Slot-VAE: Object-Centric Scene Generation with Slot Attention

Yanbo Wang [1] Letao Liu [2] Justin Dauwels [1]

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 scene generation. For 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 the global scene representation to ensure coherent scene structures. Our extensive evaluation of the scene generation ability indicates that Slot-VAE outperforms slot representation-based generative baselines in terms of sample quality and scene structure accuracy.

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 in an unsupervised way has been a desideratum for a long time (Johnson-Laird, 1983) (Ha & Schmidhuber, 2018) (Wu et al., 2021) (Schölkopf et al., 2021), since this can facilitate intelligent agents understanding scenes, reasoning about object relationships, and performing tasks efficiently (Battaglia et al., 2013) (Lake et al., 2017) (Geiger et al., 2012) (Cordts et al., 2016) (Santoro et al., 2017) (Devin et al., 2018) (Greff et al., 2020) (Mambelli et al., 2022). To that end, 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 to segment 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 complex 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.

In this paper, 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. In the proposed model, object-centric representation inference is achieved with the slot attention module (Locatello et al., 2020). Although slot attention has shown very impressive unsupervised segmentation performance, it is a deterministic module without the ability to generate novel scenes. If we naïvely combine slot attention with vanilla VAE for multi-object

[1] Department of EEMCS, Delft University of Technology, Delft, Netherlands [2] School of EEE, Nanyang Technological University, Singapore. Correspondence to: Yanbo Wang <y.wang-27@tudelft.nl>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

image generation, the generated images would be unreasonable because slots are completely independent and the scene structure (e.g., object relationships) is 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 also regularized by an independent prior to encourage object-centric disentanglement. Furthermore, the variational framework and independent prior also bring slot attention the attribute-level disentanglement. Evaluating on several multi-object datasets, we show that Slot-VAE outperforms baselines in terms of sample quality and scene structure learning.

The contributions of the paper are as follows. First, we introduce a generative model that embeds slot attention into the principled latent variable modelling framework for novel scene generation. Second, we incorporate a hierarchical latent variable model to learn both scene-level and object-centric representations. Third, we empower the slot attention baseline with object attribute-level disentanglement ability. Lastly, extensive experimental results suggest our proposed method outperforms the state-of-the-art methods in terms of sample quality and scene structure accuracy.

2. 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 incorporate object locality as inductive bias or exploit 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. GNM (Jiang & Ahn, 2020) and similarly GSGN (Deng et al., 2021) resort to a hierarchical VAE model for both distributed and symbolic representations learning, but the bounding box representations therein prevent them from modelling complex objects or backgrounds, unlike ours where more flexible slot representations are 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 learning so as to train the model.

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.

3. The Proposed Model: Slot-VAE

The overview of Slot-VAE is illustrated in Fig. 1.

3.1. Generation

For an image $\mathbf{x} \in [0,1]^{H\times W\times C}$ with height H, width W and C channels, 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. These slot representations $\mathbf{z}^s_{1:K}$ are assumed to be conditionally independent given $\mathbf{z}^g$. 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:

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

The global latent vector $\mathbf{z}^g$ serves as an information bottleneck to extract high-level information (e.g., object appearance, positions and relations) for whole image reconstruction. $\mathbf{z}^g$ is similar to the latent vector in VAE but not exactly the same. The difference is that in VAE the latent vector is directly decoded to an image, while in Slot-VAE $\mathbf{z}^g$ is used to generate slot representations $\mathbf{z}^s_{1:K}$. 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.

Slot representations $\mathbf{z}^s_{1:K}$, in contrast to $\mathbf{z}^g$, ideally embeds information of individual object components and totally ignores object relationships. Such object-centric representations explicitly model the compositional structure of images, enable compositional generation and make the generation

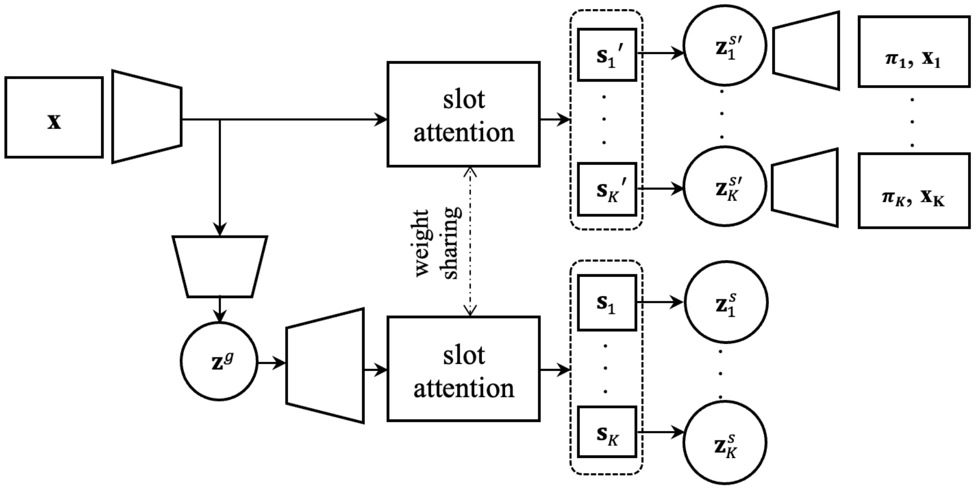

Figure 1. Slot-VAE overview. The image $\mathbf{x}$ is passed through a CNN module. The obtained image features go through two paths in parallel. On the first path, the obtained image features are input into a slot attention module to learn slot representations $\{\mathbf{s}'_k\}_{k=1}^K$. From slots $\{\mathbf{s}'_k\}_{k=1}^K$, latent vectors $\{\mathbf{z}_k^{s'}\}_{k=1}^K$ are inferred. Then, a shared decoder decodes the individual object latent vector $\{\mathbf{z}_k^{s'}\}_{k=1}^K$ 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. On the second path, the obtained image features is 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 $\{\mathbf{s}_k\}_{k=1}^K$. From $\{\mathbf{s}_k\}_{k=1}^K$, latent vectors $\{\mathbf{z}_k^s\}_{k=1}^K$ are inferred. The two paths use the same slot attention module and share weights and initialization values, and it requires $\{\mathbf{z}_k^{s'}\}_{k=1}^K$ and $\{\mathbf{z}_k^s\}_{k=1}^K$ to be as close as possible during training measured with KL divergence.

process interpretable. To generate $\mathbf{z}_{1:K}^s$ from $\mathbf{z}^g$, we first construct 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 $\mathbf{z}_{1:K}^s$. Assuming $\mathbf{z}_{1:K}^s$ are Gaussian and conditionally independent given $\mathbf{z}_g$, we have:

$$p_\theta(\mathbf{z}_{1:K}^s \mid \mathbf{z}^g) = \prod_{k=1}^K p_\theta(\mathbf{z}_k^s \mid \mathbf{z}^g). \tag{2}$$

The use of the slot attention module for object-centric latent vector generation sets the proposed Slot-VAE apart from GNM (Jiang & Ahn, 2020) where bounding box extraction is adopted. Such a difference brings the following key benefits. First, slot-based models have been shown to be more flexible in modelling objects with complex morphology compared with the spatial attention module (Lin et al., 2020). Second, the dimension of the feature map $\mathbf{f}$ in GNM fundamentally limits the maximum number of components in an image to be $H \times W$. Once a GNM model is trained, it at most can infer $H \times W$ objects. In contrast, the slot attention module can successfully generalize to infer more object components even though it only saw K object components during training. Comes with these benefits a key challenge to Slot-VAE: there is no fixed order for the slot attention outputs. Since slot attention maps an input into a set (of slots), for the same input image, multiple runs may give the same set of slot representations but with different orders. This is because slot attention employs random initialization for slots to achieve slot permutation symmetry. However, such randomness makes the learning of a hierarchical latent variable model extremely challenging, which we will explain in detail in Section 3.3 and contribute to solving it.

