SLOT-VAE: SLOT ATTENTION ENABLES OBJECT-CENTRIC SCENE GENERATION

Yanbo Wang[1], Letao Liu[2], Justin Dauwels[1]
[1] Delft University of Technology, [2] Nanyang Technological University

ABSTRACT

Slot attention has shown remarkable object-centric representation learning performance in computer vision tasks without requiring any supervision. Despite its object-centric binding ability brought by compositional modelling, as a deterministic module, slot attention lacks the ability to generate novel scenes. In this paper, we propose the Slot-VAE, a generative model that integrates slot attention with the hierarchical VAE framework for object-centric structured image generation. From each image, the model simultaneously infers a global scene representation to capture high-level scene structure and object-centric slot representations to embed individual object components. During generation, slot representations are generated from global scene representation to ensure coherent scene structure. Our experiments demonstrate that Slot-VAE achieves better sample quality and scene structure accuracy compared to slot representation-based baselines.

1 INTRODUCTION

Human intelligence is capable of visually segmenting objects out of natural scenes, implicitly learning abstract object concepts, and creatively imagining novel scenes (Yuille & Kersten, 2006) (Frankland & Greene, 2020). Equipping machines with such capabilities has been a desiderata for a long time (Johnson-Laird, 1983) (Schölkopf et al., 2021) (Mambelli et al., 2022). Recent work GNM (Jiang & Ahn, 2020) shows excellent joint object-centric representation learning and image generation performance. Although the hierarchical latent model brings GNM impressive scene structure modeling ability, the bounding box representations in GNM struggle to segment objects with extensively varied scales and are also not flexible enough to model image components of complicated morphology. In contrast, approaches GENESIS (Engelcke et al., 2019) and GENESIS-V2 (Engelcke et al., 2021) adopt more flexible slot representations to model object components. However, the autoregressive prior therein is still unable to capture complex scene structures and the generated samples are very blurry. There is a lack of an object-centric generative model that is able to simultaneously model complex object components and generate structured scenes.

In this work, we propose an object-centric generative model termed Slot-VAE that **integrates slot attention with the hierarchical VAE framework for joint slot representation inference and structured image generation**. Although slot attention (Locatello et al., 2020) has shown very impressive unsupervised segmentation performance, it is unable to generate novel scenes as a deterministic module. If we naïvely combine slot attention with vanilla VAE for multi-object image generation, the generated images would be unreasonable because slots are assumed to be independent and the scene structure (e.g., object relationships) is totally ignored. To overcome this issue, we adopt a two-layer hierarchical VAE model, which **provides both global scene representations that capture the scene structure and object-centric slot representations that characterize individual objects**. Slot representations are generated from global scene representations during the generation stage to ensure coherent scene structure. During training, besides learning from global scene representations, slot representations are additionally regularized by an independent prior to encourage object-centric disentanglement. As a byproduct beyond the generation ability, the VAE framework and independent prior also **empower the slot attention baseline with object attribute-level disentanglement ability**. Evaluating on several multi-object datasets, we show that **Slot-VAE outperforms baselines in terms of sample quality and scene structure accuracy**.

The detailed Introduction and Related Work Section can be found in the Appendix A and B.

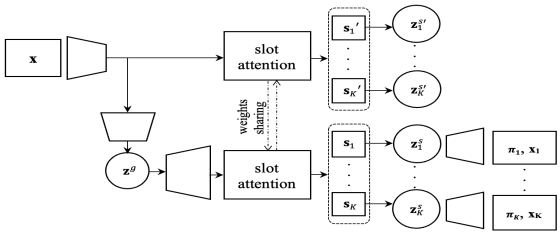

Figure 1: Slot-VAE overview. The image $\mathbf{x}$ is fed into a CNN module. The obtained image features go through two paths in parallel. On the first path, the obtained image features are input into a slot attention module to learn object-centric slot representations $\mathbf{s}'_{1:K}$. From $\mathbf{s}'_{1:K}$, latent vectors $\mathbf{z}^{s'}_{1:K}$ are inferred. On the second path, the obtained image features are encoded into a global latent vector $\mathbf{z}^g$. From $\mathbf{z}^g$, a feature map is built and fed into a slot attention module to generate slot representations $\mathbf{s}_{1:K}$. From $\mathbf{s}_{1:K}$, latent vectors $\mathbf{z}^s_{1:K}$ are obtained. Next, a decoder decodes individual object latent vectors $\mathbf{z}^s_{1:K}$ into object masks $\boldsymbol{\pi}_{1:K}$ and object components $\mathbf{x}_{1:K}$. By combining $\mathbf{x}_{1:K}$ with $\boldsymbol{\pi}_{1:K}$, the input $\mathbf{x}$ is reconstructed. The two paths share the same slot attention module and weights, and it requires $\mathbf{z}^{s'}_k$ and $\mathbf{z}^s_k$ to be as close as possible measured with KL divergence.

2 The Proposed Model: Slot-VAE

2.1 Generation

For an image $\mathbf{x} \in [0,1]^{H\times W\times C}$, we postulate a two-layer hierarchical latent model for the potential image generation process. Specifically, the first-layer latent vector $\mathbf{z}^g \in \mathbb{R}^{L\times 1}$ captures the global structure in the image, for the purpose of modelling relationships among objects. Generated from $\mathbf{z}^g$, the second-layer latent vectors $\{\mathbf{z}^s_k \in \mathbb{R}^{D\times 1}\}_{k=1}^K$ represent each individual object in the image, with the goal of incorporating object-centric slot representations. Finally, with $\mathbf{z}^s_{1:K}$, an image $\mathbf{x}$ can be rendered with a decoder. Mathematically, the complete generative model can be written as:

$$p_\theta(\mathbf{x}) = \iint p_\theta(\mathbf{x} \mid \mathbf{z}^s_{1:K}) p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g) p_\theta(\mathbf{z}^g) d\mathbf{z}^s_{1:K} d\mathbf{z}^g. \tag{1}$$

For the prior of $\mathbf{z}^g$, we can choose a powerful StructDRAW prior Jiang & Ahn (2020) or a simple normal distribution depending on image complexity. To generate $\mathbf{z}^s_{1:K}$, we first decode a feature map $\mathbf{f} \in \mathbb{R}^{H\times W\times D}$ from $\mathbf{z}^g$ and then feed $\mathbf{f}$ to a slot attention module Locatello et al. (2020) to obtain slot representations $\{\mathbf{s}_k \in \mathbb{R}^{D\times 1}\}_{k=1}^K$. Since slot attention is a deterministic module, an additional MLP is needed to map deterministic $\mathbf{s}_{1:K}$ to probabilistic latent vectors $\mathbf{z}^s_{1:K}$. Assuming $\mathbf{z}^s_{1:K}$ are Gaussian and conditionally independent given $\mathbf{z}_g$, we have $p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g) = \prod_{k=1}^K p_\theta(\mathbf{z}^s_k \mid \mathbf{z}^g)$. To render an image $\mathbf{x}$ from $\mathbf{z}^s_{1:K}$, K sub-images $\{\mathbf{x}_k \in [0,1]^{H\times W\times C}\}_{k=1}^K$ are first rendered associated with masks $\boldsymbol{\pi}_{1:K} \in [0,1]^{H\times W}$. Combining $\mathbf{x}_{1:K}$ with $\boldsymbol{\pi}_{1:K}$, the pixel-wise likelihood is written as:

$$p_\theta(\mathbf{x}_{i,j} \mid \mathbf{z}^s_{1:K}) = \mathcal{N}\Big(\big(\sum_{k=1}^K \pi_{i,j,k}(\mathbf{z}^s_{1:K}) \mu_{i,j,k}(\mathbf{z}^s_k) \big), \sigma_x^2 \Big), \tag{2}$$

where (i,j) is the pixel coordinate, σ_x is the standard deviation with a fixed value, and $\pi_{i,j,k}(\cdot)$ and $\mu_{i,j,k}(\cdot)$ are nonlinear functions mapping from latent vectors to masks $\boldsymbol{\pi}_k$ and mean values of $\mathbf{x}_k$ at pixel (i,j). Since $\pi_{i,j,k}$ serves as mixing probability, it requires $\sum_{k=1}^K \pi_{i,j,k} = 1, \forall (i,j)$.

