Generative Scene Inference and Sampling with Object-Centric Latent Representations

Martin Engelcke
Applied AI Lab, University of Oxford
martin@robots.ox.ac.uk

Adam R. Kosiorek
Applied AI Lab & Dept. of Statistics
University of Oxford
adamk@robots.ox.ac.uk

Oiwi Parker Jones
Applied AI Lab, University of Oxford
oiwi@robots.ox.ac.uk

Ingmar Posner
Applied AI Lab, University of Oxford
ingmar@robots.ox.ac.uk

Abstract:
Generative models are emerging as promising tools in robotics and reinforcement learning. Yet, even though tasks in these domains typically involve distinct objects, most state-of-the-art methods do not explicitly capture the compositional nature of visual scenes. Two exceptions, MONet and IODINE, decompose scenes into objects in an unsupervised fashion via a set of latent variables. Their underlying generative processes, however, do not account for component interactions. Hence, neither of them allows for principled sampling of coherent scenes. Here we present GENESIS, the first object-centric generative model of visual scenes capable of both decomposing and generating complete scenes by explicitly capturing relationships between scene components. GENESIS parameterises a spatial GMM over pixels which is encoded by component-wise latent variables that are inferred sequentially or sampled from an autoregressive prior. We train GENESIS on two publicly available datasets and probe the information in the latent representations through a set of classification tasks, outperforming several baselines.

1 Introduction

Task execution in robotics and reinforcement learning requires accurate reasoning about elements of an environment and the ability to generalise to situations not encountered during training. While in some cases it is feasible to facilitate skill acquisition through corpora of manually collected labels for task-relevant objects (He et al. [1]), it is generally intractable to do so for every new task. As a consequence, deep generative models have gained in popularity due to their suitability to unsupervised training, for example, to improve sample efficiency in reinforcement learning (Ha and Schmidhuber [2], Racanière et al. [3]), to synthesise training instances for data augmentation (Lee et al. [4]) or to capture beliefs about the state of a scene (Gregor et al. [5]). Of particular interest recently have been models that are able to decompose visual scenes into meaningful objects without supervision (Burgess et al. [6], Greff et al. [7]).

Generative modelling, or probability density estimation, finds a parameterised distribution \( p_\theta(x) \) that explains the training data, sometimes via a set of latent variables \( z \). Typical tasks in robotics require interaction with distinct elements in an environment. Thus, \( p_\theta(x) \) should capture that visual scenes consist of individual components that can each be described with a concise latent code. Most state-of-the-art generative models of images, though, do not cater to this (Parmar et al. [8]). Burgess et al. [6] and Greff et al. [7] recently proposed two latent-variable models, MONet and IODINE, that decompose visual scenes into objects. However, neither provides an object-centric generative model of scenes that accounts for relationships between constituent parts: for example, two physical objects cannot occupy the same location. Consequently, it is not possible to employ these models in applications such as generating synthetic scenes for data augmentation in downstream tasks (Lee et al. [4]) or as a “world model” (Ha and Schmidhuber [2]).

Submitted to the 3rd Conference on Robot Learning (CoRL 2019).
Moreover, despite being motivated as a probabilistic generative model, MONet relies on a deterministic attention network and lacks a probabilistic inference mechanism. This supposedly makes the optimisation problem easier, but also implies that MONet cannot estimate \( p_\theta(x) \) and that it does not constitute a probabilistic generative model of visual scenes. In addition, the attention mechanism of MONet embeds a convolutional neural network (CNN) in a recurrent neural network (RNN), posing an issue in terms of scalability. These two considerations do not apply to IODINE, but IODINE employs a computationally expensive iterative refinement mechanism, also limiting its applicability.

Therefore, we propose GENERative Scene Inference and Sampling (GENESIS) which is, to the best of our knowledge, the first object-centric generative model of visual scenes that explicitly captures dependencies between scene components\(^1\). Crucially, this makes GENESIS significantly more suitable for various robotics and reinforcement learning applications. We demonstrate the ability of GENESIS to (i) decompose scenes into meaningful components, (ii) manipulate scenes, and (iii) generate novel scenes in a compositional fashion that resemble the training data. In GENESIS, relationships between scene components are modelled with an autoregressive prior that is learned alongside a sequential inference network. This sequential inference mechanism is performed in low-dimensional latent space, allowing all convolutional encoders and decoders to be run in parallel to better exploit modern graphics processing hardware.