With $\mathbf{z}_{1:K}^s$, rendering an image $\mathbf{x}$ is as follows. First, from $\mathbf{z}_{1:K}^s$ (or $\mathbf{z}_{1:K}^{s'}$ in Fig. 1), K sub-images $\{\mathbf{x}_k \in [0,1]^{H \times W \times C}\}_{k=1}^K$ are rendered, each of which has the same dimension as $\mathbf{x}$ and ideally contains only one object. Meanwhile, this process also produces K object masks $\boldsymbol{\pi}_{1:K} \in [0,1]^{H \times W}$ corresponding to each $\mathbf{x}_k$. Then the image $\mathbf{x}$ is obtained by combining $\mathbf{x}_{1:K}$ with masks $\boldsymbol{\pi}_{1:K}$. Pixel-wisely, the likelihood can be written as

$$p_\theta(\mathbf{x}_{i,j} \mid \mathbf{z}_{1:K}^s) = \mathcal{N}\Bigg(\Big(\sum_{k=1}^K \pi_{i,j,k}(\mathbf{z}_{1:K}^s)\mu_{i,j,k}(\mathbf{z}_k^s)\Big), \sigma_x^2\Bigg), \tag{3}$$

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). These nonlinear functions are parameterized by neural networks with learnable parameters θ, and implementation details are provided in the appendix. In equation 3, $\pi_{i,j,k}$ serves as mixing probability, so it is constrained by $\sum_{k=1}^K \pi_{i,j,k} = 1, \forall (i,j)$.

In summary, to generate a novel scene, we first draw a random sample from the prior distribution of the global latent vector $\mathbf{z}^g$, from which a feature map $\mathbf{f}$ is built. Then, object-centric latent vectors $\mathbf{z}^s_{1:K}$ are generated by using the slot attention module with the feature map $\mathbf{f}$ as input. Finally, object components $\mathbf{x}_{1:K}$ and corresponding masks $\boldsymbol{\pi}_{1:K}$ are generated from $\mathbf{z}^s_{1:K}$ with parallel decoders, and a novel scene is rendered by combining $\mathbf{x}_{1:K}$ with $\boldsymbol{\pi}_{1:K}$.

3.2. Inference

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

$$p_\theta(\mathbf{z}^g, \mathbf{z}^s_{1:K} \mid \mathbf{x}) \approx q_\phi(\mathbf{z}^g \mid \mathbf{x}) q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}), \tag{4}$$

wherein the global latent posterior $q_\phi(\mathbf{z}^g \mid \mathbf{x})$ is modelled by an autoregressive model or Gaussian distribution depending on StructDRAW prior or Gaussian prior is used (Jiang & Ahn, 2020).

We further assume the factorization $q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) = \prod_{k=1}^K q_\phi(\mathbf{z}^s_k \mid \mathbf{x})$. Such conditional independence assumption on the posterior distribution of slot representations enables the inference of individual $\mathbf{z}^s_k$ to be performed in parallel, which avoids sequential inference like in GENESIS. We adopt slot attention (Locatello et al., 2020) followed by an MLP to infer $\mathbf{z}^s_{1:K}$, 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 $\{(\mu^s_k, \sigma^s_k)\}_{k=1}^K$. To that end, we use an MLP shared across objects mapping from slots to Gaussian means and variances: $(\mu^s_k, \sigma^s_k) :=$ MLP$(\mathbf{s}_k)$.

3.3. Training

Given the above generative and inference model, the ELBO can be derived as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{x}; \theta, \phi) = \ & \mathbb{E}_{q_\phi(\mathbf{z}^s_{1:K}|\mathbf{x})}\big[\log p_\theta(\mathbf{x} \mid \mathbf{z}^s_{1:K})\big] \\ & - D_{\mathrm{KL}}\big[q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g)\big] \\ & - D_{\mathrm{KL}}\big[q_\phi(\mathbf{z}^g \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^g)\big]\end{aligned} \tag{5}$$

where $D_{\mathrm{KL}}(q||p)$ is Kullback-Leibler Divergence.

Slot Order Matching in KL. Observing the second term on the RHS of equation 5, we can identify a key challenge for the calculation of this KL divergence: since the slots given by slot attention come with no fixed order, how can we determine the correspondence between $\mathbf{z}^s_{1:K}$ inferred from input $\mathbf{x}$ (which is denoted $\mathbf{z}^s_{1:K}{}'$ in Fig. 1) and $\mathbf{z}^s_{1:K}$ generated from $\mathbf{z}^g$? This challenge 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. To address such a challenge 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 $\mathbf{z}^s_k{}'$) in Fig. 1, enabling the calculation of $D_{\mathrm{KL}}\big[q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g)\big]$. We will empirically demonstrate the efficacy of such an architectural inductive bias for slot order matching in Section 4.

Furthermore, since $p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g)$ in the second term of equation 5 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 explicitly provide guidance to the learning of $q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x})$, the following auxiliary loss could be incorporated:

$$\mathcal{L}_{aux} = -D_{\mathrm{KL}}\big[q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) \mid\mid \prod_{k=1}^K \mathcal{N}(\mathbf{0}, \mathbf{1})\big], \tag{6}$$

where independent normal prior constrains $\mathbf{z}^s_{1:K}$ to be independent on each other. As a result, such independence 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, which we will show in experiments.

Combining the derived ELBO in equation 5 and the introduced auxiliary loss in equation 6, the overall objective function is:

$$\tilde{\mathcal{L}} = \mathcal{L} + \mathcal{L}_{aux}, \tag{7}$$

which is minimized to train Slot-VAE. For effective training, we also introduce hyperparameters to balance the reconstruction loss and KL terms (Rezende & Viola, 2018) (Fu et al., 2019), which will be detailed in the Appendix.

4. Experiments

The experiments are to evaluate: i) image decomposition performance, ii) sample quality and structure accuracy of generated samples, iii) and disentanglement performance.

Dataset. The experiments involve three datasets including *ObjectRoom* (Kabra et al., 2019), *ShapeStacks* (Groth et al., 2018) and *Arrow Room*(Jiang & Ahn, 2020). The datasets *ObjectRoom* and *ShapeStacks* are commonly used by previous works to test object-centric inference and generation, while *Arrow Room* is less considered 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 of the scene and three objects in the back. Two of the three objects in the back have the same shape, while a third one has a unique shape. The arrow in the front always points to the object with a unique shape in the back.

Baselines. We compare Slot-VAE against state-of-the-art object-centric generative models including GENESIS, GENESIS-V2, SRI and GNM. In these baselines, GNM is based on the spatial attention model (i.e., bounding box representations) with hierarchical generation process, while GENESIS, GENESIS-V2 and SRI are scene-mixture models (i.e., slot representations) that assume an autoregressive prior. Some of the baseline models already released their trained models for *ObjectRoom*, *ShapeStacks* or *Arrow Room*. We do not retrain them and directly use their weights for comparison. For these of baseline models without trained models on some datasets, we train them with the official code.

