GENERATIVE HIERARCHICAL MODELS FOR PARTS, OBJECTS, AND SCENES

Anonymous authors
Paper under double-blind review

ABSTRACT

Hierarchical structure such as part-whole relationship in objects and scenes are the most inherent structure in natural scenes. Learning such representation via unsupervised learning can provide various benefits such as interpretability, compositionality, and transferability, which are important in many downstream tasks. In this paper, we propose the first hierarchical generative model for learning multiple latent part-whole relationships in a scene. During inference, taking top-down approach, our model infers the representation of more abstract concept (e.g., objects) and then infers that of more specific concepts (e.g., parts) by conditioning on the corresponding abstract concept. This makes the model avoid a difficult problem of routing between parts and whole. In experiments on images containing multiple objects with different shapes and part compositions, we demonstrate that our model can learn the latent hierarchical structure between parts and wholes and generate imaginary scenes.

1 INTRODUCTION

Modeling latent hierarchical structure in natural scenes which consist of objects constructed with several parts is an important problem of unsupervised representation learning. Learning the relationship between parts and whole, this representation can naturally provide interpretability, compositionality, and transferability, for which current deep learning models are not quite successful. Although it may be easy to learn such hierarchical structures using the advanced image segmentation algorithms learned through a supervised learning approach, unsupervised approach should provide more flexibility and generalization ability because the definition of parts and whole can vary depending on situations.

Despite the importance of this problem of unsupervised hierarchical scene structure learning, there have not been many literature in the past. AIR (Eslami et al., 2016) and SPAIR (Crawford & Pineau, 2019) are models that can model multiple objects in a scene but it does not provide hierarchical part-whole relationship. In capsule networks (Sabour et al., 2017; Hinton et al., 2018), the part-whole relationship is used for achieving viewpoint invariance but is not modeled as probabilistic generative model. One difficult challenge of capsule networks is to learn the routing algorithm assigning low level parts to upper level wholes. This is inevitable due to the bottom-up processing.

In this paper, we propose a hierarchical generative model of the structure among scenes, objects, and parts. At each level, the representation contains an appearance component which summarizes lower level composition, and a pose component which specifies how to compose the upper level from the current level. Once learned, the model is able to generate all lower level latents and render a partial scene given the appearance component at a specific level. The inference process is similarly top-down, where we first infer the top level appearance component, and then infer all lower level latents following the same order as the generative process. To further increase interpretability, we assume that the lowest level contains a small number of primitive shapes, and we learn a set of templates that is able to capture all of them.

Importantly, taking top-down approach, our model avoids the challenging problem of routing. In addition, it is also able to deal with objects that contain multiple parts of the same type, and scenes that contain the same object multiple times.

The contributions of the paper are as follows. We propose the first unsupervised hierarchical object-oriented representation learning model through probabilistic latent variable modeling. We then demonstrate in experiments that the proposed model can learn such representations for images containing multiple compositional objects. Further, we showcase other properties such as generalization ability, e.g., w.r.t. the unseen number of objects.

2 Related work

Unsupervised object detection AIR (Eslami et al., 2016) is the first generative model that learns object-wise scene representation. It is able to assign a latent vector $(z^{pres}, \boldsymbol{z}^{where}, \boldsymbol{z}^{what})$ to each object in the scene, describing the presence, size, center position and appearance of the object. SPAIR (Crawford & Pineau, 2019) improves the scalability of AIR to large images containing many objects. It divides the image into spatially distributed cells, and auto-regressively infers the latent vector for each cell. This crucially reduces the search space for individual cells, since they are each responsible only for detecting objects near themselves. Our model builds upon SPAIR to detect objects and parts. To enable efficient hierarchical inference, we replace the sequential cell-by-cell processing in SPAIR with a single feedforward pass of convolutional networks, allowing inference of all cells to be done in parallel.

Hiearchical scene representation Modeling the part-whole relationship in scenes has attracted growing interest, and it has been utilized for improving image classification, parsing and segmentation. In HCM (Zhu et al., 2008), the hierarchical structure is represented as a graph, where leaf nodes interact with image segments, and upper level nodes store the absolute pose (with respect to the whole image) of lower level nodes. In capsule networks (Sabour et al., 2017; Hinton et al., 2018) the part-whole relationship is used for achieving viewpoint invariance. The key insight is that the relative pose of parts with respect to objects is viewpoint invariant, and is thus suitable to be learned as network weights. However, neither of these two approaches uses generative modeling, and they have been applied only to scenes with one dominant object.