We conduct experiments on two canonical and publicly available datasets: the GQN dataset (Esliami et al. [9]) and ShapeStacks (Groth et al. [10]). These simulated environments serve as testing grounds for navigation and object manipulation tasks, respectively. We use the scene annotations available for ShapeStacks to show the utility of object-centric latent representations for tasks such as predicting whether a block tower is stable or not, where GENESIS outperforms recent baselines.

We will release our PyTorch implementation and trained models for further community evaluation.

2 Related Work

This work draws inspiration from several different lines of research from the machine learning, robotics, and computer vision communities.

Applications of Environment Models Three use cases of environment models for sample efficient reinforcement learning are presented in Ha and Schmidhuber [2], Racanière et al. [3], and Gregor et al. [5]. Ha and Schmidhuber [2] are able to train an agent entirely inside a generative model, while Racanière et al. [3] enable agents to imagine different future outcomes to learn better policies. Gregor et al. [5] recently showed that generative models can capture beliefs about the state of the environment and that this can be leveraged to considerably improve the sample efficiency of reinforcement learning algorithms. In robotics, image editing and generating synthetic scenes are of particular interest as techniques for data augmentation. For example, Yao et al. [11] segment objects and subsequently render them in different poses. Alhaija et al. [12] overlay synthetic objects on real backgrounds and Lee et al. [4] edit scenes in pixel-wise semantic label space. Integrating GENESIS into methods such as these presents an exciting avenue for future research.

Structured Models While perception modules in robotics are typically trained in a supervised fashion (He et al. [1]), collecting labels for every new task is intractable. Several methods leveraging structured latent variables have been proposed to discover meaningful objects without supervision. Attend-Infer-Repeat (Eslami et al. [13]) and its sequential counterpart in Kosiorek et al. [14] use spatial attention to partition scenes into objects. Tagger (Greff et al. [15]) and Neural Expectation Maximisation (Greff et al. [16]) perform unsupervised segmentation by modelling images as spatial mixture models. Stacked Capsule Autoencoders (Kosiorek et al. [17]) discover geometric relationships between objects and their parts by using an affine-aware decoder. Yet, these approaches are limited with regards to the image complexity they can handle and none of these works demonstrate the ability to generate novel scenes with more complex, relational structure. While Xu et al. [18] present an extension of Eslami et al. [13] to generate images, their method only works for images with a uniform background and makes a prior assumption that object bounding boxes do not overlap. In contrast, we apply GENESIS to images from Eslami et al. [9] and Groth et al. [10] which contain simulated 3D scenes with complex backgrounds and considerable occlusion.

\(^1\)We use the terms “object” and “scene component” synonymously in this work.
**MONet & IODINE** While this work is most directly related to MONet (Burgess et al. [6]) and IODINE (Greff et al. [7]), it sets itself apart by introducing a generative model of complete visual scenes. MONet employs a deterministic, recurrent CNN to implement a stick-breaking process (SBP) that outputs pixel-wise attention masks for scene components. Subsequently, a variational auto-encoder (VAE) (Kingma and Welling [19], Rezende et al. [20]) with a spatial broadcast decoder (Watters et al. [21]) models individual components. Due to the deterministic attention network, MONet does not constitute a proper probabilistic generative model, limiting its applicability. IODINE uses a decoder and a refinement network with a computationally expensive, gradient-based iterative inference procedure. Structurally, GENESIS is more comparable to MONet and avoids the expensive iterative mechanism in IODINE. Unlike MONet, though, the convolutional encoders and decoders in GENESIS can be run in parallel, rendering the model computationally more scalable to inputs with a larger number of scene components. Both MONet and IODINE use an independent prior over scene components, whereas the autoregressive prior of GENESIS allows the modelling of relationships between components and the generation of coherent novel scenes.