2.2 Inference

Considering that the true posterior is intractable, we approximate the posterior with:

$$p_\theta(\mathbf{z}^g, \mathbf{z}^s_{1:K} \mid \mathbf{x}) \approx q_\phi(\mathbf{z}^g \mid \mathbf{x}) q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}), \tag{3}$$

where $q_\phi(\mathbf{z}^g \mid \mathbf{x})$ is modelled by Gaussian distribution or an auto-regressive model depending on the used prior Jiang & Ahn (2020). As for $q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x})$, we assume conditional independence $q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) = \prod_{k=1}^K q_\phi(\mathbf{z}^s_k \mid \mathbf{x})$, which allows the inference of individual $\mathbf{z}^s_k$ to be performed in parallel, avoiding sequential inference like in GENESIS. The inference of $\mathbf{z}^s_{1:K}$ is achieved with slot attention followed by an MLP (i.e., the first path in Fig. 1), which is detailed in the Appendix C.

2.3 TRAINING

The ELBO $\mathcal{L}(\mathbf{x};\theta,\phi) = \mathbb{E}_{q_\phi(\mathbf{z}^s_{1:K}|\mathbf{x})}\big[\log p_\theta(\mathbf{x} \mid \mathbf{z}^s_{1:K})\big] - D_{\mathrm{KL}}\big[q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g)\big] - D_{\mathrm{KL}}\big[q_\phi(\mathbf{z}^g \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^g)\big]$ is maximized to train the model, where $D_{\mathrm{KL}}(q \mid\mid p)$ is Kullback-Leibler (KL) divergence. Observing the second term on the RHS of ELBO, we can identify a key challenge for the calculation of this KL term: since the slots output by slot attention come with no fixed order, how can we determine the correspondence between $\mathbf{z}^s_{1:K}$ inferred from input $\mathbf{x}$ and $\mathbf{z}^s_{1:K}$ generated from $\mathbf{z}^g$? This issue is very challenging and essential to be solved to train the hierarchical model. **We provide an effective slot order matching solution and elaborate on it** in Appendix D. To further encourage object-centric learning, we also introduce an auxiliary KL term in Appendix E.

3 EXPERIMENTS

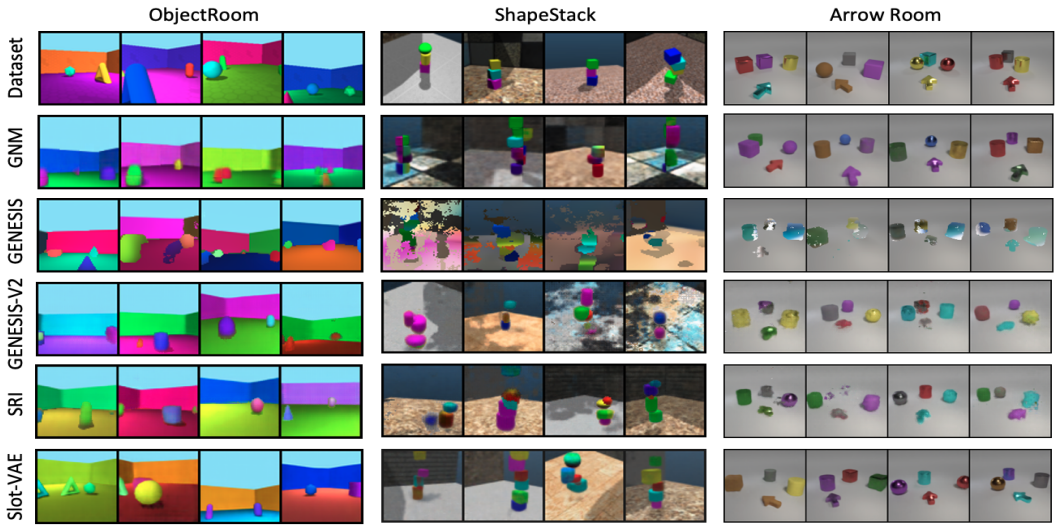

Figure 2: Datasets and generation examples of Slot-VAE and baselines.

Dataset. Three datasets are considered: *ObjectsRoom* (Kabra et al., 2019), *ShapeStacks* (Groth et al., 2018) and *Arrow Room*(Jiang & Ahn, 2020). Among them, *Arrow Room* is less considered by previous works possibly because this dataset is highly structured and its probabilistic density is hard to model. In *Arrow Room*, there is always an arrow shape object in the front and three objects in the back. The arrow always points to the object with a unique shape in the back. We use *Arrow Room* to evaluate the structure accuracy of novel scenes generated by each model.

Baselines. Slot-VAE is compared against four baseline models including GENESIS, GENESIS-V2, SRI and GNM. Among them, GNM is based on the bounding box representations, while the others are based on slot representations and assume an autoregressive prior.

Decomposition Performance. Our experimental results in Appendix F show that Slot-VAE achieves comparable or better object decomposition performance in comparison to other slot representation-based models GENESIS, GENESIS-V2 and SRI. In contrast, the bounding box representation-based model GNM fails to segment *ObjectsRoom* and *ShapeStacks* images because the bounding box representations are not flexible to model complex components. This further limits the generation performance of GNM as described below.

Generation Performance. We show random samples generated by Slot-VAE and baseline models in Fig. 2 (a zoom-in version of Fig. 2 is Fig. 13). For *ObjectRoom*, samples generated by GNM show stripe artifacts due to its inaccurate object-centric representations captured by bounding boxes. The sample quality of SRI is better than that of GENESIS and GENESIS-V2, but not as good as the proposed Slot-VAE. One can more easily identify object shapes (e.g., balls and triangles) and sharp edges with Slot-VAE compared to baselines. For *ShapeStacks*, GNM again shows its limitation where it generates one individual object component with several parts. For example, a cube is

Table 1: FID (↓) and S-Acc (↑) score. Definition of S-Acc can be found in the Appendix G.

