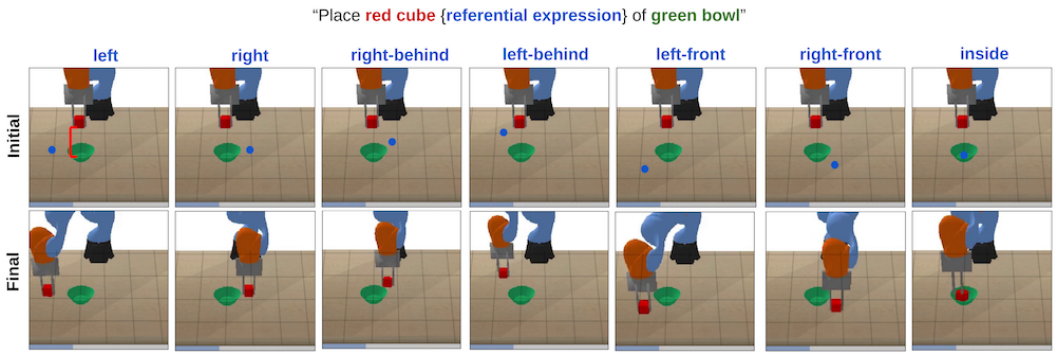

Figure 12: **Language-guided placement policy learning.** Displayed are the final configurations of the learned policy using different language expressions. *Top:* Goals generated with our method. *Bottom:* Goals generated with baseline method. Note that certain baseline configurations that seem correct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