**Adversarial Methods** A few recent works have proposed to use an adversary for scene segmentation and generation. While Chen et al. [22] and Bielski and Favaro [23] segment a single foreground object per image, Arandjelović and Zisserman [24] are able to segment several synthetic objects superimposed on natural images. Azadi et al. [25] introduce a model to combine two objects or an object and a background scene in a sensible fashion, whereas van Steenkiste et al. [26] can generate scenes with an arbitrary number of components. In comparison, GENESIS performs both inference and generation without exhibiting the instabilities common in adversarial training. Furthermore, the computational complexity of GENESIS increases with $\mathcal{O}(K^2)$, where $K$ is the number of components, compared to the $\mathcal{O}(K^3)$ complexity of the relational stage in van Steenkiste et al. [26].

### 3 GENESIS: Generative Scene Inference and Sampling

Akin to state-space representations, latent representations learned by generative models have been shown to improve sample efficiency of machine learning models in downstream tasks (van Steenkiste et al. [27]). Separating scene representations into object-centric ones facilitates relational reasoning (Santoro et al. [28]) and has the potential to further improve sample efficiency. Existing solutions, however, either require expensive inference procedures (Greff et al. [7, 15, 16]) or lack a proper probabilistic formulation (Burgess et al. [6]). Therefore, we introduce GENESIS which features a probabilistic formulation and an efficient inference procedure. Moreover, GENESIS has an autoregressive prior which is capable of learning relationships between scene components. We first describe the generative model of GENESIS, as well as a simplified variant called GENESIS-S, and follow by detailing the associated inference procedure. See Figure 1 for an overview of GENESIS and Figure 2 for the graphical model in comparison to alternative methods. GENESIS-S is illustrated in Figure 7 in Appendix A.1.

**Generative model** Let $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ be an image. We formulate the problem of image generation as a spatial gaussian mixture model (GMM). That is, every Gaussian component $k = 1, \ldots, K$, where $K \in \mathbb{N}_+$ is the maximum number of scene components, represents an image-sized scene component $\bar{x}_k$. The corresponding image-sized mixing probability $\pi_k$ represents whether the component $k$ is present at a location in the image. The mixing probabilities are normalised across scene components, i.e. $\forall_{i,j} \sum_k \pi_{i,j,k} = 1$, and can be regarded as pixel-wise attention masks. Since there are strong spatial dependencies between different scene components, we formulate an autoregressive prior distribution over mask variables $z^m_k \in \mathbb{R}^{D_m}$ which encode the mixing probabilities $\pi_k$, as

$$p_\theta(z^m_{1:k}) = \prod_{k=1}^{K} p_\theta(z^m_k | z^m_{1:k-1}) = \prod_{k=1}^{K} p_\theta(z^m_k | u_k) | u_k = R_\theta(z^m_{k-1}, u_{k-1}) \quad .$$

The dependence on previous latents $z^m_{1:k-1}$ is implemented via an RNN $R_\theta$ with hidden state $u_k$.

Next, we assume that the appearance of scene components $\bar{x}_k$ is conditionally independent given their spatial allocation in the scene. The corresponding conditional distribution over component variables $z^c_k \in \mathbb{R}^{D_c}$ which encode component appearances factorises as follows,

$$p_\theta(z^c_{1:k} | z^m_{1:k}) = \prod_{k=1}^{K} p_\theta(z^c_k | z^m_k) \quad .$$

3
with parameters $\phi$.

Approximate posterior We amortise inference by using an approximate posterior distribution $q_\phi(z_{1:K}^m \mid x)$, which is similar to the generative model. The full approximate posterior reads as follows,

$$q_\phi(z_{1:K}^c \mid x) = q_\phi(z_{1:K}^m \mid x) q_\phi(z_{1:K}^c \mid x, z_{1:K}^m),$$

$$q_\phi(z_k^c \mid x) = \prod_{k=1}^K q_\phi(z_k^c \mid x, z_{1:k-1}^m), \quad \text{and} \quad q_\phi(z_k^m \mid x, z_{1:k}^m) = \prod_{k=1}^K q_\phi(z_k^m \mid x, z_{1:k}^m),$$