	OBJECTSROOM	SHAPESTACKS	ARROW ROOM	
MODEL	FID	FID	FID	S-ACC
GNM	51.6*±5	49.3*±2	11.2±2	0.97
GENESIS	62.8*±3	186.8*±18	173.8±13	0.11
GENESIS-V2	52.6*±3	112.7*±3	111.8±5	0.20
SRI	48.4*±4	70.4*±3	123.3±2	0.18
SLOT-VAE (OURS)	**34.9±1**	**50.0 ± 1**	**60.3±1**	**0.94**

represented by two small parts with completely different colors. Only SRI and Slot-VAE generate reasonable samples, while the sample quality of Slot-VAE is better in terms of sharp object edges. For *Arrow Room*, the highly structured dataset, the samples generated by GENESIS, GENESIS-V2 and SRI are very blurry and seldom show the underlying true scene structure (neither arrow directions nor object shapes are properly learned). In contrast, GNM and Slot-VAE, both exploiting the hierarchical model to capture scene structure, generate very coherent and high-quality samples on *Arrow Room*. The reason why GNM performs better on *Arrow Room* than *ObjectRoom* and *ShapeStacks* is that object shapes are simple in *Arrow Room*. Quantitatively, the FID and S-Acc (Scene Accuracy) score in Table 1 further demonstrates that Slot-VAE outperforms baselines in terms of sample quality and scene structure accuracy. Scores with * in Table 1 are from (Engelcke et al., 2020) and (Emami et al., 2022). Additional random samples can be found in Appendix H.

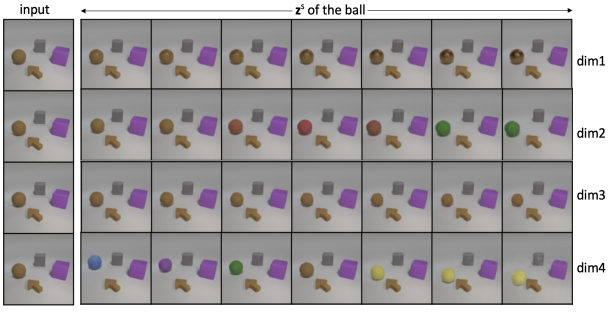

Figure 3: Slot-VAE latent traversal. Each row varies a dimension of $\mathbf{z}^s$ corresponding to the ball.

Controllable Generation. We show controllable scene generation to highlight the disentanglement performance of Slot-VAE. In Fig. 3, in each row we vary a certain dimension of the object-centric latent vector corresponding to the ball object while keeping other object latent vectors unchanged. As is shown, only attributes of the ball are changed, and all other objects remain unaffected. This demonstrates object-level disentanglement of Slot-VAE. Furthermore, attribute-level disentanglement also naturally appears in Slot-VAE. Specifically, when we vary dimension 1, the texture of the ball changes, while when we vary dimension 2, the color of the ball changes. Although some dimensions (e.g., dim 4) entangle color and position, this can be further improved with existing attribute-level disentanglement techniques. Unlike Slot-VAE, the original deterministic slot attention comes with no obvious attribute-level disentanglement as analyzed in (Singh et al., 2022).

4 CONCLUSION

We propose Slot-VAE that integrates slot attention with a hierarchical model for joint object-centric representation inference and scene structure modelling. The proposed model can generate novel scenes controllable at both the object and attribute level. Experiment results show that Slot-VAE achieves state-of-the-art sample quality and scene structure accuracy. One limitation of Slot-VAE is that slot attention requires simple decoders like SBD to serve as a reconstruction bottleneck to decompose objects, which, however, may not scale well to complex real-world scenes. This can be improved by using a transformer as the decoder (Singh et al., 2021), which we leave for future work.

REFERENCES

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3412–3420, 2019.

Fei Deng, Zhuo Zhi, Donghun Lee, and Sungjin Ahn. Generative scene graph networks. In *International Conference on Learning Representations*, 2021.

Sébastien Ehrhardt, Oliver Groth, Aron Monszpart, Martin Engelcke, Ingmar Posner, Niloy Mitra, and Andrea Vedaldi. Relate: Physically plausible multi-object scene synthesis using structured latent spaces. *Advances in Neural Information Processing Systems*, 33:11202–11213, 2020.

Gamaleldin F Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. *arXiv preprint arXiv:2206.07764*, 2022.

Patrick Emami, Pan He, Sanjay Ranka, and Anand Rangarajan. Efficient iterative amortized inference for learning symmetric and disentangled multi-object representations. In *International Conference on Machine Learning*, pp. 2970–2981. PMLR, 2021.

Patrick Emami, Pan He, Sanjay Ranka, and Anand Rangarajan. Slot order matters for compositional scene understanding. *arXiv preprint arXiv:2206.01370*, 2022.

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019.

Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Reconstruction bottlenecks in object-centric generative models. *arXiv preprint arXiv:2007.06245*, 2020.

Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. *Advances in Neural Information Processing Systems*, 34:8085–8094, 2021.

SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in neural information processing systems*, 29, 2016.

Steven M Frankland and Joshua D Greene. Concepts and compositionality: in search of the brain's language of thought. *Annual review of psychology*, 71:273–303, 2020.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *arXiv preprint arXiv:1903.10145*, 2019.

Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. *Advances in Neural Information Processing Systems*, 30, 2017.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pp. 2424–2433. PMLR, 2019.

Oliver Groth, Fabian B Fuchs, Ingmar Posner, and Andrea Vedaldi. Shapestacks: Learning vision-based physical intuition for generalised object stacking. In *Proceedings of the european conference on computer vision (eccv)*, pp. 702–717, 2018.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

Jindong Jiang and Sungjin Ahn. Generative neurosymbolic machines. *Advances in Neural Information Processing Systems*, 33:12572–12582, 2020.

Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. *arXiv preprint arXiv:1910.02384*, 2019.

Philip Nicholas Johnson-Laird. *Mental models: Towards a cognitive science of language, inference, and consciousness*. Number 6. Harvard University Press, 1983.

Rishabh Kabra, Chris Burgess, Loic Matthey, Raphael Lopez Kaufman, Klaus Greff, Malcolm Reynolds, and Alexander Lerchner. Multi-object datasets. https://github.com/deepmind/multi-object-datasets/, 2019.

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

Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021.

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. *Advances in Neural Information Processing Systems*, 31, 2018.

Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5871–5880, 2020.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. *arXiv preprint arXiv:2001.02407*, 2020.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

Davide Mambelli, Frederik Träuble, Stefan Bauer, Bernhard Schölkopf, and Francesco Locatello. Compositional multi-object reinforcement learning with linear relation networks. *arXiv preprint arXiv:2201.13388*, 2022.

Thu H Nguyen-Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. *Advances in Neural Information Processing Systems*, 33:6767–6778, 2020.

Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11453–11464, 2021.

Danilo Jimenez Rezende and Fabio Viola. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. *arXiv preprint arXiv:2209.14860*, 2022.

Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate dall-e learns to compose. In *International Conference on Learning Representations*, 2021.

Gautam Singh, Yeongbin Kim, and Sungjin Ahn. Neural systematic binder, 2022. URL https://arxiv.org/abs/2211.01177.

Sjoerd Van Steenkiste, Karol Kurach, Jürgen Schmidhuber, and Sylvain Gelly. Investigating object compositionality in generative adversarial networks. *Neural Networks*, 130:309–325, 2020.

Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006.

A Detailed Introduction

To equip machines with object-centric learning and imagination abilities in an unsupervised way, most of the recent models resort to the variational autoencoder (VAE) framework (Kingma & Welling, 2013) (Rezende et al., 2014) for the purpose of joint object-centric representation inference and image generation. Depending on how to model the compositionality of images, existing works can be roughly categorized as spatial attention-based generative models and scene-mixture-based generative models .