3 Model

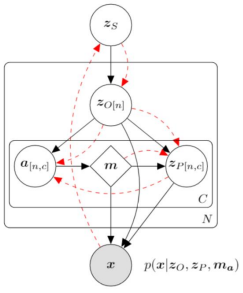

Figure 1: Hierarchical model for parts, objects and scenes. Black solid lines depict generative process, while red dashed lines are for inference process.

Our proposed generative model is a VAE (Kingma & Welling, 2013) with three-level hierarchical latent representation $\boldsymbol{z}_S$, $\boldsymbol{z}_O$, $\boldsymbol{z}_P$ for scenes, objects and parts respectively. $\boldsymbol{z}_O$ is organized into spatially distributed cells similar to SPAIR, and the object in each cell is in turn composed from a spatial grid of parts. The composition is specified by the relative pose of parts with respect to the objects they belong, as represented by $\boldsymbol{z}_P^{pose}$. The objects are then composed similarly into scenes according to $\boldsymbol{z}_O^{pose}$. We have included $\boldsymbol{z}_S$ to summarize the object composition, and $\boldsymbol{z}_O^{appr}$ to summarize part composition. This crucially enables top-down recursive generation. To achieve interpretable part representation, we employ a memory buffer $\boldsymbol{M}$ with variational memory addressing (Bornschein et al., 2017) to learn templates for parts. We now describe our model in detail.

3.1 GENERATIVE MODEL

Our proposed generative model has the factorized form:

$$p(\boldsymbol{x}, \boldsymbol{z}_S, \boldsymbol{z}_O, \boldsymbol{z}_P, \boldsymbol{a}|\boldsymbol{M}) = p(\boldsymbol{z}_S)p(\boldsymbol{z}_O|\boldsymbol{z}_S)p(\boldsymbol{z}_P, \boldsymbol{a}|\boldsymbol{z}_O, \boldsymbol{M})p(\boldsymbol{x}|\boldsymbol{z}_O, \boldsymbol{z}_P, \boldsymbol{m_a}) \tag{1}$$

where $\boldsymbol{a}$ denote memory addresses and $\boldsymbol{m_a}$ the corresponding memory content.

The generative process is illustrated in Figure 1 (black solid line), where we first sample the scene-level representation $\boldsymbol{z}_S$, and then sample object-level representation $\boldsymbol{z}_O$ according to:

$$p(\boldsymbol{z}_O|\boldsymbol{z}_S) = \prod_{n=1}^{N} p(\boldsymbol{z}^{pres}_{O[n]}|\boldsymbol{z}_S) \left(p(\boldsymbol{z}^{pose}_{O[n]}|\boldsymbol{z}_S)p(\boldsymbol{z}^{appr}_{O[n]}|\boldsymbol{z}_S)p(\boldsymbol{z}^{ratio}_{O[n]}|\boldsymbol{z}^{appr}_{O[n]})\right)^{z^{pres}_{O[n]}} \tag{2}$$

where the object representation $\boldsymbol{z}_O$ contains the following variables $\boldsymbol{z}^{pres}_{O[n]}$, $\boldsymbol{z}^{pose}_{O[n]}$, $\boldsymbol{z}^{appr}_{O[n]}$, $\boldsymbol{z}^{ratio}_{O[n]}$, which describe the existence, position, appearance of the nth object and the ratio between width and length in its cropped image.