with the dependence on $z_{1:k-1}^m$ realised by an RNN $R_\phi$. The RNN could, in principle, be shared with the prior, but we have not investigated this option. All conditional distributions are Gaussian. For GENESIS-S, the approximate posterior takes the form $q_\phi(z_{1:K} \mid x) = \prod_{k=1}^K q_\phi(z_k \mid x, z_{1:k-1}).$
Learning  GENESIS can be trained by maximising the evidence lower bound (ELBO) on the log-marginal likelihood $\log p_\theta(x)$, given by

$$
L_{\text{ELBO}}(x) = \mathbb{E}_{q_\phi(z^c, z^m \mid x)} \left[ \log p_\theta(x \mid z^c, z^m) p_\theta(z^c \mid z^m) p_\theta(z^m) \right] q_\phi(z^m \mid x) q_\phi(z^c \mid z^m, x) q_\phi(z^m) \right] - \text{KL} (q_\phi(z^c, z^m \mid x) \mid \mid p_\theta(z^c, z^m)).
$$

However, this often leads to a strong emphasis on the likelihood term (first term in Equation (9)), while allowing the marginal approximate posterior $q_\phi = \mathbb{E}_{p_{\text{data}}(x)} [q_\phi(z \mid x)]$ to drift away from the prior distribution, hence increasing the KL-divergence (second term in Equation (9)). This also decreases the quality of samples drawn from the model.

To prevent this behaviour, we use the generalised ELBO with constrained optimisation ( GE CO) objective from Rezende and Viola [29] instead, which changes the learning problem to minimising the KL-divergence subject to a reconstruction constraint. Let $C \in \mathbb{R}$ be the minimum allowed reconstruction log-likelihood, GE CO then uses Lagrange multipliers to solve the following problem,

$$
\theta^*, \phi^* = \arg \min_{\theta, \phi} \text{KL} (q_\phi(z^c, z^m \mid x) \mid \mid p_\theta(z^c, z^m))
$$

such that $\mathbb{E}_{q_\phi(z^c, z^m \mid x)} [\log p_\theta(x \mid z^c, z^m)] \geq C$.

4 Experiments

In this section, we present qualitative and quantitative results on the “rooms-ring-camera” dataset from GQN (Eslami et al. [9]) and the ShapeStacks dataset (Groth et al. [10]). We use an image resolution of 64-by-64 for all experiments. The number of steps was set to $K = 7$ for GQN and $K = 9$ for ShapeStacks. Additional results are included in the Appendix, also illustrating a typical failure case of the model.

GQN The “rooms-ring-camera” dataset includes simulated 3D scenes of a square room with different floor and wall textures, containing one to three objects of various shapes and sizes.

ShapeStacks Images show simulated block towers of different heights (two to six blocks) where individual blocks have different shapes, sizes, and colours. Scenes have annotations for: stability of the tower (binary), number of blocks (two to six), properties of individual blocks, locations in the tower of centre-of-mass violations and planar surface violations, wall and floor textures (five each), light presets (five), and camera view points (sixteen).

4.1 Implementation Details

For the results in this section, we use the architecture from Berg et al. [30] to encode and decode $z^m_k$. This architecture was modified for GENESIS by applying batch normalisation (Ioffe and Szegedy [31]) before the GLUs (Dauphin et al. [32]). The convolutional layers in the encoder and decoder have five layers with size-5 kernels, strides of [1, 2, 1, 2, 1], and filter sizes of [32, 32, 64, 64, 64] and [64, 32, 32, 32, 32], respectively. Fully-connected layers are used at the lowest resolution.
The encoded image is passed to a long short-term memory (LSTM) cell (Hochreiter and Schmidhuber [33]) followed by a linear layer to compute the mask latents $z_k^{m}$ of size 64. The LSTM state size is twice the latent size. Importantly, unlike the analogous counterpart in MONet, the decoding of $z_k^{c}$ is performed in parallel. The autoregressive prior $p_{\theta}(z_k^{c} | z_{1:k-1}^{m})$ is implemented as an LSTM with 256 units. The conditional distribution $p_{\theta}(z_k^{m} | z_{1:k-1}^{m})$ is parameterised by a multilayer perceptron (MLP) with two hidden layers, 256 units per layer, and ELUs (Clevert et al. [34]). We use the same component VAE featuring a spatial broadcast decoder as MONet to encode and decode $z_k^{c}$, but we replace ReLUs (Glorot et al. [35]) with ELUs.