4.1. Qualitative Comparison of Image Decomposition, Image Reconstruction and Sample Generation

Decomposition and Reconstruction Performance. We illustrate the input, reconstruction and decomposed object components of Slot-VAE and baselines in Fig. 2 - 4. Note that GNM infers bounding box representations instead of slot representations. So in the figures, GNM has only two components, one for the foreground with bounding boxes and another for the background.

As shown in Fig. 2, for the *ObjectRoom* dataset that comes with simple object shapes and complex background components, scene mixture 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, 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 and objects are blurred. 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 complex morphology. In Fig. 3, we observe similar results for the *ShapeStacks* dataset, where GENESIS, GENESIS-V2, SRI and Slot-VAE decompose and reconstruct the image reasonably well while 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 will be presented below. For the *Arrow Room* dataset that has simple object shapes but complex scene structures in Fig. 4, 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.

Generation Performance. We show random samples generated by Slot-VAE and baseline models in Fig. 5. It can be seen Slot-VAE generates the sharpest samples that highly resemble all the datasets. For *ObjectRoom*, samples generated by GNM show stripe artifacts due to its inaccurate object-centric representations captured by bounding boxes as discussed above. The sample quality of SRI is better than that of GENESIS and GENESIS-V2, but not as good as the proposed Slot-VAE. This can be reflected by the sharpness of object edges in the images. One can more easily identify object shapes (e.g., balls and triangles) 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 represented by two small parts with completely different colors. Only SRI and Slot-VAE generate reasonable samples reflecting the scene structure of the *ShapeStacks* dataset (i.e., one object is stacked on another), while the sample quality of Slot-VAE is better in terms of sharp object edges. For *Arrow Room*, the most structured dataset, we find samples generated by GENESIS, GENESIS-V2 and SRI are very blurry and seldom show the underlying true scene structure (i.e., the arrow in the front always points to the object with a unique shape in the back). Both arrow directions or object shapes are not properly learned. This indicates that the autoregressive prior adopted in GENESIS, GENESIS-V2 and SRI is not strong enough to capture the complex scene structure in *Arrow Room*. In contrast, GNM and Slot-VAE,

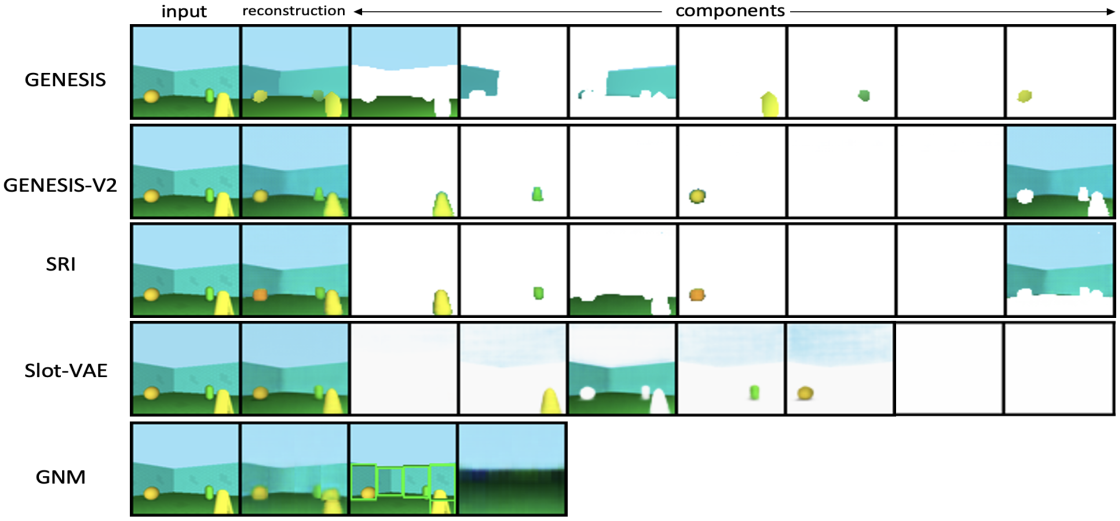

Figure 2. Image decompostion and reconstruction performance on the ObjectsRoom dataset.

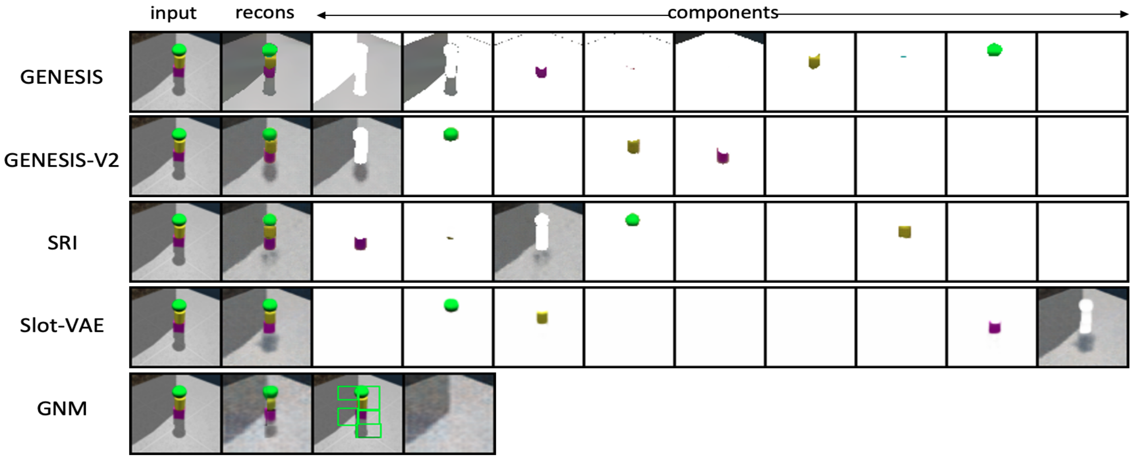

Figure 3. Image decompostion and reconstruction performance on the ShapeStacks dataset.

both exploiting hierarchical model to capture scene structure, generate very coherent and high-quality samples on the *Arrow Room* dataset. The reason why GNM works better on *Arrow Room* in comparison to *ObjectRoom* and *ShapeStacks* is that object shapes are simple in *Arrow Room*. In summary, Slot-VAE outperforms baselines in terms of sample quality and scene structure learning. Additional random generation results of Slot-VAE can be found in the Appendix.