Given the nth object representation $\boldsymbol{z}_{O[n]}$ and a trained memory buffer , the conditional distribution of the final part-level variables is modeled in the form:

$$p(\boldsymbol{z}_P, \boldsymbol{a}|\boldsymbol{z}_O, \boldsymbol{M}) = \prod_{n=1}^{N}\prod_{c=1}^{C} \left(p(\boldsymbol{z}_{P[n,c]}, \boldsymbol{a}_{[n,c]}|\boldsymbol{z}^{appr}_{O[n]}, \boldsymbol{M})\right)^{z^{pres}_{O[n]}} \tag{3}$$

Likewise, the variables describing the existence, position, appearance of a part is still adopted to compose the part representation $\boldsymbol{z}_P$, and the maximum number of the parts appearing in an object is assumed to be a constant C. Therefore, we extend the distribution in 3 as:

$$\begin{aligned} p(\boldsymbol{z}_{P[n,c]}, \boldsymbol{a}_{[n,c]}|\boldsymbol{z}^{appr}_{O[n]}, \boldsymbol{M}) = p(z^{pres}_{P[n,c]}|\boldsymbol{z}^{appr}_{O[n]}) &\left(p(\boldsymbol{z}^{pose}_{P[n,c]}|\boldsymbol{z}^{appr}_{O[n]})\right)^{z^{pres}_{P[n,c]}} \\ &\left(p(\boldsymbol{a}_{[n,c]}|\boldsymbol{z}^{appr}_{O[n]}, \boldsymbol{M})p(\boldsymbol{z}^{appr}_{P[n,c]}|\boldsymbol{m}_{\boldsymbol{a}_{[n,c]}})\right)^{z^{pres}_{P[n,c]}} \end{aligned} \tag{4}$$

$$\tag{5}$$

Given all the latent variables, we render the image as follows:

$$p(\boldsymbol{x}|\boldsymbol{z}_O, \boldsymbol{z}_P, \boldsymbol{m_a}) = \mathcal{N}(\boldsymbol{\mu}_S, \sigma^2\boldsymbol{I}) \tag{6}$$

$$\boldsymbol{\mu}_S = \sum_{n=1}^{N} z^{pres}_{O[n]} \cdot \mathcal{ST}^{-1}\left(\boldsymbol{\mu}_{O[n]}, \boldsymbol{z}^{pose}_{O[n]}\right) \tag{7}$$

$$\boldsymbol{\mu}_{O[n]} = \sum_{c=1}^{C} z^{pres}_{P[n,c]} \cdot \mathcal{ST}^{-1}(\boldsymbol{\mu}_{P[n,c]}, \boldsymbol{z}^{pose}_{P[n,c]}, \boldsymbol{z}^{ratio}_{O[n]}) \tag{8}$$

$$\boldsymbol{\mu}_{P[n,c]} = \boldsymbol{f}^{dec}_P(\boldsymbol{z}^{appr}_{P[n,c]}, \boldsymbol{m}_{\boldsymbol{a}_{[n,c]}}) \tag{9}$$

where we utilize the memory content $\boldsymbol{m}_{\boldsymbol{a}_{[n,c]}}$ and part appearance representation $\boldsymbol{z}^{appr}_{P[n,c]}$ to first draw the parts, and apply spatial transformers $\mathcal{ST}$ to draw the objects from their parts and draw the images from their objects.

3.2 INFERENCE AND LEARNING

Inference Figure 1 (dashed line) illustrates the inference process proceeding by approximating scene posterior, object posterior and part posterior hierarchically. The overall approximate posterior has the form:

$$q(\boldsymbol{z}_S, \boldsymbol{z}_O, \boldsymbol{z}_P, \boldsymbol{a}|\boldsymbol{M}, \boldsymbol{x}) = q(\boldsymbol{z}_S|\boldsymbol{x})q(\boldsymbol{z}_O|\boldsymbol{z}_S, \boldsymbol{x})q(\boldsymbol{z}_P, \boldsymbol{a}|\boldsymbol{z}_O, \boldsymbol{M}, \boldsymbol{x}) \tag{10}$$

where each level posterior approximation is extended as follows:

$$q(\boldsymbol{z}_O|\boldsymbol{z}_S,\boldsymbol{x}) = \prod_{n=1}^{N} q(z^{pres}_{O[n]}|\boldsymbol{z}_S,\boldsymbol{x})\Big(q(\boldsymbol{z}^{pose}_{O[n]}|\boldsymbol{z}_S,\boldsymbol{x})q(\boldsymbol{z}^{appr}_{O[n]}|\boldsymbol{z}_S,\mathcal{ST}(\boldsymbol{x},\boldsymbol{z}^{pose}_{O[n]})\Big)^{z^{pres}_{O[n]}}$$
$$\Big(q(\boldsymbol{z}^{ratio}_{O[n]}|\boldsymbol{z}^{appr}_{O[n]},\mathcal{ST}(\boldsymbol{x},\boldsymbol{z}^{pose}_{O[n]}))\Big)^{z^{pres}_{O[n]}} \tag{11}$$