For GENESIS-S, the encoder of $z_k$ is the same as for $z_k^{m}$ above and the decoder from Berg et al. [30] is again used to compute the mixing probabilities. However, GENESIS-S also has a second decoder with spatial broadcasting to obtain the object appearances $x_k$ from $z_k$. We found the use of two different decoders to be important for GENESIS-S to learn meaningful scene decompositions. While this shows that it is not necessary to use separate latent variables for component masks and appearances in order to decompose scenes, we found that GENESIS consistently trained more quickly with better qualitative results. We conjecture that this gap might close to some degree when training the models for more iterations. Hence, qualitative results reported in this section are from GENESIS.

All models are trained for $5 \times 10^5$ iterations with a batch size of 32 using the ADAM optimiser (Kingma and Ba [36]) and a learning rate of $10^{-4}$. With these settings, training GENESIS takes about two days on a single GPU, though we expect performance to improve with further training. We deliberately choose a comparatively weak reconstruction constraint for the GECO objective to emphasise KL minimisation and sample quality. We increase the GECO step size by a constant factor when the reconstruction constraint is satisfied to accelerate training.

### 4.2 Scene Generation

Unlike previous works, GENESIS has an autoregressive prior to capture intricate dependencies between scene components. Modelling these relationships is necessary to generate coherent scenes. For example, different parts of the background need to fit together; we do not want to create components several times that can only be present once, such as the sky; and several physical objects cannot be in the same location. GENESIS is able to generate novel scenes by sequentially sampling scene components from the prior and conditioning each new component on those that have been generated during previous steps.

Figure 3 shows two scenes generated by GENESIS after training on GQN. At every step, either an object in the foreground or a part of the background is generated. Crucially, these components fit together to make up a semantically consistent scene that looks similar to the training data.

![Figure 3: Two scenes generated in a component-wise fashion by GENESIS after training on the GQN dataset. The first pane shows the final image and the subsequent panes show the different scene components generated at each step. The model first generates the sky and the floor, followed by individual objects, and finally distinct parts of the wall in the background.](image-url)
4.3 Scene Decomposition

Similar to MONet and IODINE, GENESIS is able to segment scenes into meaningful components, with some slots reconstructing individual foreground objects and other slots reconstructing different parts of the background. Step-by-step reconstructions for a model trained on the GQN dataset are shown in Figure 4. The model starts by reconstructing part of the background in the first step and individual objects are reconstructed in separate steps.

We observed that the model pursues a similar decomposition strategy for all input images. Models trained with different hyperparameters also typically learn the same strategy. We hypothesise that this is due to the fact that large surfaces with little variation in appearance have a big influence on the reconstruction loss and are also easy to learn compared to more involved parts of the scene with finer details. In particular, the mask latents in the first step only depend on the input, whereas later masks depend on previous steps which could introduce distracting noise early on in training.

In Figure 5, we show the step-by-step reconstructions on ShapeStacks. Again, the model learns a similar reconstruction strategy as for GQN. Notably though, GENESIS is able to differentiate between all five objects in the block tower compared to the smaller number of objects in GQN even though the background is quite noisy and the objects in the block tower occlude each other significantly.

4.3.1 Scene Manipulation

Learning disentangled representations (van Steenkiste et al. [27]) combined with the ability to decompose scenes enables GENESIS to deliberately manipulate selected parts of the scene while leaving the rest unchanged. This is achieved by linearly traversing the latent space of individual component latents. This can be used, for example, to change the appearance of the background or any of the foreground objects as illustrated in Figure 6.
4.4 Evaluating Unsupervised Representations on ShapeStacks

We employ a set of discriminative tasks to evaluate how well the representations learned by GENESIS capture various properties of input scenes and compare to several VAE and MONet baselines. Alternative metrics such as ELBO or negative log-likelihood cannot be computed for MONet. To this end, ShapeStacks is an interesting testing ground as scenes contain multiple objects with surface contact. In particular, we consider three tasks: (1) Is a tower stable or not? (2) What is the tower’s height in terms of the number of blocks? (3) What is the quantised camera pose (out of 16 possibilities)? The first two tasks focus on information relating to foreground objects in the scene and the third task was inspired by Eslami et al. [9].