Scene Manipulation. We elaborate on controllable scene generation to highlight the disentanglement performance of Slot-VAE. In Fig. 6, in each row we vary a certain dimension of the object-centric latent vector corresponding to the ball object while keeping other object-centric latent vectors unchanged. As is shown, only attributes of the ball are changed in each row, and all other objects remain unaffected. Such object-level disentanglement is very useful for image editing and compositional generation. Besides object-level disentanglement, attributes-level disentanglement also naturally appears in Slot-VAE due to the adopted probabilistic framework. As shown in Fig. 6, when we vary dimension 1,

Table 1. 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

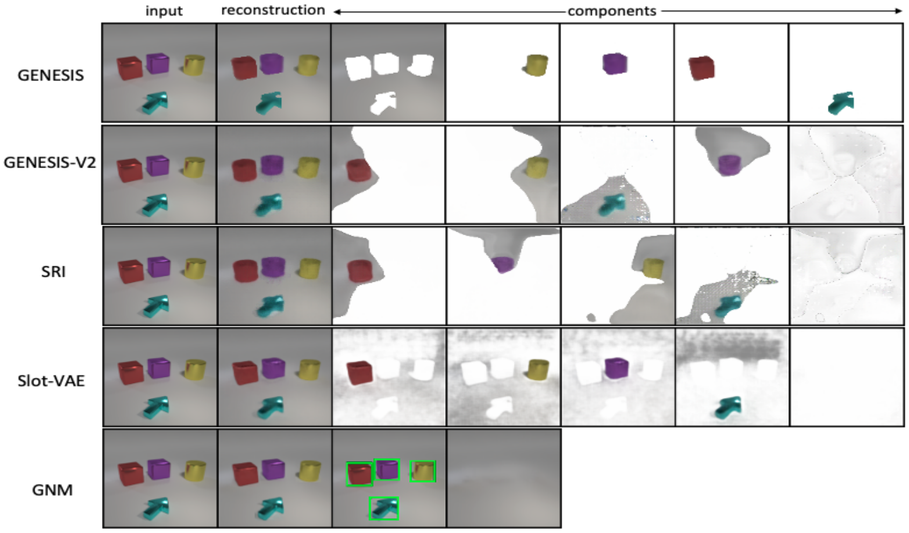

Figure 4. Image decompostion and reconstruction performance on the Arrow Room dataset.

Table 2. Fréchet Inception Distances (FID ↓) and Structure Accuracy (S-Acc ↑) for Slot-VAE and Baselines. Mean and standard deviation of FID with three runs are presented. Scores labelled with * are from original works (Engelcke et al., 2020) and (Emami et al., 2022).

	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**

the texture of the ball changes; when we vary dimension 2, the color of the ball changes; when we vary dimension 3, the size of the ball changes. Although some dimensions (e.g., dim 4) entangle color and position a little, this can be further improved with existing attribute-level disentanglement techniques like β-VAE (Higgins et al., 2017) or β-TCVAE (Chen et al., 2018), which is out of the scope of this paper. In the proposed Slot-VAE, attribute-level disentanglement is a by-product brought by the VAE framework. By contrast, the original deterministic slot attention module comes with no obvious attribute-level disentanglement as analyzed in (Singh et al., 2022).

4.2. Quantitative Comparison

We report the Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) score, Frechet Inception Distance (FID) (Heusel et al., 2017) score and scene structure accuracy (S-Acc) (Jiang & Ahn, 2020) score to quantitatively evaluate the decomposition performance, sample quality, and scene structure accuracy. Since the *Arrow Room* dataset comes with no ground truth masks, the ARI score on this dataset is not calculated. As shown in Table 1, slot-VAE achieves comparable ARI scores to baselines. For the FID score, the calculation involves 10000 real and generated samples. Table 2 reflects non-trivial FID score improvement by Slot-VAE against slot-representation baselines, highlighting the sample quality of Slot-VAE. Although the FID score of GNM on *ObjetsRoom* and *ShapeStacks* seems quite good, it should be emphasized that the generated images are unrealistic (i.e., generated objects are composed of multiple rectangular parts) due to inaccurate object representation learning as analyzed in the qualitative comparison results. For the S-Acc score, we manually classified 100 generated images per model, and calculated the ratio of successful images that correctly reflect scene structure. The datasets *ObjetsRoom* and *ShapeStacks* have relatively less clearly defined structures, which may result in difficulty in deciding if generated images truly reflect scene structures. To reduce subjective decisions, we mainly evaluate S-Acc of Slot-VAE and baseline models on the *Arrow Room* dataset because this dataset has a clearly defined structure: the arrow object should always point to the object with a unique shape in the back. Slot-VAE achieves the best S-Acc score among all the

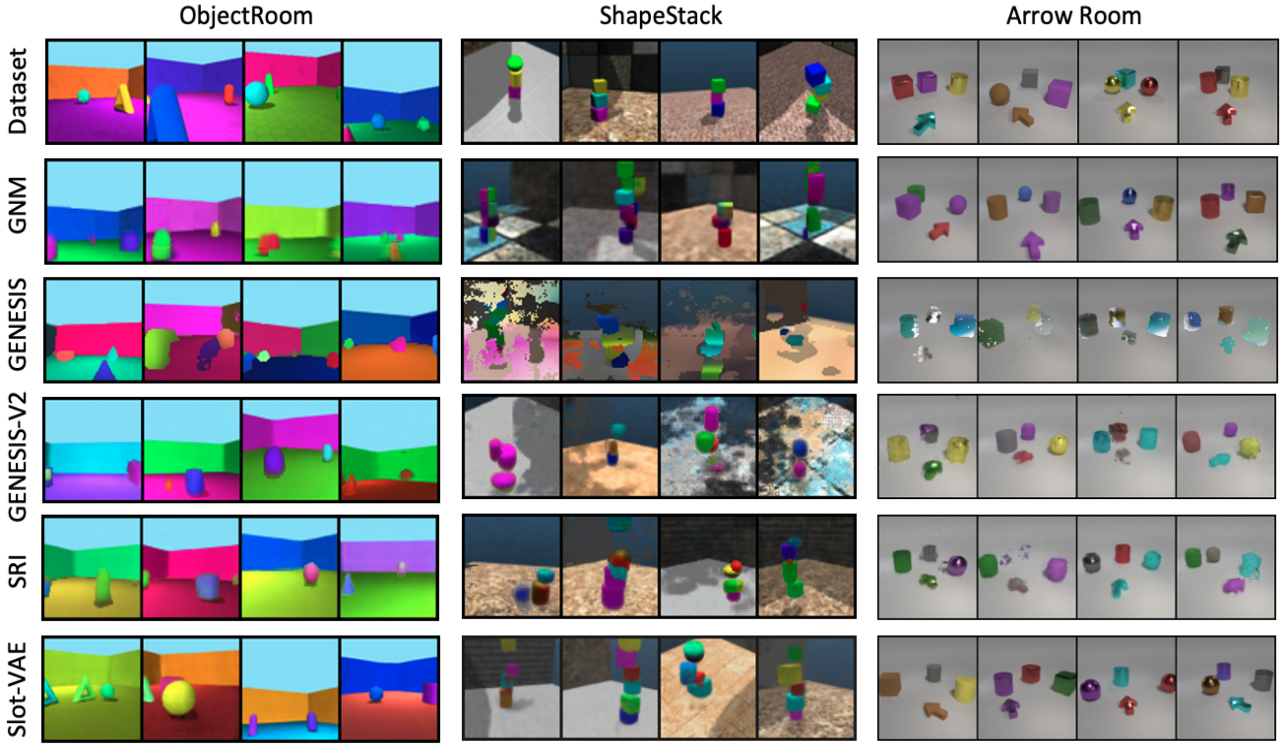

Figure 5. Datasets and generation examples of Slot-VAE and baselines.

Table 3. FID (↓) score and S-Acc (↑) score of Slot-VAE and variants on the *Arrow Room* dataset.