Spatial attention-based generative models infer object-centric representations by extracting a bounding box for each individual object (Eslami et al., 2016) (Crawford & Pineau, 2019) (Lin et al., 2020) (Jiang et al., 2019) (Jiang & Ahn, 2020). Such bounding boxes explicitly represent the position and size of object components enabling interpretable object manipulation. However, this type of model was pointed out to struggle with segmenting objects with extensively varied scales because the size of objects is to some extent presumed (Engelcke et al., 2021) (Emami et al., 2022). Moreover, rectangular bounding boxes are also not flexible enough to model image components of complicated morphology (Lin et al., 2020). In contrast, scene-mixture generative models decompose a visual scene into image-sized components (also known as slots), and infer slot representations corresponding to individual objects (Burgess et al., 2019) (Greff et al., 2019) (Engelcke et al., 2019) (Engelcke et al., 2021). Such models segment objects with masks and are flexible enough to capture complex object components. Recent advances in scene-mixture models have shown remarkable object segmentation performance (Engelcke et al., 2019) (Engelcke et al., 2021). However, although the design of such models advocates autoregressive priors for the purpose of generating coherent scenes, they are still unable to model object relationships in highly structured images and the generated samples are very blurry.

B Related Works

Object-Centric Generative Modelling. Compositional image modelling approaches (Greff et al., 2017) (Greff et al., 2017) (Kosiorek et al., 2018) (Crawford & Pineau, 2019) (Burgess et al., 2019) (Greff et al., 2019)(Lin et al., 2020) (Locatello et al., 2020) (Emami et al., 2021) (Singh et al., 2021) (Kipf et al., 2021) Seitzer et al. (2022) (Singh et al., 2022) (Elsayed et al., 2022) typically incorporates object locality as inductive bias or exploits simple decoder networks as reconstruction bottlenecks (Engelcke et al., 2020) to achieve object-centric disentanglement. However, these approaches, unlike ours, cannot generate coherent novel scenes. GENESIS and GENESIS-V2 (Engelcke et al., 2019) (Engelcke et al., 2021) adopt autoregressive prior for coherent scene generation, but unlike ours, they lack the scene-level representation learning ability and generate blurry samples without accurate scene structure. GNM (Jiang & Ahn, 2020) and similarly (Deng et al., 2021) resorts to a hierarchical VAE model for both distributed and symbolic representations learning, but the bounding box representations therein prevent them from modelling complicated objects or backgrounds, unlike ours where more flexible slot representation is used. SRI Emami et al. (2022) learns slot representations and scene-level representations, but it has to sequentially infer object representations due to the assumed autoregressive posterior. In contrast, our approach poses an independent prior on slot representations allowing parallel inference. Besides, our approach trains the model without the need to learn a fixed object order, but SRI requires specialized auxiliary loss for object order alignment so as to learn the model. Lastly, SRI infers object-centric representations based on GENESIS-V2, while the proposed Slot-VAE exploits slot attention.

GANs for Compositional Generation: GANs-based methods (Van Steenkiste et al., 2020) (Nguyen-Phuoc et al., 2020) (Liao et al., 2020) (Niemeyer & Geiger, 2021) (Ehrhardt et al., 2020) are able to map independent random noise vectors to individual object components on images allowing object-level controllability, but these models lack an inference process and thus cannot edit a given image unlike ours. Meanwhile, these GANs models share common unstable training issues.

C Slot Representation Inference

We adopt slot attention (Locatello et al., 2020) followed by an MLP to infer object-centric slot representations $\mathbf{z}^s_{1:K}$, which is detailed as follows.

CNN for feature extraction. Instead of directly working in the pixel domain, the slot representation inference starts from passing the input image $\mathbf{x}$ through a CNN backbone to extract a feature map $\mathbf{f}_x = f_{enc}(\mathbf{x}) \in \mathbb{R}^{H \times W \times D}$, where the CNN backbone is augmented with positional embeddings.

Slot attention for component discovery. To discover object components, the feature map $\mathbf{f}_x$ is first flattened into vectors $\mathbf{f}_{input} \in \mathbb{R}^{(H \times W) \times D}$. Then, $\mathbf{f}_{input}$ is mapped to K object slots $\mathbf{s}_{1:K}$ with a slot attention module.

MLP for latent vector inference. From slots $\mathbf{s}_{1:K}$, we would like to infer the latent variables $\mathbf{z}^s_{1:K}$. We assume the approximate posterior distribution of each individual slot $q_\phi(\mathbf{z}^s_k \mid \mathbf{x})$ to be Gaussian. Hence, inferring $\mathbf{z}^s_k$ is equivalent to infer Gaussian parameters $\{(\mu^s_k, \sigma^s_k)\}^K_{k=1}$. To that end, we use an MLP shared across objects mapping from slots to Gaussian means and variances: $(\mu^s_k, \sigma^s_k) := \text{MLP}(\mathbf{s}_k)$.

D Slot Order Matching

Problem: how can we determine the correspondence between $\mathbf{z}^s_{1:K}$ inferred from input $\mathbf{x}$ and $\mathbf{z}^s_{1:K}$ generated from $\mathbf{z}^g$?

This issue does not appear in GNM because the spatial attention module therein provides fixed order for each object component, which makes the calculation of KL divergence in GNM possible. SRI proposes to learn a fixed order with a complicated specific auxiliary loss, which does not improve scene generation performance a lot.

Solution: to address the slot order matching issue in Slot-VAE, we propose to implement $q_\phi(\mathbf{z}^s_k \mid \mathbf{x})$ and $p_\theta(\mathbf{z}^s_k \mid \mathbf{z}^g)$ with a shared slot attention module. That is to say, as shown in Fig. 1, the two slot attention modules share parameters. Meanwhile, slots $\mathbf{s}^{'}_k$ and $\mathbf{s}_k$ in Fig. 1 share initialization values. Intuitively, such an architecture design encourages the feature map $\mathbf{f}$ generated from $\mathbf{z}_g$ to be consistent with the feature map $\mathbf{f}_x$ encoded from input $\mathbf{x}$. With similar inputs and the same random initialization values, we can expect the output of the two slot attention modules could keep close to each other. As a result, the order of $\mathbf{s}_k$ (or $\mathbf{z}^s_k$) can have a good chance to align well with that of $\mathbf{s}^{'}_k$ (or $\mathbf{z}^{s\prime}_k$) in Fig. 1, facilitating the calculation of $D_{\text{KL}}\big[q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g)\big]$.

E Auxiliary Loss

Observe again the ELBO $\mathcal{L}(\mathbf{x}; \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}^s_{1:K}|\mathbf{x})}\big[\log p_\theta(\mathbf{x} \mid \mathbf{z}^s_{1:K})\big] - D_{\text{KL}}\big[q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g)\big] - D_{\text{KL}}\big[q_\phi(\mathbf{z}^g \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^g)\big]$. Since $p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g)$ in the second term of the ELBO is learned from the posterior distribution $p_\theta(\mathbf{z}_g \mid \mathbf{x})$, it provides no explicit prior information to guide the learning of the posterior distribution $q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x})$ during training. To guide the learning of $q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x})$, the following auxiliary loss could be incorporated:

$$\mathcal{L}_{aux} = -D_{\text{KL}}\big[q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) \mid\mid \prod_{k=1}^{K} \mathcal{N}(\mathbf{0}, \mathbf{1})\big], \tag{4}$$

where independent normal prior constrains $\mathbf{z}^s_{1:K}$ to be independent on each other. Such a prior encourages each slot representation to capture only a single object leading to object-centric disentanglement. Meanwhile, attribute-level disentanglement within an object can also be achieved due to diagonal variance of the normal prior.