We compare against four baselines: a VAE with a deconvolutional decoder (DeconvVAE), a VAE with a spatial broadcast decoder (BroadcastVAE), and two different variants of MONet with either [32, 32, 64, 64, 64] (MONet32) or [64, 64, 128, 128, 128] (MONet64) filters in the attention encoder and the reverse in the decoder. The VAEs have 64 latents and the same encoder (and decoder in case of the DeconvVAE) as in Berg et al. [30]. The BroadcastVAE has the same decoder with ELUs as GENESIS, but with twice the number of filters to enable a better comparison. For the MONet variants, the prior on the masks was normalised with a softmax function to compute the KL-divergence.

MLPs with one hidden layer, 512 units, and ELU activations were used for classification. The classifiers were trained for 100 epochs on 50,000 labelled examples with a batch size of 128 using a cross-entropy loss, the ADAM optimiser, and a learning rate of $10^{-4}$. As inputs to the classifiers, we concatenate $z^m_k$ and $z^c_k$ for GENESIS, $z^c_k$ for GENESIS-S, and the component VAE latents for the two MONet variants. Table 1 shows the test accuracy of the models. The first row of the table shows the accuracy obtained by always predicting the class with the largest number of examples in the test set.

| Model       | Stability | Height  | View  |
|-------------|-----------|---------|-------|
| Largest class | 50.0      | 30.5    | 6.25  |
| DeconvVAE   | 58.7      | 67.8    | 98.7  |
| BroadcastVAE| 59.2      | 77.5    | 99.8  |
| MONet32     | 59.3      | 85.0    | 99.1  |
| MONet64     | 60.4      | 86.2    | 99.4  |
| GENESIS     | 65.2      | 81.5    | 98.8  |
| GENESIS-S   | 63.5      | 78.0    | 99.8  |

None of models are able to reach the stability prediction accuracies in Groth et al. [10]. This is not surprising considering that the representation networks were trained on sub-sampled images without data augmentation and with a pixel-wise reconstruction loss. Notably, however, both GENESIS and MONet do much better in terms of predicting tower stability and height than the VAE baselines, indicating that object-centric latents are beneficial for these tasks. MONet performs particularly well on predicting the height of the towers which might be facilitated by its deterministic segmentation network. Nevertheless, both variants of GENESIS outperform all baselines in terms of stability prediction. All models do well at predicting the camera view.

5 Conclusions

In this work, we propose a novel object-centric latent variable model of scenes called GENESIS. We show that GENESIS is, to the best of our knowledge, the first unsupervised model to both decompose visual scenes into semantically meaningful constituent parts, while at the same time being able to generate novel coherent scenes in a component-wise fashion. Importantly, this is achieved by capturing relationships between scene components with an autoregressive prior, allowing complete scenes to be modelled as a collection of components. Regarding future work, the most interesting challenge is to scale GENESIS to more complex datasets. Another potentially promising research direction is to adapt the formulation to only model parts of the scene that are relevant for a certain task. We hope that this work will open up promising avenues for further research, in particular with regards to learning and planning in latent space as well as better harnessing the capabilities of generative models in robotics and reinforcement learning applications.
Acknowledgments

This research was supported by an EPSRC Programme Grant (EP/M019918/1). The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work, http://dx.doi.org/10.5281/zenodo.22558, and the use of Hartree Centre resources. The authors thank Yizhe Wu for his help with re-implementing MONet.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. *International Conference on Computer Vision*, 2017.

[2] D. Ha and J. Schmidhuber. World Models. *Neural Information Processing Systems*, 2018.

[3] S. Racanière, T. Weber, D. Reichert, L. Buesing, A. Guez, D. J. Rezende, A. P. Badia, O. Vinyals, N. Heess, Y. Li, et al. Imagination-Augmented Agents for Deep Reinforcement Learning. *Neural Information Processing Systems*, 2017.

[4] D. Lee, S. Liu, J. Gu, M.-Y. Liu, M.-H. Yang, and J. Kautz. Context-Aware Synthesis and Placement of Object Instances. *Neural Information Processing Systems*, 2018.