MODEL	FID	S-ACC
SLOT-VAE	**60.3±1**	**0.94**
SLOT-VAE-MLP	289±9	0.00
SLOT-VAE-TRANSFORMER	182.1±3	0.03
SLOT-VAE-W/O-WS	215.5±2	0.00
SLOT-VAE-W/O-IVS	142.1±3	0.05

slot representation-based models (GENESIS, GENESIS-V2 and SRI), as is shown in Table 2.

4.3. Ablation Study

We further conduct experiments to demonstrate the efficacy of the proposed architectural design in Fig. 1. Specifically, we aim to answer the following questions: (1) whether slot attention is necessary for generating slot representations from the global representation and (2) whether slot attention weight sharing and initialization value sharing are necessary for slot order matching. To that end, we evaluated the FID score and S-Acc score of several Slot-VAE variants.

To answer question (1), we investigate two approaches that could be used as alternatives to slot attention to generating slot representations $\{\mathbf{s}_k\}_{k=1}^K$ from the global representation $\mathbf{z}^g$. The first approach (termed as Slot-VAE-MLP) is by using an MLP to directly map the $\mathbf{z}^g$ to $\{\mathbf{s}_k\}_{k=1}^K$. Although this approach is straightforward, it cannot work well intuitively. Specifically, an MLP learns a deterministic mapping that always outputs slots $\{\mathbf{s}_k\}_{k=1}^K$ with a fixed order for a given global latent vector, whereas the slots $\{\mathbf{s}'_k\}_{k=1}^K$ that are directly inferred from the input image with slot attention come with a random order. As a result, the order of $\{\mathbf{z}_k^s\}_{k=1}^K$ and that of $\{\mathbf{z}_k^{s'}\}_{k=1}^K$ can rarely match each other, leading to fluctuating KL divergence $D_{\text{KL}}\left[q_\phi(\mathbf{z}_{1:K}^s \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}_{1:K}^s \mid \mathbf{z}^g)\right]$ between slot prior and slot posterior and hence diverged training. This can be reflected by the very high FID score and low S-Acc score in Table 3. The second approach (termed as Slot-VAE-Transformer) is by using a transformer to map the global vector $\mathbf{z}^g$ and random initialization values of slots $\{\mathbf{s}_k\}_{k=1}^K$ shared with $\{\mathbf{s}'_k\}_{k=1}^K$ to slot representations. In this approach, slots generated by the transformer is permutation invariant due to random initialization, which addresses the fixed slot order issue in Slot-VAE-MLP. Intuitively, with shared initialized values, slots $\{\mathbf{s}_k\}_{k=1}^K$ generated from $\mathbf{z}^g$ and slots $\{\mathbf{s}'_k\}_{k=1}^K$ inferred from the input image could have a good chance to match each other. Indeed, with this approach, our model matches the orders of the slots well. However, the generated slots turn out not so good in the sense that their corresponding decoded object components are very blurry. As a result, Slot-VAE-Transformer also

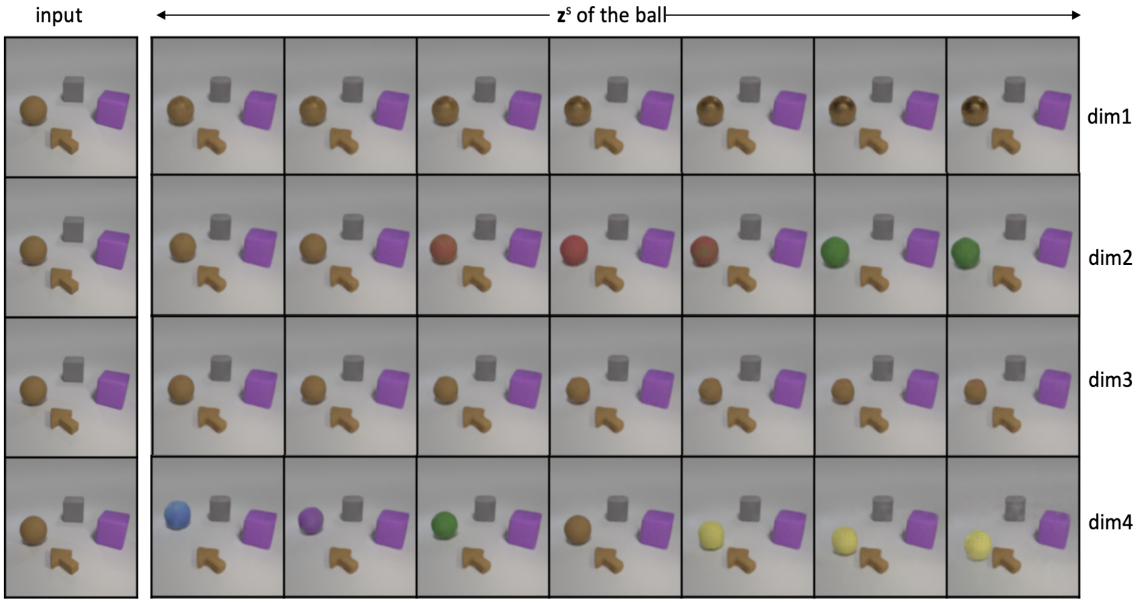

Figure 6. Slot-VAE latent traversal on *Arrow Room*. Each row only varies a certain dimension of $\mathbf{z}^s$ corresponding to the ball object.

has a very high FID score and a low S-Acc score. In contrast, Slot-VAE outperforms Slot-VAE-MLP and Slot-VAE-Transformer significantly, which demonstrates the effectiveness of slot attention for generating slot representations from the global representation.

To answer question (2), we trained a variant of Slot-VAE (termed as Slot-VAE-W/O-WS) without the weight sharing strategy in Fig. 1. In this case, the two slot attention modules update their weights respectively with no common initialization values. Without weight sharing, we anticipate that the KL divergence $D_{\mathrm{KL}}\left[q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g)\right]$ could be large because the learned slot representations of the two attention modules can be quite different, which may result in unrealistic generation samples. This is demonstrated by the experimental results in Table 3. We also trained another variant of Slot-VAE (termed as Slot-VAE-W/O-IVS) with weight sharing between the two slot attention modules but without initialization value sharing. Without initialization value sharing, the order of slots $\{\mathbf{s}_k\}_{k=1}^K$ generated from $\mathbf{z}^g$ and the order of slots $\{\mathbf{s}'_k\}_{k=1}^K$ inferred from the input image cannot match each other very well. As a result, the KL divergence $D_{\mathrm{KL}}\left[q_\phi(\mathbf{z}^s_{1:K} \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^s_{1:K} \mid \mathbf{z}^g)\right]$ can not be properly calculated, and generated samples cannot reflect the dataset structure as quantitatively shown in Table 3.

In summary, we empirically find that the slot attention module for generating slot representations from the global representation, weight sharing and initialization value sharing between the two attention modules improve the generation performance significantly.

5. Conclusion

We propose an object-centric generative model, Slot-VAE, that integrates the slot attention module with a hierarchical VAE model for joint object-centric representation inference and scene structure modelling. The proposed model can discover object components in an unsupervised way and generate novel scenes controllable at both the object and attribute level. Experiment results show that Slot-VAE achieves better sampling quality and scene structure accuracy compared to slot representation-based generative baselines.