The overall objective function for training Slot-VAE is:

$$\tilde{\mathcal{L}} = \mathcal{L} + \mathcal{L}_{aux}. \tag{5}$$

For effective training, we also introduce hyperparameters to balance the reconstruction loss and KL terms (Rezende & Viola, 2018) (Fu et al., 2019).

F Image Decomposition Performance

Decomposition and Reconstruction Performance. We illustrate the input, reconstruction and decomposed object components of Slot-VAE and baselines in Fig. 4 - 6. Note that GNM infers

bounding box representations instead of slot representations. So in the figures, GNM only has two components, one for the foreground with bounding boxes and another for the background.

As shown in Fig. 4, for the *ObjectRoom* dataset that comes with simple object shapes and complex background components, slot representation-based models GENESIS, GENESIS-V2, SRI and Slot-VAE achieve comparable decomposition and reconstruction performance, The only difference is that some of them capture the background with one slot while others use multiple slots. In contrast, the bounding box representation-based model GNM fails to segment objects correctly. It segments the scene into stripes containing parts of objects and parts of the background, and a single object is segmented into multiple bounding boxes. As a result, the reconstructed images of GNM show rectangular artifacts. This is not surprising because with the use of grid sampling and bounding box representations, spatial-attention generative models like GNM struggle with modelling objects that have complicated morphology. In Fig. 5, we observe similar results for the *ShapeStacks* dataset, where GNM again tries to model one single object with multiple bounding boxes. Failing to learn correct obeject-centric representations, GNM will also suffer during the generation stage as shown in generation results. For *Arrow Room* dataset that has simple object shapes but complicated scene structures in Fig. 6, we can see all models successfully segment objects out of the scene and reconstruct the input image. However, GENESIS-V2 and SRI learn object representations that severely involve part of the background. Such representations will make the generated image samples very blurry, as will be shown below. We conjecture this is because the *Arrow Room* dataset has too strong object position relationships, and GENESIS-V2 and SRI (based on GENESIS-V2) do not have enough capacity and have to choose simple ways to segment images. In summary, Slot-VAE achieves either better or comparable segmentation and reconstruction performance in comparison to baselines. Additional decomposition results of Slot-VAE can be found in the Appendix I.

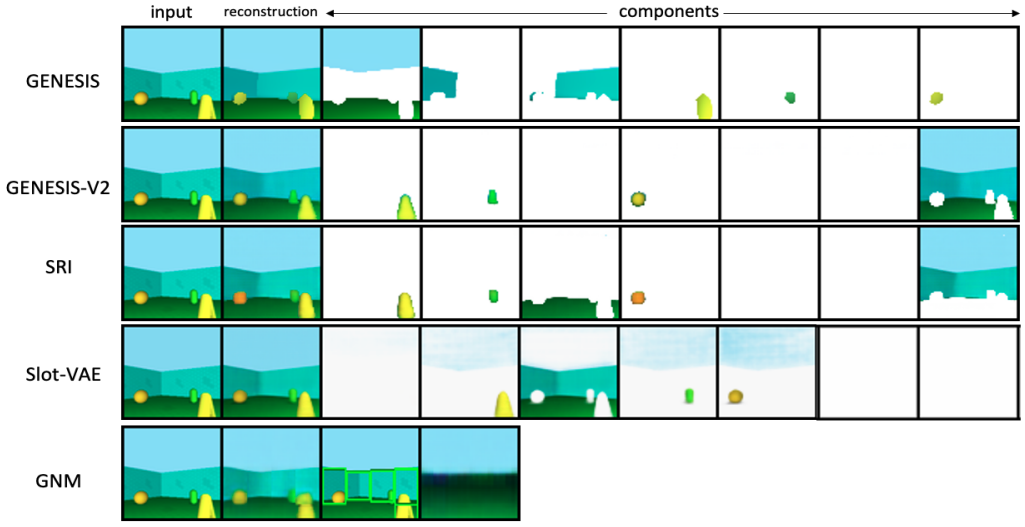

Figure 4: Image decompostion and reconstruction performance on the ObjectsRoom dataset.

Besides, we calculate the Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) score to quantitatively evaluate the decomposition performance. Since the *Arrow Room* dataset comes with no ground truth masks, ARI score on this dataset is not calculated. As shown in Table 2, slot-VAE achieves comparable ARI-FG scores to baselines.

G Quantitative Generation Performance

We report the Frechet Inception Distance (FID) (Heusel et al., 2017) score and scene structure accuracy (S-Acc) (Jiang & Ahn, 2020) score to quantitatively evaluate sample quality and scene structure accuracy. For the FID score, the calculation involves 10000 real and generated samples. Table 1 reflects non-trivial FID score improvement (at least 27% on all datasets and can reach 45%) by Slot-VAE against baselines, highlighting the sample quality of Slot-VAE. Note that the FID score of baselines can be found in (Engelcke et al., 2021) (Emami et al., 2022).

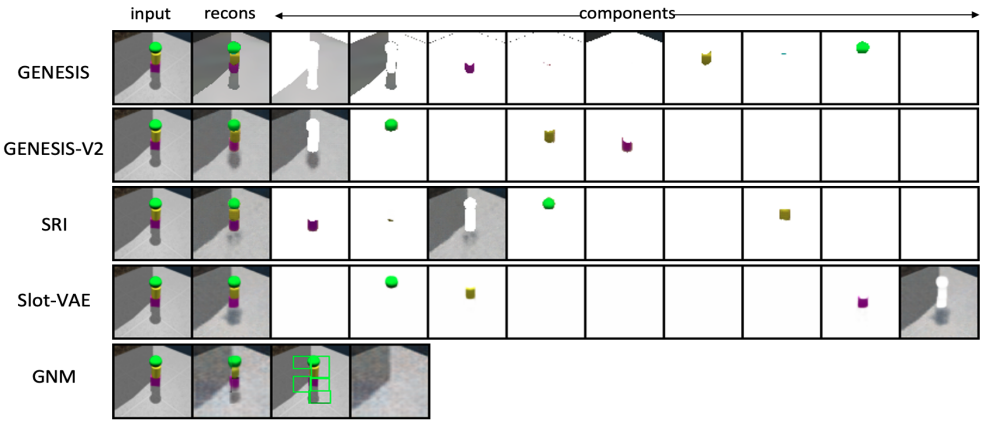

Figure 5: Image decompostion and reconstruction performance on the ShapeStacks dataset.