$$q(\boldsymbol{z}_P,\boldsymbol{a}|\boldsymbol{z}_O,\boldsymbol{M},\boldsymbol{x}) = \prod_{n=1}^{N}\prod_{c=1}^{C}\Big(q(\boldsymbol{z}_{P[n,c]},\boldsymbol{a}_{[n,c]}|\boldsymbol{z}^{appr}_{O[n]},\boldsymbol{M},\boldsymbol{x}_{O[n]})\Big)^{z^{pres}_{O[n]}} \tag{12}$$

$$q(\boldsymbol{z}_{P[n,c]},\boldsymbol{a}_{[n,c]}|\boldsymbol{z}_{O[n]},\boldsymbol{M},\boldsymbol{x}) = q(z^{pres}_{P[n,c]}|\boldsymbol{z}^{appr}_{O[n]},\boldsymbol{x}_{O[n]})\Big(q(\boldsymbol{z}^{pose}_{P[n,c]}|\boldsymbol{z}^{appr}_{O[n]},\boldsymbol{x}_{O[n]})\Big)^{z^{pres}_{P[n,c]}}$$
$$\Big(q(\boldsymbol{a}_{[n,c]}|\boldsymbol{z}^{appr}_{O[n]},\boldsymbol{M},\boldsymbol{x}_{P[n,c]})\Big)^{z^{pres}_{P[n,c]}}$$
$$\Big(q(\boldsymbol{z}^{appr}_{P[n,c]}|\boldsymbol{m}_{\boldsymbol{a}_{[n,c]}},\boldsymbol{x}_{P[n,c]})\Big)^{z^{pres}_{P[n,c]}} \tag{13}$$

where $\boldsymbol{x}_{O[n]} = \mathcal{ST}(\boldsymbol{x},\boldsymbol{z}^{pose}_{O[n]},\boldsymbol{z}^{ratio}_{O[n]})$ and $\boldsymbol{x}_{P[n,c]} = \mathcal{ST}(\boldsymbol{x}_{O[n]},\boldsymbol{z}^{pose}_{P[n,c]})$ are the glimpses extracted from the input image and the cropped image of the nth object respectively.

Learning We jointly optimize the parameters of our $p(\cdot)$ and $q(\cdot)$ with the variational lower bound:

$$\log p(x|\boldsymbol{M}) \geq \mathop{\mathbb{E}}_{\boldsymbol{a},\boldsymbol{z}\sim q(\cdot|\boldsymbol{M},\boldsymbol{x})}[\log p(\boldsymbol{x},\boldsymbol{z}_S,\boldsymbol{z}_O,\boldsymbol{z}_P,a|\boldsymbol{M}) - \log q(\boldsymbol{z}_S,\boldsymbol{z}_O,\boldsymbol{z}_P,a|\boldsymbol{M},\boldsymbol{x})] \tag{14}$$

4 EXPERIMENTS

4.1 DATASETS

To evaluate our model, we have made two datasets of 2D and 3D scenes respectively. Both datasets contain 128×128 color images, split into 64000 for training, 12800 for validation, and 12800 for testing. They present challenges of (i) multi-pose, variable number of objects and parts, (ii) multiple occurrences of the same type of objects and parts within one scene, and (iii) severe occlusion in 3D scenes.

In making each dataset, we first choose a set of primitive shapes to be the parts, and then construct the objects and scenes by recursively composing these parts. Specifically, we have chosen three shapes as parts, and defined ten types of objects in terms of the identity of the constituent parts and their relative position, scale and orientation. Among these ten types, three contain a single part, another three contain two parts, and the remaining four contain three parts. To construct a scene, we first randomly sample the number of objects (between one and four) and their types, and then instantiate these objects. This means for each object, we choose a random color for each of its parts, apply random scaling (within 10% of object size), and draw it at a random position in the scene. In 2D case, the instantiation process also includes random perturbation of parts and random rotation of the object as a whole. We ensure that different objects have minimal overlap. In 3D case, we use MuJoCo (Todorov et al., 2012) to place the objects on a plane, and then take observations from ten different viewpoints, some of which can lead to severe occlusion.