One limitation of Slot-VAE is that the adopted slot attention module requires simple decoders like SBD (Watters et al., 2019) to serve as a reconstruction bottleneck to decompose objects, which, however, may not scale to complex real-world scenes. This can be improved by using a transformer decoder (Singh et al., 2021) or diffusion model-based decoder (Jiang et al., 2023), which we leave for future work.

Social Impact

The proposed Slot-VAE model shows no negative social impacts in its current form since the evaluation is carried out on synthetic datasets at this stage. However, with improved slot representation learning modules available in the future, our model has the potential to be applied to generate more sophisticated and realistic scenes. In that case, misuse should be avoided for malicious purposes. Proper use of the proposed model can actually benefit practical applications like artwork generation, scene understanding, and dataset augmentation, to name just a few.

References

Battaglia, P. W., Hamrick, J. B., and Tenenbaum, J. B. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110 (45):18327–18332, 2013.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., and Lerchner, A. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

Chen, R. T. Q., Li, X., Grosse, R., and Duvenaud, D. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

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

Deng, F., Zhi, Z., Lee, D., and Ahn, S. Generative scene graph networks. In *International Conference on Learning Representations*, 2021.

Devin, C., Abbeel, P., Darrell, T., and Levine, S. Deep object-centric representations for generalizable robot learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7111–7118. IEEE, 2018.

Ehrhardt, S., Groth, O., Monszpart, A., Engelcke, M., Posner, I., Mitra, N., and Vedaldi, A. Relate: Physically plausible multi-object scene synthesis using structured latent spaces. *Advances in Neural Information Processing Systems*, 33:11202–11213, 2020.

Elsayed, G. F., Mahendran, A., van Steenkiste, S., Greff, K., Mozer, M. C., and Kipf, T. Savi++: Towards end-to-end object-centric learning from real-world videos. *arXiv preprint arXiv:2206.07764*, 2022.

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

Emami, P., He, P., Ranka, S., and Rangarajan, A. Slot order matters for compositional scene understanding. *arXiv preprint arXiv:2206.01370*, 2022.

Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019.

Engelcke, M., Jones, O. P., and Posner, I. Reconstruction bottlenecks in object-centric generative models. *arXiv preprint arXiv:2007.06245*, 2020.

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

Eslami, S., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Hinton, G. E., et al. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in neural information processing systems*, 29, 2016.

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

Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., and Carin, L. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *arXiv preprint arXiv:1903.10145*, 2019.

Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361. IEEE, 2012.

Greff, K., Van Steenkiste, S., and Schmidhuber, J. Neural expectation maximization. *Advances in Neural Information Processing Systems*, 30, 2017.

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pp. 2424–2433. PMLR, 2019.

Greff, K., Van Steenkiste, S., and Schmidhuber, J. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

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

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

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

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

Jiang, J., Janghorbani, S., De Melo, G., and Ahn, S. Scalor: Generative world models with scalable object representations. *arXiv preprint arXiv:1910.02384*, 2019.

Jiang, J., Deng, F., Singh, G., and Ahn, S. Object-centric slot diffusion. *arXiv preprint arXiv:2303.10834*, 2023.

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

Kabra, R., Burgess, C., Matthey, L., Kaufman, R. L., Greff, K., Reynolds, M., and Lerchner, A. Multi-object datasets. https://github.com/deepmind/multi-object-datasets/, 2019.

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

Kipf, T., Elsayed, G. F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., and Greff, K. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021.

Kosiorek, A., Kim, H., Teh, Y. W., and Posner, I. Sequential attend, infer, repeat: Generative modelling of moving objects. *Advances in Neural Information Processing Systems*, 31, 2018.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Liao, Y., Schwarz, K., Mescheder, L., and Geiger, A. 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.

Lin, Z., Wu, Y.-F., Peri, S. V., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. *arXiv preprint arXiv:2001.02407*, 2020.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

Mambelli, D., Träuble, F., Bauer, S., Schölkopf, B., and Locatello, F. Compositional multi-object reinforcement learning with linear relation networks. *arXiv preprint arXiv:2201.13388*, 2022.

Nguyen-Phuoc, T. H., Richardt, C., Mai, L., Yang, Y., and Mitra, N. Blockgan: Learning 3d object-aware scene representations from unlabelled images. *Advances in Neural Information Processing Systems*, 33:6767–6778, 2020.

Niemeyer, M. and Geiger, A. 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.

Rezende, D. J. and Viola, F. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30, 2017.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, 2021.

Seitzer, M., Horn, M., Zadaianchuk, A., Zietlow, D., Xiao, T., Simon-Gabriel, C.-J., He, T., Zhang, Z., Schölkopf, B., Brox, T., et al. Bridging the gap to real-world object-centric learning. *arXiv preprint arXiv:2209.14860*, 2022.

Singh, G., Deng, F., and Ahn, S. Illiterate dall-e learns to compose. In *International Conference on Learning Representations*, 2021.

Singh, G., Kim, Y., and Ahn, S. Neural systematic binder, 2022. URL https://arxiv.org/abs/2211.01177.

Van Steenkiste, S., Kurach, K., Schmidhuber, J., and Gelly, S. Investigating object compositionality in generative adversarial networks. *Neural Networks*, 130:309–325, 2020.

Watters, N., Matthey, L., Burgess, C. P., and Lerchner, A. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

Wu, B., Nair, S., Martin-Martin, R., Fei-Fei, L., and Finn, C. Greedy hierarchical variational autoencoders for large-scale video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2318–2328, 2021.

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

A. Additional Results of Slot-VAE.

We show additional scene decomposition and novel scene generation examples of Slot-VAE on *ObjectsRoom ShapeStacks* and *Arrow Room* in Fig.7 - Fig. 12

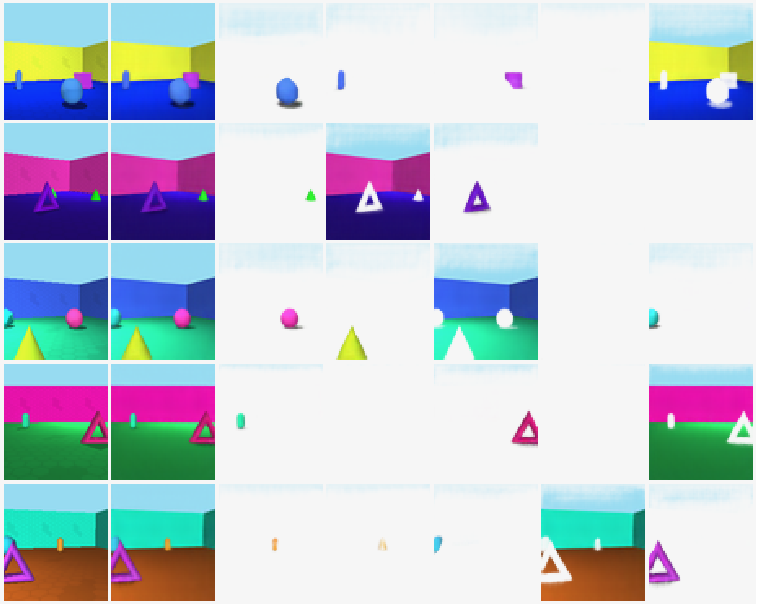

Figure 7. Additional decomposition resulst of Slot-VAE (ObjectsRoom dataset).