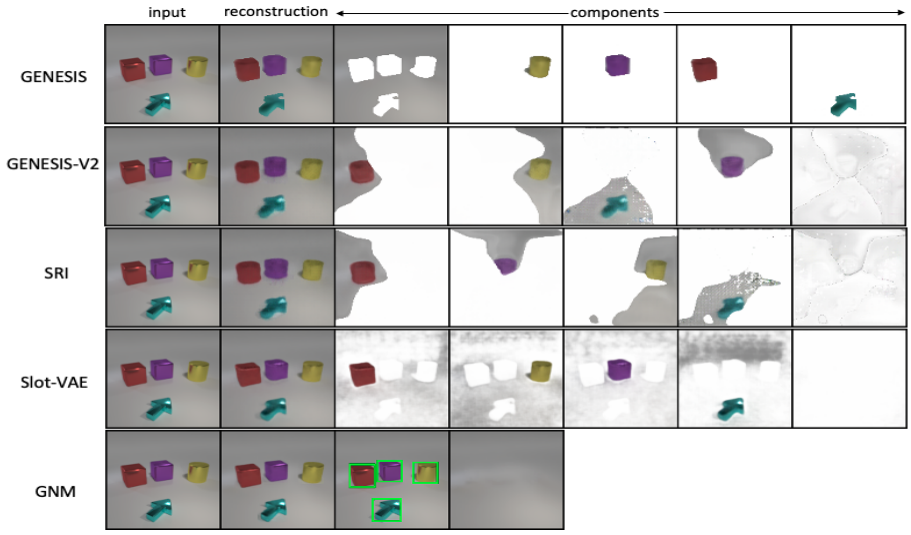

Figure 6: Image decompostion and reconstruction performance on the Arrow Room dataset.

For the S-Acc score, we manually classified 100 generated images per model, and calculate the ratio of successful images that correctly reflect scene structure(i.e., the arrow object should always point the object with a unique shape in the back as we decribed in the dataset introduction). As shown in Table 1, Slot-VAE achieves the best S-Acc score among all slot representation-based models (GENESIS, GENESIS-V2 and SRI).

H Additional Generation Results of Slot-VAE.

We show additional random novel scene generation examples of Slot-VAE on *ObjectsRoom ShapeStacks* and *Arrow Room* in Fig.7 - Fig. 9

I Additional Decomposition Results of Slot-VAE.

We show additional scene decomposition examples of Slot-VAE on *ObjectsRoom ShapeStacks* and *Arrow Room* in Fig.10 - Fig. 12

Table 2: ARI-FG (↑) for Slot-VAE and Baselines on ObjectsRoom and ShapeStacks. Mean and standard deviation of ARI with three runs are presented. Scores labelled with * are from original works (Engelcke et al., 2020) and (Emami et al., 2022).

MODEL	OBJECTSROOM	SHAPESTACKS
GNM	$0.63^{*} \pm 0.00$	$0.37^{*} \pm 0.07$
GENESIS	$0.63^{*} \pm 0.03$	$0.70^{*} \pm 0.05$
GENESIS-V2	$0.84^{*} \pm 0.01$	$0.81^{*} \pm 0.00$
SRI	$0.83^{*} \pm 0.02$	$0.78^{*} \pm 0.02$
SLOT-VAE (OURS)	0.79 ± 0.01	0.80 ± 0.01

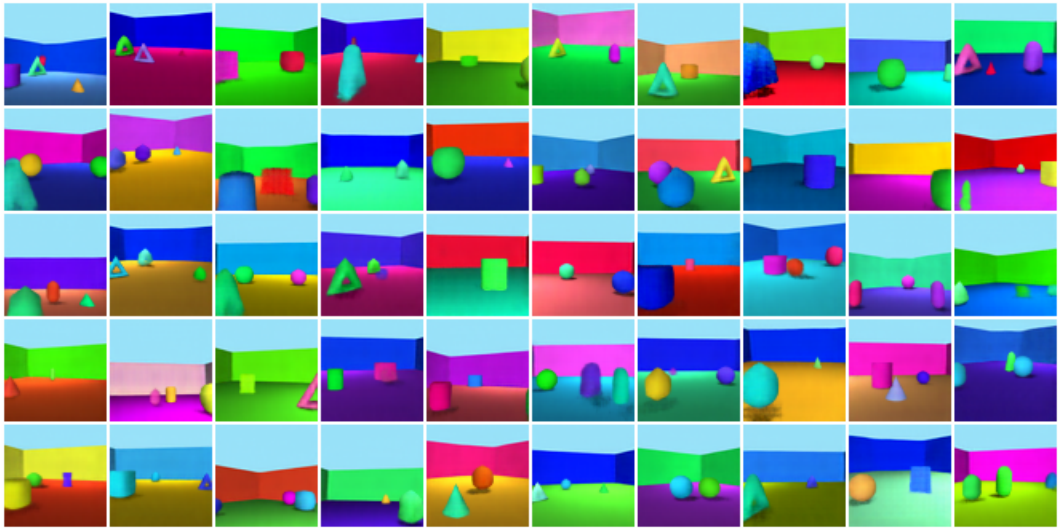

Figure 7: Additional generation resulst of Slot-VAE (Arrow Room dataset).

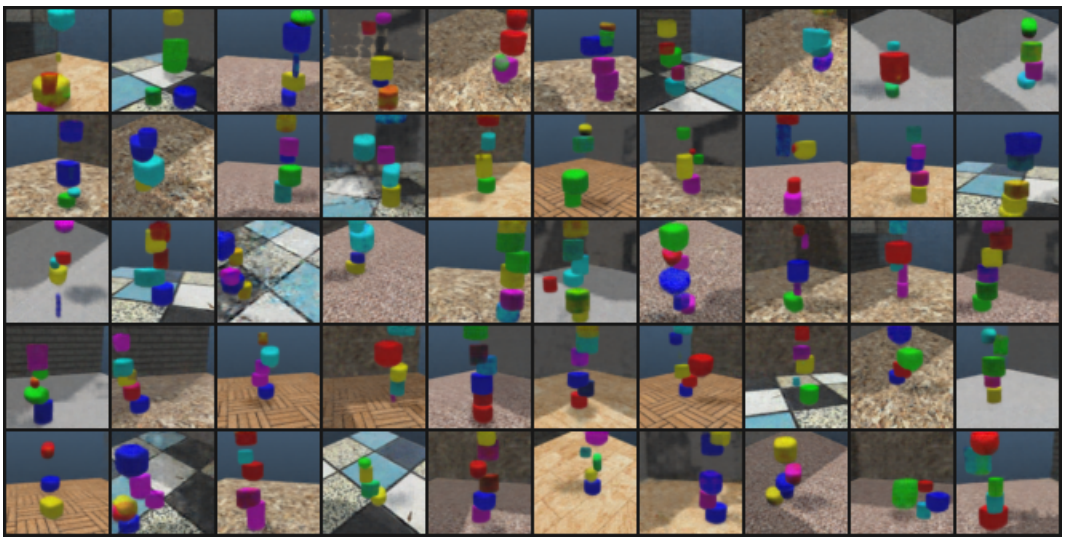

Figure 8: Additional generation resulst of Slot-VAE (ShapeStacks dataset).

Figure 9: Additional generation resulst of Slot-VAE (Arrow Room dataset).

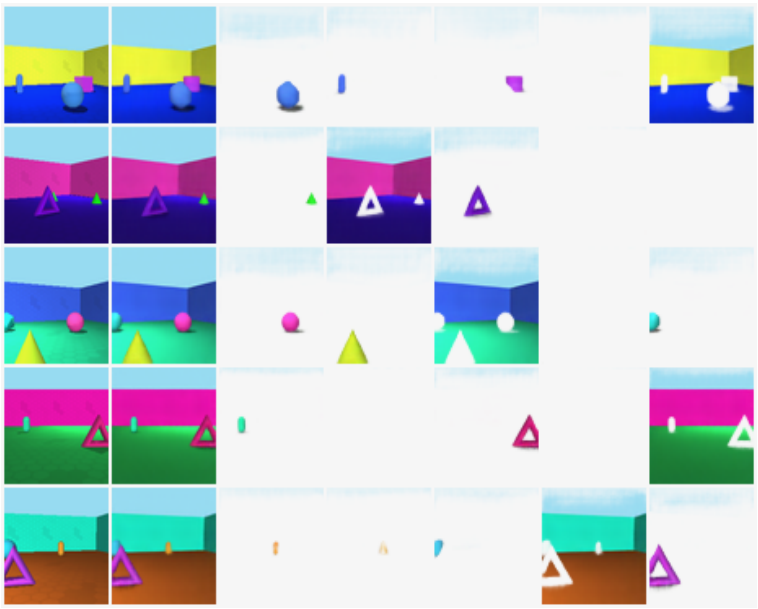

Figure 10: Additional decomposition resulst of Slot-VAE (ObjectsRoom dataset).