4.2 SCENE DECOMPOSITION

Our model is able to give interpretable, tree-structured decomposition of scenes into objects and parts. We visualize such decomposition in Figure 2 and Figure 3 for 2D and 3D scenes respectively, where we also show the learned memory templates for parts. Notice that the templates should be in gray scale, but for visualization purposes we have assigned a color to each template. The bounding boxes are drawn on top of the input images, according to the inferred pose of objects and parts with $z^{pres} = 1$. Object bounding boxes are drawn in white, while part bounding boxes are drawn in color to indicate the template chosen for each part.

We find that the templates have learned the appearance of parts at several canonical poses (rotation in 2D and viewpoint in 3D), and our model predicts the pose of parts with respect to these canonical

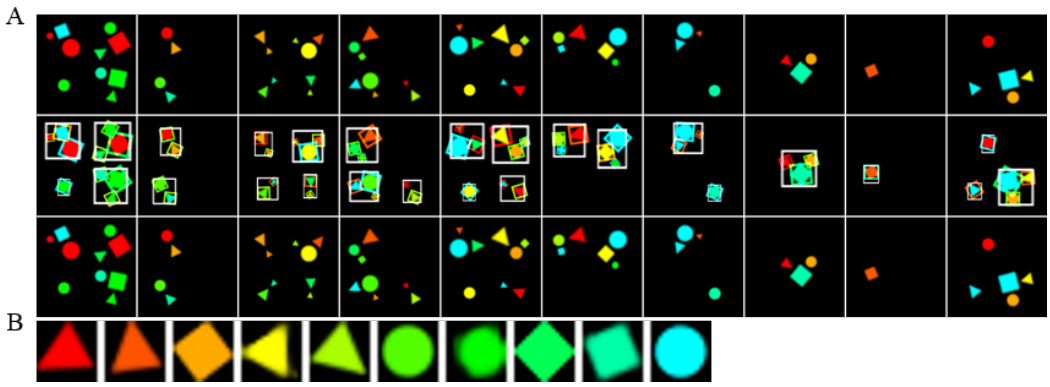

Figure 2: Qualitative results on 2D dataset. *(A) (Top)* Input image. *(Middle)* Input image superimposed with predicted bounding boxes, drawn according to $\boldsymbol{z}^{pres}$, $\boldsymbol{z}^{pose}$, $\boldsymbol{z}_O^{ratio}$ and $\boldsymbol{a}$. *(Bottom)* Reconstruction. *(B)* Learned part-level templates. Template colors indicate identity. Part bounding box colors indicate the chosen template.