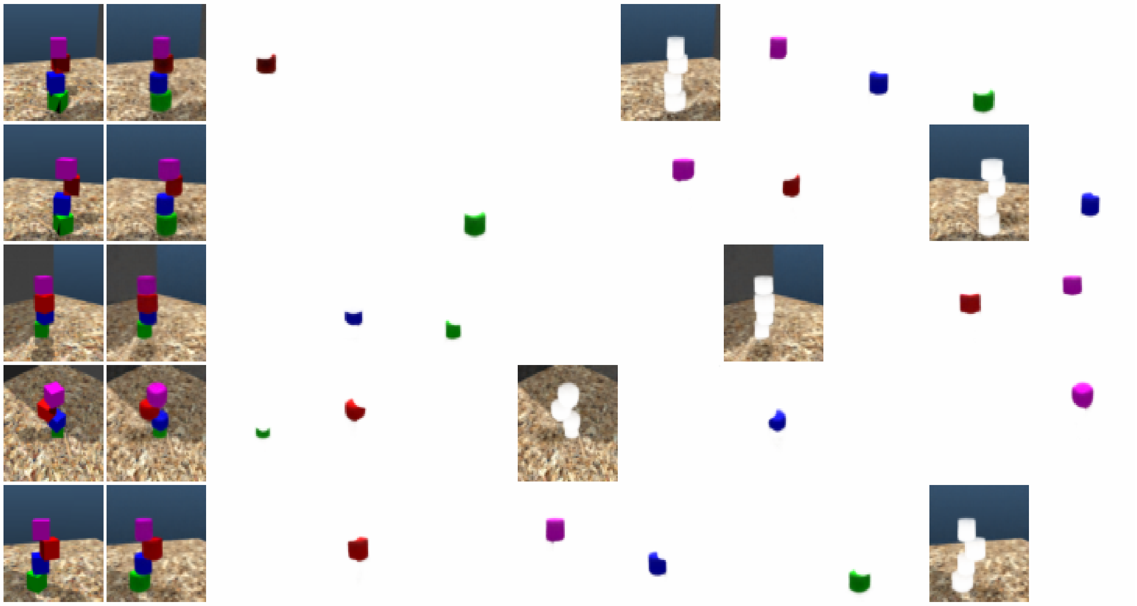

Figure 8. Additional decomposition resulst of Slot-VAE (ShapeStacks dataset).

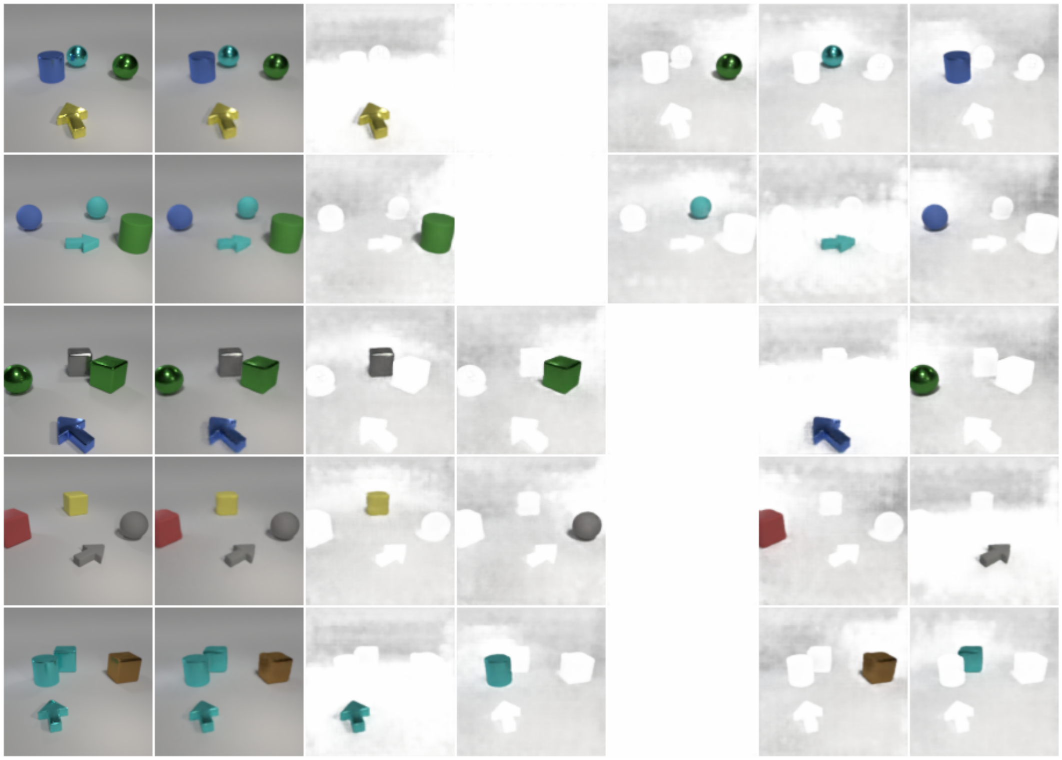

Figure 9. Additional decomposition resulst of Slot-VAE (ShapeStacks dataset).

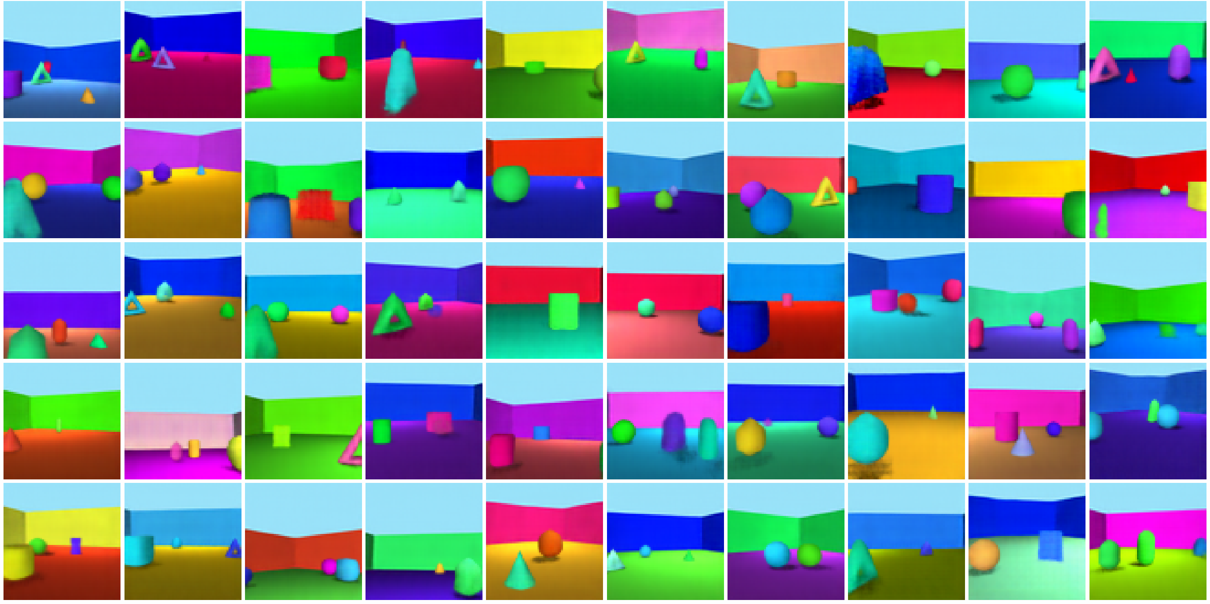

Figure 10. Additional generation resulst of Slot-VAE (Arrow Room dataset).