J Implementation Details of Slot-VAE.

In this section, we introduce the implementation details of Slot-VAE. As shown in 1, Slot-VAE has two parallel paths to train a two-layer hierarchical VAE model, which mainly includes the following four modules.

CNN backbone. Before inferring the global latent representation and slot representations, the input image is first fed into a convolutional neural network to extract relatively high-level features. This convolutional neural network has 4 layers, each layer is with kernel size 5 and stride 1 and the final layer has 64 channels. The obtained feature map $\mathbf{f}_x$ still has image-sized dimensions and each feature (channel) has a dimension of 64, i.e., the dimension is $H \times W \times 64$. Soft position embeddings are then added to the feature map to provide position information for the following modules. The architecture of the CNN backbone model is shown in Tabel

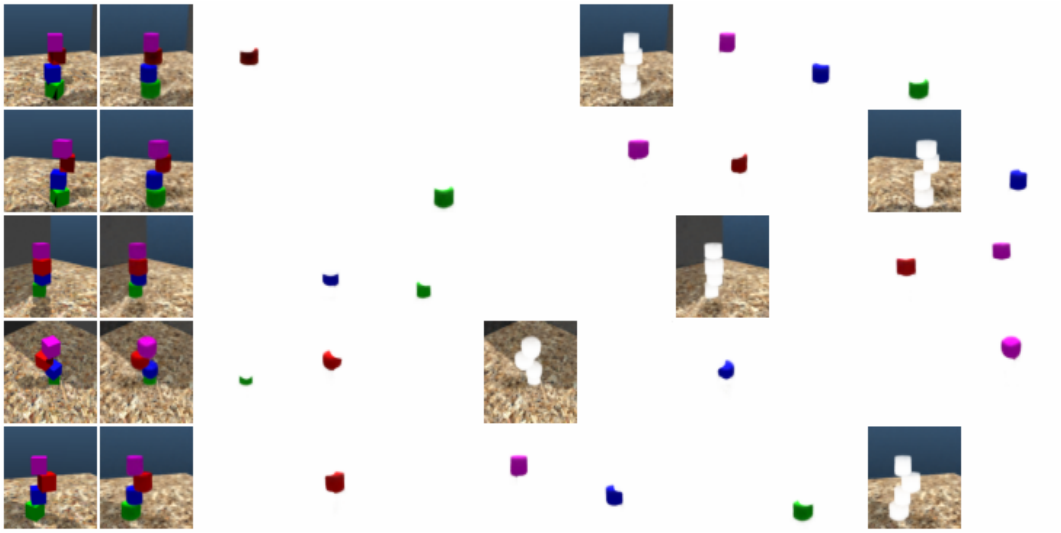

Figure 11: Additional decomposition resulst of Slot-VAE (ShapeStacks dataset).

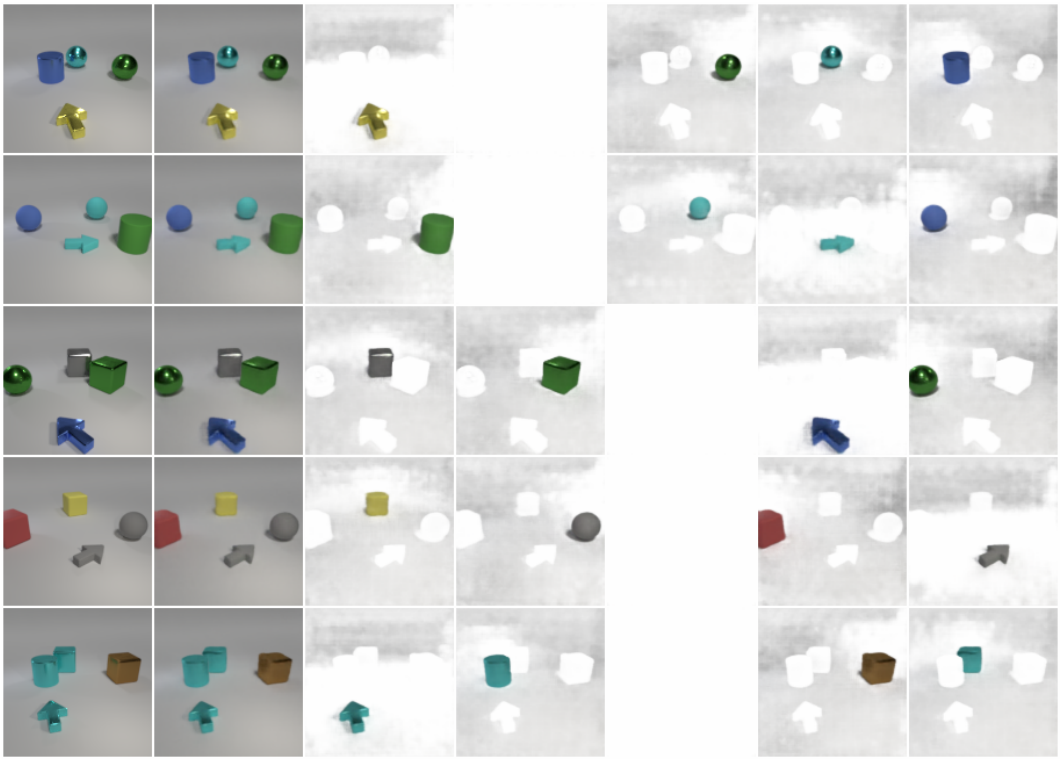

Figure 12: Additional decomposition resulst of Slot-VAE (ShapeStacks dataset).

Slot Attention Module. On the first path, we adopt the slot attention module (Locatello et al., 2020) for object-centric representation learning. We include the details for the self-containing purpose. To prepare for slot learning, the feature map $\mathbf{f}_x$ is first flattened into vectors $\mathbf{f}_{input}$ with dimension $(H \times W) \times 64$. To cluster the feature vectors into object components, the clustering center, i.e., slots, should be initialized first. The initialization values for object slots are from Gaussian distribution respectively, i.e., $\mathbf{s}_{1:K} \sim \mathcal{N}(\mu, \text{diag}(\sigma)) \in \mathbb{R}^{K \times 64}$, where μ and σ are learnable parameters. These slots are then updated iteratively to compete for explaining feature vectors $\mathbf{f}_{input}$. The slot