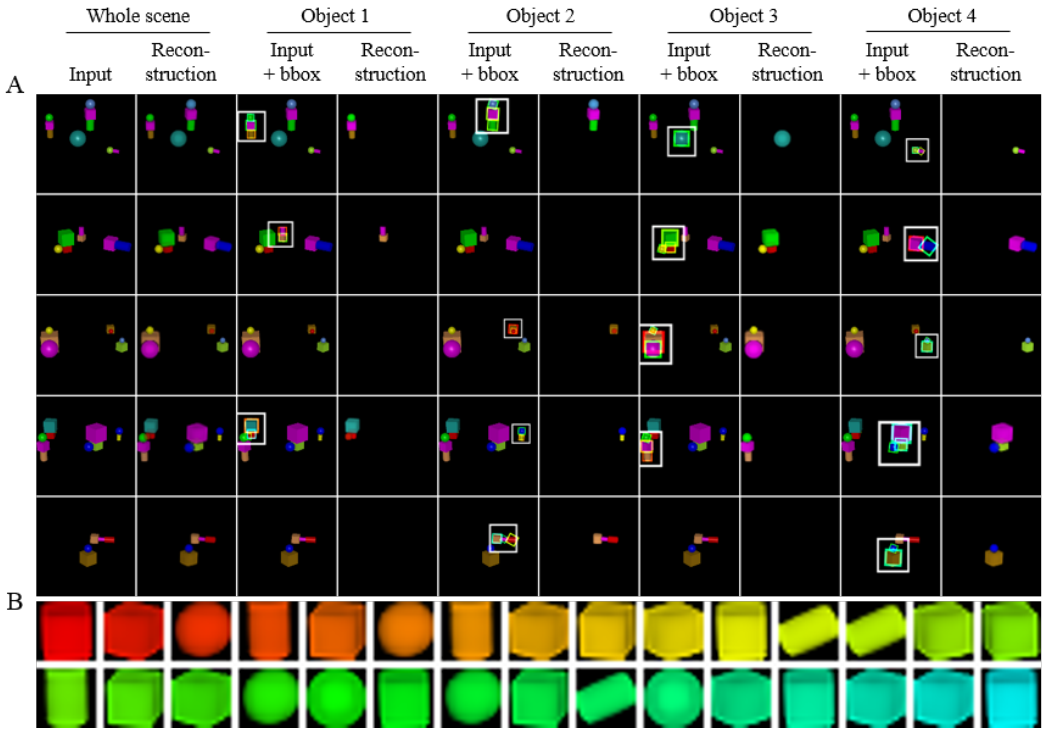

Figure 3: Qualitative results on 3D dataset. *(A)* Each row shows the overall reconstruction, and the predicted bounding boxes and reconstruction from each object cell for a given input image. *(B)* Learned part-level templates. Template colors indicate identity. Part bounding box colors indicate the chosen template.

poses. This makes the decomposition even more interpretable. Moreover, equipped with templates, our model is able to correctly identify the parts even when they have severe occlusion. See Figure 3A third row where a ball occludes an equally sized cube. This example (and many others) also demonstrate that our model can successfully deal with objects composed of multiple parts that are of the same type.

In addition to part-level templates, we believe that the learned object-level $\boldsymbol{z}_O^{appr}$ also helps scene decomposition, especially when there is ambiguity in part assignment and occlusion between ob-

Table 1: Quantitative results on object and part detection.

Dataset	2D dataset			3D dataset		
Training set	1∼4 objects	1&3 objects		1∼4 objects	1&3 objects	
Test set	1∼4 objects	2 objects	4 objects	1∼4 objects	2 objects	4 objects
Object count error	0.00083	0.0014	0.00036	0.094	0.26	0.47
Object precision	0.9985	0.9987	0.9996	0.9639	0.9157	0.9581
Object recall	0.9984	0.9982	0.9995	0.9597	0.9758	0.8462
Part count error	0.0086	0.011	0.014	0.80	1.1	1.3
Part precision	0.9991	0.9988	0.9991	0.8282	0.7579	0.8100
Part recall	0.9989	0.9985	0.9993	0.9116	0.9258	0.8347

Table 2: Comparison on negative log-likelihood

Dataset	2D dataset			3D dataset		
Training set	1∼4 objects	1&3 objects		1∼4 objects	1&3 objects	
Test set	1∼4 objects	2 objects	4 objects	1∼4 objects	2 objects	4 objects
VAE	-13761.9	-13801.4	-13590.5	-13712.0	-13788.3	-13433.2
Ours	**-13890.3**	**-13908.3**	**-13796.6**	**-13818.0**	**-13867.3**	**-13539.7**

jects. For example, in Figure 2A first column, the triangle and circle near the center are close to each other and may well constitute an object. However, because this pose configuration is relatively rare in the training set (compare second column), our model has correctly rejected this composition and instead assigned these two parts to separate objects, which better agrees with the training distribution. In Figure 3A fourth row, object 1 is occluded by object 3. Our model has successfully detected object 1 and added in the reconstruction a ball of the same color as the occluded part. This is quite reasonable since the augmented object is one of our predefined types and appears frequently in the dataset.

To quantify our model's ability of scene decomposition and representation learning, we report absolute counting error, precision and recall for detection of objects and parts in Table 1, and compare the negative log-likelihood of our model with a VAE (Kingma & Welling, 2013) baseline in Table 2. Here the counting error measures the absolute difference between the predicted and true number of objects and parts. To obtain precision and recall, we need to match the predictions with the groundtruth. We set the matching priority as the distance between the predicted and true center positions, namely closer pairs of prediction and groundtruth will be matched first. We only match the pair if their distance is less than 10 pixels (less than half of the size of large parts). This ensures that the matched predictions will have approximately correct center positions. The VAE baseline shares the same scene-level encoder with our model, and uses sub-pixel convolution (Shi et al., 2016) for the decoder. We approximate the negative log-likelihood using 50 importance-weighted samples. The counting error, precision and recall are also averaged over 50 samples from the posterior. As can be seen from Table 1, our model gives almost perfect detection of objects and parts on 2D dataset, and still performs reasonably well on the challenging 3D dataset. We observe that the model tends to split a long cylinder into two parts, leading to the drop in precision for parts.