B. Implementation Details of Slot-VAE.

In this section, we introduce the implementation details of Slot-VAE. As shown in Fig. 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

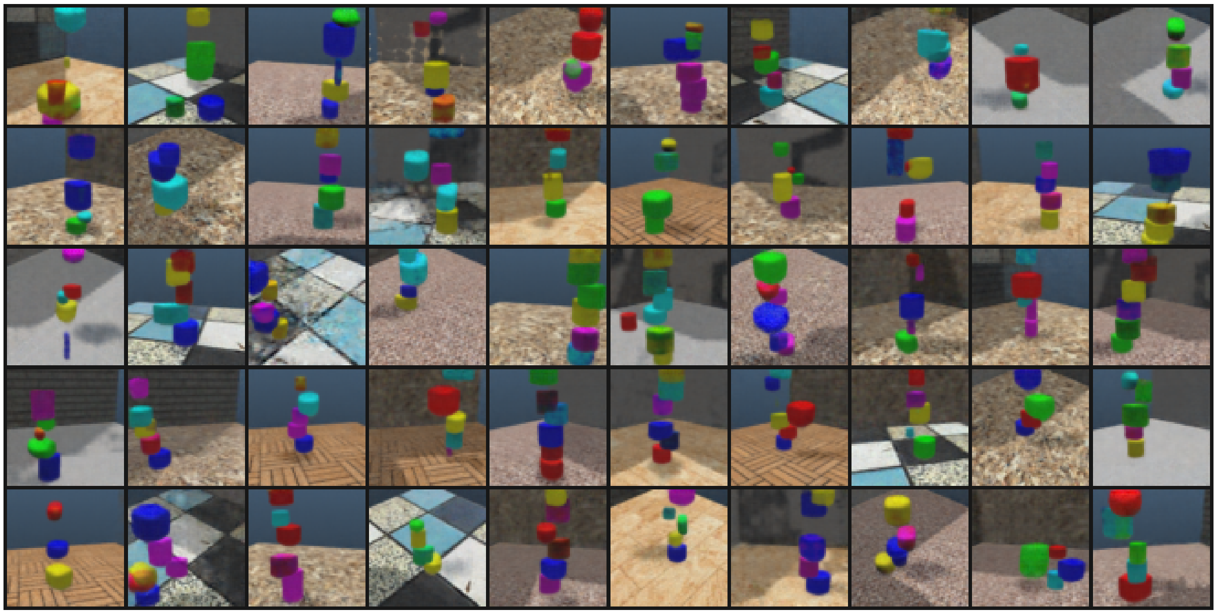

Figure 11. Additional generation resulst of Slot-VAE (ShapeStacks dataset).

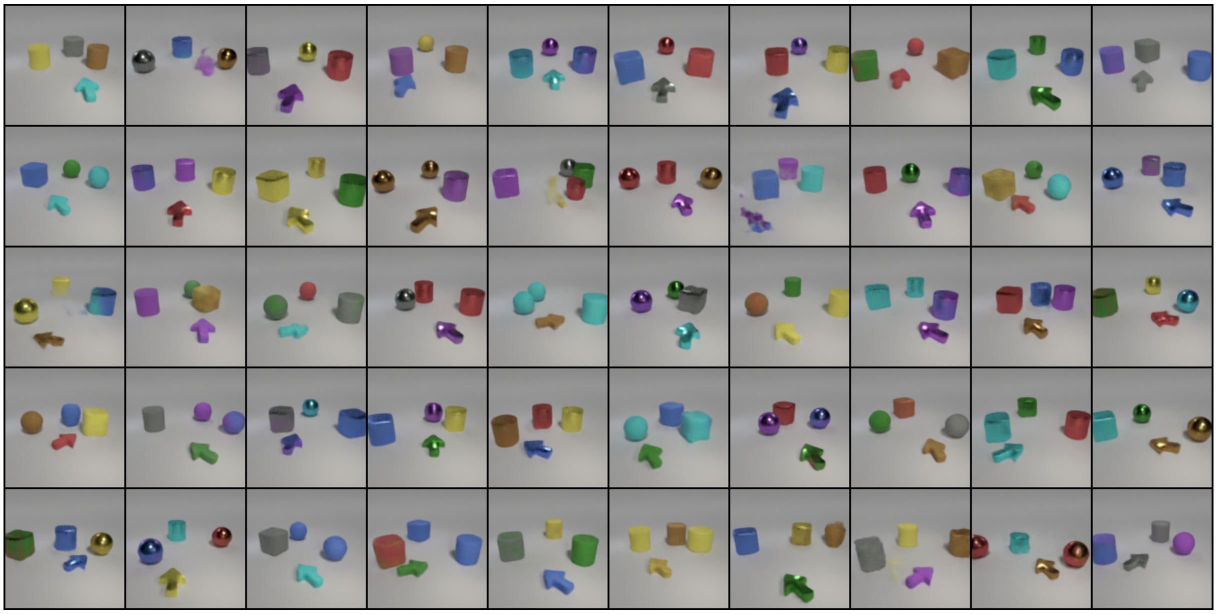

Figure 12. Additional generation resulst of Slot-VAE (Arrow Room dataset).

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 embedding are then added to the feature map to provide position information for the following modules.

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 self-containing purpuse. 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 :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 attn$_{i,j}$, image feature vectors $\mathbf{f}_{input}$ are aggregated via weighted mean: updates $:= \mathbf{W}^T \cdot v(\mathbf{f}_{input}) \in \mathbb{R}^{K\times D}$, where $\mathbf{W}_{i,j} :=$ attn$_{i,j}/(\sum_{l=1}^{N}$ 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): $\mathbf{s}_{1:K} \leftarrow$ GRU$(\mathbf{s}_{1:K},$ 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 $\mathbf{z}_k^s$. 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 objet representations. The obtained object-centric latent vector $\mathbf{z}_k^s$ 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 image, rather generated object-centric latent vectors $\mathbf{z}_k^s$, 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 $\mathbf{z}_k^s$ 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 $\mathbf{z}_k^s$.

Hyperparameter for the KL term $D_{\mathbf{KL}}\left[q_\phi(\mathbf{z}^g \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^g)\right]$. During training, we empirically find that multiplying $D_{\mathrm{KL}}\left[q_\phi(\mathbf{z}^g \mid \mathbf{x}) \mid\mid p_\theta(\mathbf{z}^g)\right]$ with a small hyperparameter β helps $\mathbf{z}^g$ to encode meaningful scene representations. When β is too large, $\mathbf{z}^g$ tends to totally collapse to $p_\theta(\mathbf{z}^g)$, i.e., normal distribution. In the experiments, for *ObjectRoom*, β is 0.01; for *ShapeStacks*, β is 0.1; and for *Arrow Room*, β is 0.1.

Training Details. Learning rate warm-up is important for object-centric representation learning as acknowledged by prior works. In the experiments, 10000 warm-up steps are used. For *ObjectRoom*, the batch size is 64, and the learning rate is 0.0004; for *ShapeStacks*, the batch size is 32, and the learning rate is 0.0001; and for *Arrow Room*, the batch size is 32, and the learning rate is 0.0001 in the early training steps and is decreased to 0.00005 after object-centric representations show up for stable training purpose.