[5] K. Gregor, D. J. Rezende, F. Besse, Y. Wu, H. Merzic, and A. v. d. Oord. Shaping Belief States with Generative Environment Models for RL. *arXiv preprint arXiv:1906.09237*, 2019.

[6] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner. MONet: Unsupervised Scene Decomposition and Representation. *arXiv preprint arXiv:1901.11390*, 2019.

[7] K. Greff, R. L. Kaufmann, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. Multi-Object Representation Learning with Iterative Variational Inference. *International Conference on Machine Learning*, 2019.

[8] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image Transformer. *International Conference on Machine Learning*, 2018.

[9] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, et al. Neural Scene Representation and Rendering. *Science*, 2018.

[10] O. Groth, F. B. Fuchs, I. Posner, and A. Vedaldi. ShapeStacks: Learning Vision-Based Physical Intuition for Generalised Object Stacking. *European Conference on Computer Vision*, 2018.

[11] S. Yao, T. M. Hsu, J.-Y. Zhu, J. Wu, A. Torralba, B. Freeman, and J. Tenenbaum. 3D-Aware Scene Manipulation via Inverse Graphics. *Neural Information Processing Systems*, 2018.

[12] H. A. Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes. *International Journal of Computer Vision*, 2018.

[13] S. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, G. E. Hinton, et al. Attend, Infer, Repeat: Fast Scene Understanding with Generative Models. *Neural Information Processing Systems*, 2016.

[14] A. Kosiorek, H. Kim, Y. W. Teh, and I. Posner. Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects. *Neural Information Processing Systems*, 2018.

[15] K. Greff, A. Rasmus, M. Berglund, T. Hao, H. Valpola, and J. Schmidhuber. Tagger: Deep Unsupervised Perceptual Grouping. *Neural Information Processing Systems*, 2016.

[16] K. Greff, S. van Steenkiste, and J. Schmidhuber. Neural Expectation Maximization. *Neural Information Processing Systems*, 2017.

[17] A. R. Kosiorek, S. Sabour, Y. W. Teh, and G. E. Hinton. Stacked Capsule Autoencoders. *arXiv preprint arXiv:1906.06818*, 2019.
[18] K. Xu, C. Li, J. Zhu, and B. Zhang. Multi-Objects Generation with Amortized Structural Regularization. *Neural Information Processing Systems*, 2018.

[19] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*, 2014.

[20] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *International Conference on Machine Learning*, 2014.

[21] N. Watters, L. Matthey, C. P. Burgess, and A. Lerchner. Spatial Broadcast Decoder: A Simple Architecture for Learning Disentangled Representations in VAEs. *arXiv preprint arXiv:1901.07017*, 2019.

[22] M. Chen, T. Artiètes, and L. Denoyer. Unsupervised Object Segmentation by Redrawing. *arXiv preprint arXiv:1905.13539*, 2019.

[23] A. Bielski and P. Favaro. Emergence of Object Segmentation in Perturbed Generative Models. *arXiv preprint arXiv:1905.12663*, 2019.

[24] R. Arandjelović and A. Zisserman. Object Discovery with a Copy-Pasting GAN. *arXiv preprint arXiv:1905.11369*, 2019.

[25] S. Azadi, D. Pathak, S. Ebrahimi, and T. Darrell. Compositional GAN: Learning Image-Conditional Binary Composition. *arXiv preprint arXiv:1807.07560*, 2019.

[26] S. van Steenkiste, K. Kurach, and S. Gelly. A Case for Object Compositionality in Deep Generative Models of Images. *NeurIPS Workshop on Modeling the Physical World: Learning, Perception, and Control*, 2018.

[27] S. van Steenkiste, F. Locatello, J. Schmidhuber, and O. Bachem. Are Disentangled Representations Helpful for Abstract Visual Reasoning? *arXiv preprint arXiv:1905.12506*, 2019.

[28] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. W. Battaglia, and T. P. Lillicrap. A Simple Neural Network Module for Relational Reasoning. *Neural Information Processing Systems*, 2017.

[29] D. J. Rezende and F. Viola. Taming VAEs. *arXiv preprint arXiv:1810.00597*, 2018.