4.3 Object and scene generation

Apart from learning the part-level templates, our model also has the ability to generate objects and scenes by recursively composing the learned templates. We show generation results in Figure 4. To generate the scenes, we first sample $\boldsymbol{z}_S \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and then sample other latents following the learned conditional prior distributions, and finally use the decoder to render the image. The objects are generated similarly, except that we ignore $\boldsymbol{z}_O^{pres}$ and $\boldsymbol{z}_O^{pose}$. We find that the model has captured

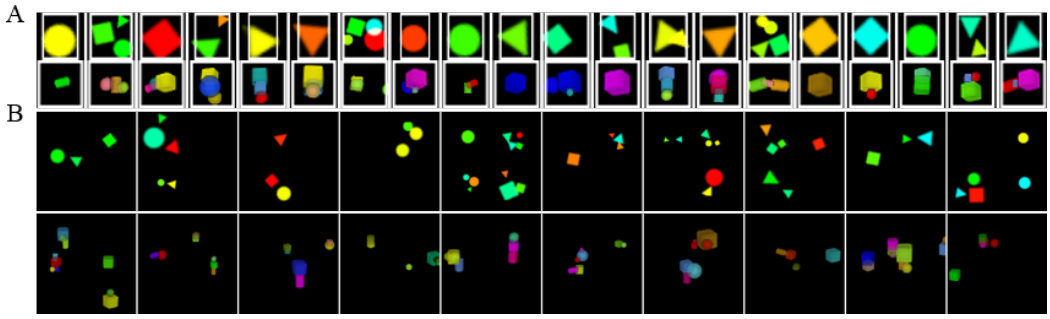

Figure 4: *(A)* Generated objects on *(Top)* 2D dataset and *(Bottom)* 3D dataset. White boxes indicate aspect ratio, and are drawn according to $\boldsymbol{z}_O^{ratio}$. *(B)* Generated scenes on *(Top)* 2D dataset and *(Bottom)* 3D dataset.

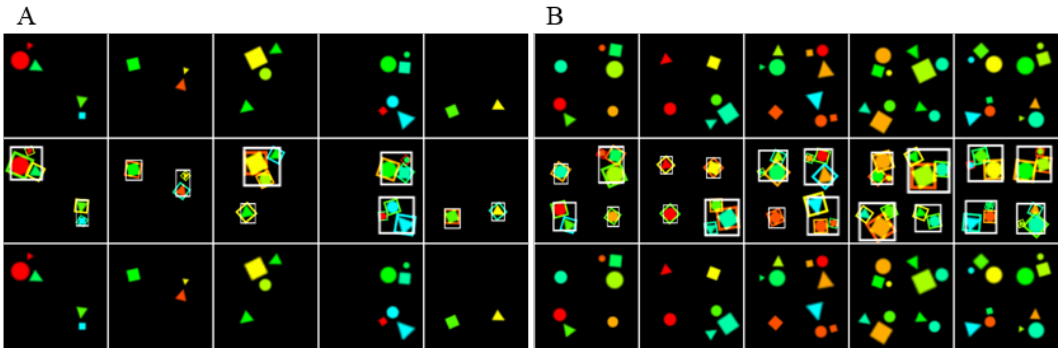

Figure 5: Generalization results on 2D dataset. Model has been trained on scenes with 1 and 3 objects only, and tested on scenes with *(A)* 2 objects and *(B)* 4 objects. *(Top)* Input image. *(Middle)* Input image superimposed with predicted bounding boxes. *(Bottom)* Reconstruction.

many predefined object types in the dataset, and also managed to come up with novel compositions. The generated scenes are also reasonable, with moderate distance and occlusion between objects.