competition is achieved via a softmax-based attention mechanism :$\text{attn}_{i,j} := \frac{\exp(M_{i,j})}{\sum_l \exp(M_{i,l})}$, where $M := \frac{1}{\sqrt{D}} k(\mathbf{f}_{input}) \cdot q(\mathbf{s}_{1:K})^T \in \mathbb{R}^{(H\times W)\times K}$, and k and q are learnable linear mappings $\mathbb{R}^{D\to D}$ as commonly used in the attention mechanism, and $\sqrt{D}$ is a fixed value for softmax temperature. With the calculated attention scores $\text{attn}_{i,j}$, image feature vectors $\mathbf{f}_{input}$ are aggregated via weighted mean: $\text{updates} := \mathbf{W}^T \cdot v(\mathbf{f}_{input}) \in \mathbb{R}^{K\times D}$, where $\mathbf{W}_{i,j} := \text{attn}_{i,j}/(\sum_{l=1}^{N} \text{attn}_{l,j})$, and v is also learnable linear mappings similar to k and q. The update of slots in each iteration is completed via a learnable mapping parameterized by a Gated Recurrent Unit (GRU): $\mathbf{s}_{1:K} \leftarrow \text{GRU}(\mathbf{s}_{1:K}, \text{updates})$. The attention computation and updating are repeated 3 iterations to output final object-centric representations $\mathbf{s}_{1:K}$. Finally we obtain K vectors $\mathbf{s}_k$ each of dimension 64. To infer probabilistic random variables from $\mathbf{s}_k$, a MLP is used to map $\mathbf{s}_k$ to $\mathbf{z}^s_k$. This MLP is implemented with two layers with the first layer followed by a RELU layer. To be emphasized, the MLP is shared across $\mathbf{s}_k$, to encourage common formats of object representations. The obtained object-centric latent vector $\mathbf{z}^s_k$ is still with a dimension of 64.

Global Auto-Encoding Module. To learn a global latent vector, the CNN backbone outputs $\mathbf{f}_x$ needs to be encoded by an encoder. Depending on the chosen prior distribution of the global latent vector, the encoder could have different structures. In the case that the global prior is Normal distribution, the encoder can be common ones used in vanilla VAE. Specifically, the $(H \times W) \times 64$ feature map is further flattened into one dimension, i.e., $(H \times W \times 64) \times 1$. Then a three-layer MLP, severing as an information bottleneck, reduces the dimension of obtained feature map to $\mathbf{z}^g$ of dimension 32×1. The obtained $\mathbf{z}^g$ can be decoded with deconvolutional neural nets back to the dimension of $(H \times W) \times 64$, trying to reconstruct the feature map. However, since the decoded feature map $\mathbf{f}$ is not used to recover the image, rather generated object-centric latent vectors $\mathbf{z}^s_k$, there is no guarantee that $\mathbf{f}$ will be the same as $\mathbf{f}_x$. But with proper training, they should be close to each other. In summary, the auto-encoding structure is the same as commonly used VAE architecture. Another case for this global auto-encoding module is that a more powerful Strucdraw prior is used for the global latent vector learning. In that case, $\mathbf{z}^g$ is inferred autoregressively, the detail of such an encoder architecture could be found in (Jiang & Ahn, 2020). Along the path of global auto-encoding, the obtained $\mathbf{z}^g$ of dimension 32 is then fed into a slot attention module. This slot attention module has exactly the same architecture as the one on the first path. The two slot attention modules share parameters.

Object Component Decoder. We choose the SBD decoder (Watters et al., 2019) as part of the object component decoder in our model. Different from (Locatello et al., 2020) and (Engelcke et al., 2019) where a pure SBD is used, we combine SBD decoder with deconvolutional neural networks to balance the capacity of the decoder. Specifically, each object-centric latent vectors $\mathbf{z}^s_k$ of dimension 64 is first broadcast to a feature with shape $8 \times 8 \times 64$. Then this feature is decoded with deconvolutional neural nets with each layer having stride 2 and kernel size 5, to reconstruct an image-sized tensor with an additional channel as the mixing masks. The final output of the decoder has the shape $H \times W \times 4$. This decoder is shared across object-centric latent vectors $\mathbf{z}^s_k$.

K A ZOOM-IN VERSION OF FIG. 2

This section shows Fig. 13 that is a zoom-in version of Fig. 2 for better visualization. In this zoom-in version illustration, one can clearly see that ObjectRoom image samples generated by GNM show severe artifacts. The walls are composed of multiple stripe parts, and the objects also have multiple parts. This is because the GNM fails to learn object-centric representation by using bounding box representations, as shown in Fig. 4.

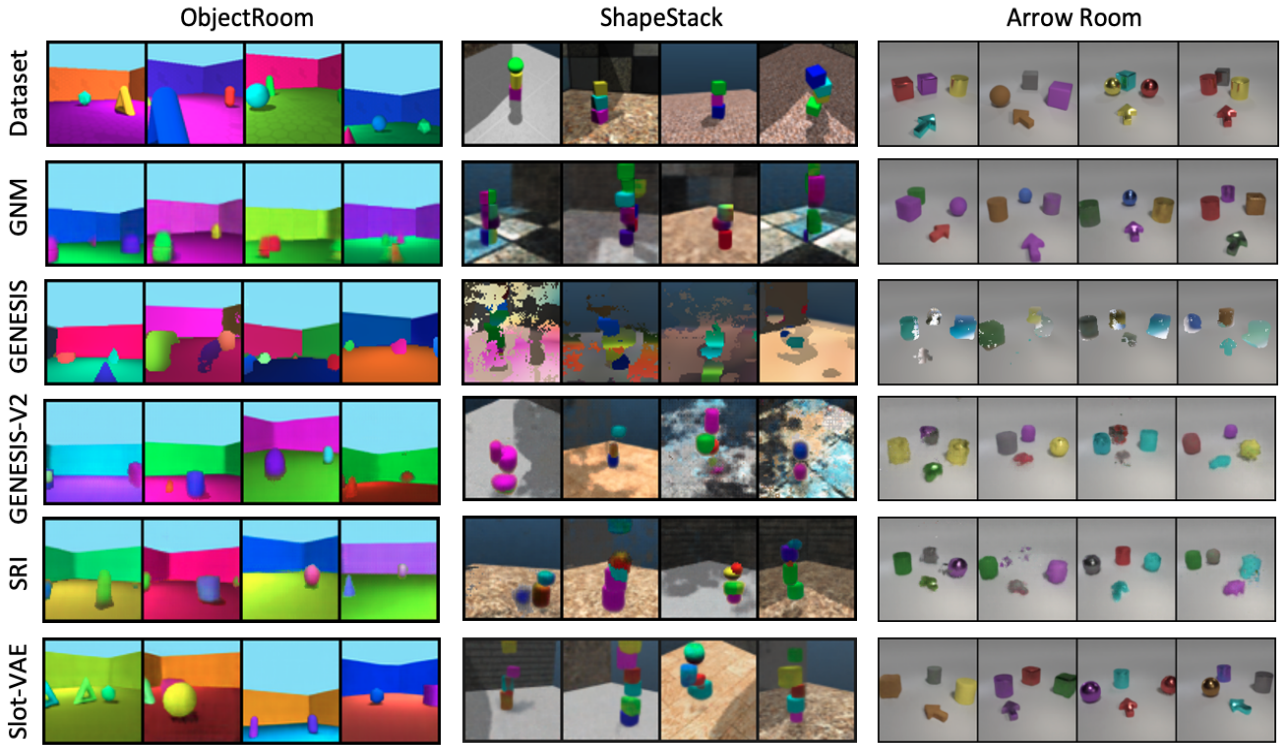

Figure 13: Datasets and generation examples of Slot-VAE and baselines.