4.4 Generalization performance

Our model represents the scene as composition of objects and parts. This naturally enables generalization to novel scenes. Here we evaluate our model's capacity to generalize to scenes with novel number of objects. The training and validation sets of this task contain scenes of one and three objects only. We trained our model and the VAE baseline again, and report the metrics in Table 1 and Table 2 on two test sets, one having two-object scenes only, and the other having four-object scenes only. We also show qualitative results in Figure 5 and Figure 6. As can be seen, our model demonstrates quite decent generalization performance in both 2D and 3D scenes. We notice that in 3D case, there is a drop in recall when the model is tested on four-object scenes. One reason is that four-object scenes exhibit more severe occlusion than the training set, and we have observed that when two objects are close and have occlusion, the model would sometimes merge them into one object. Another reason is that in four-object scenes, objects are more likely to be partially outside the scene. In this case, the model has difficulty predicting the precise object position, leading to unmatched predictions when we compute the recall.

5 Conclusion

We have proposed the first generative model for learning hierarchical scene representation. During inference, we take top-down approach where high-level representation (e.g., object) is learned first and then the low-level representation (e.g., parts of the objects) are inferred conditioning on the

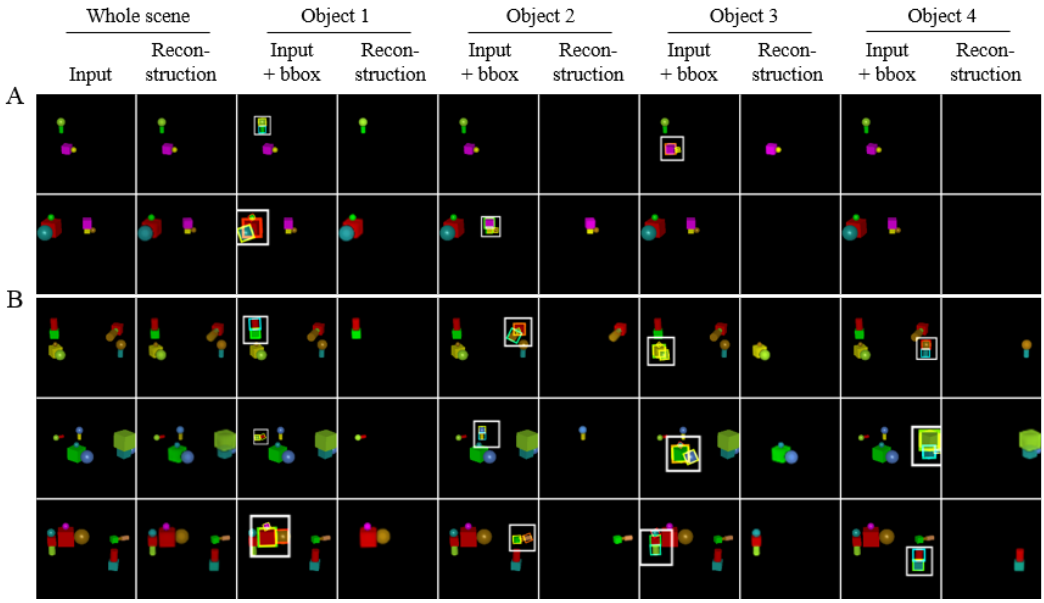

Figure 6: Generalization results on 3D dataset. Model has been trained on scenes with 1 and 3 objects only, and tested on scenes with *(A)* 2 objects and *(B)* 4 objects. Each row shows the overall reconstruction, and the predicted bounding boxes and reconstruction from each object cell for a given input image.

high-level. With this, we can avoid solving difficult routing problem from low-level to high-level. In experiments, we demonstrate that the proposed model can learn such hierarchical structure on images containing multiple compositional objects. An interesting future direction is to extend the proposed model to sequential models.

REFERENCES

Jörg Bornschein, Andriy Mnih, Daniel Zoran, and Danilo Jimenez Rezende. Variational memory addressing in generative models. In *Advances in Neural Information Processing Systems*, pp. 3920–3929, 2017.

Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of AAAI*, 2019.

SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pp. 3225–3233, 2016.

Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pp. 3856–3866, 2017.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1874–1883, 2016.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Long Leo Zhu, Chenxi Lin, Haoda Huang, Yuanhao Chen, and Alan Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *European Conference on Computer Vision*, pp. 759–773. Springer, 2008.