[30] R. v. d. Berg, L. Hasenclever, J. M. Tomczak, and M. Welling. Sylvester Normalizing Flows for Variational Inference. *Conference on Uncertainty in Artificial Intelligence*, 2018.

[31] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning*, 2015.

[32] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language Modeling with Gated Convolutional Networks. *International Conference on Machine Learning*, 2017.

[33] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 1997.

[34] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *International Conference on Learning Representations*, 2016.

[35] X. Glorot, A. Bordes, and Y. Bengio. Deep Sparse Rectifier Neural Networks. *International Conference on Artificial Intelligence and Statistics*, 2011.

[36] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 2015.

[37] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv preprint arXiv:1607.08022*, 2016.
A Appendix

A.1 GENESIS-S

![Diagram](https://via.placeholder.com/150)

Figure 7: GENESIS-S overview. Given an image $x$, an encoder and an RNN compute latent variables $z_k$. These are decoded to directly obtain the mixing probabilities $\pi_k$ and the scene components $\hat{x}_k$.

A.2 Architecture Variations

We investigate several architecture variations of GENESIS and GENESIS-S and compare their performances on the ShapeStacks tasks.

The basic GatedConv architecture is adapted from Berg et al. [30] as described in Section 4.1. Here, we report results for three variants of GENESIS and GENESIS-S with the following modifications:

- increase the dimension of the mask latents from 64 to 128 and reduce the number of filters in the mask decoder by a factor of two;
- use four stride-2 layers with [32, 32, 64, 64] and [64, 32, 32, 32] filters in the encoder and decoder, respectively, and use instance normalisation (IN) (Ulyanov et al. [37]) instead of batch normalisation (BN);
- use the attention architecture from MONet32 without skip-connections and replace ReLU activations with ELUs.

The results of are summarised in Table 2. Overall, we found that performance on ShapeStacks was not strongly affected by these architecture changes. Both GENESIS and GENESIS-S still perform better on the stability prediction task than the baselines in Table 1. It would be interesting to conduct a more comprehensive study to establish best practices with regards to architecture design for these types of models, in particular in consideration of performance vs. run-time trade-offs.

| Model | $z^m_k$ dim | Architecture | Norm | Stability | Height | View |
|-------|-------------|--------------|------|-----------|--------|------|
| GENESIS | 64 | GatedConv | BN | 65.2 | 81.5 | 98.8 |
| 128 | GatedConv, 1/2 channels in decoder | BN | 64.3 | 79.8 | 98.6 |
| 128 | GatedConv, four stride-2 layers | IN | 61.0 | 77.1 | 99.6 |
| 128 | MONet32-like | IN | 63.1 | 78.8 | 99.6 |
| GENESIS-S | 64 | GatedConv | BN | 63.5 | 78.0 | 99.8 |
| 128 | GatedConv, 1/2 channels in decoder | BN | 64.1 | 81.7 | 99.7 |
| 128 | GatedConv, four stride-2 layers | IN | 64.4 | 79.0 | 99.8 |
| 128 | MONet32-like | IN | 62.6 | 77.2 | 99.9 |
A.3 Scene Generation

Figure 8: Step-by-step generation of a synthetic scene by GENESIS after training on GQN, showing the unmasked component appearances in row two, the masks of the mixing probabilities in row three, and the “scope” of the stick-breaking process which represents pixels in the image that still need to be explained. Notably, the model decides to not use the second reconstruction step as there is only one object in the foreground.
A.4 Scene Decomposition

Figure 9: From left to right, the eight panels show: the input scene from GQN presented to the model, the final model reconstruction, the first reconstruction step which attends to the floor and the sky, the second step which attends to the yellow shape, the third step which attends to the purple shape, and three further steps that attend to different parts of the background.
Figure 10: De-rendering an image from the ShapeStacks validation set. Despite a noisy background and considerable occlusion, GENESIS is able to segment all five shapes in the block tower. It interesting to observe how the component appearances in steps three to eight all capture the light conditions of the scene, even though the top surfaces of the blocks below the top block are mostly occluded.
Figure 11: Failure case on the ShapeStacks dataset. Especially when there are a large number of objects in the block tower, in this example there are six, GENESIS sometimes explains several objects in a single reconstruction step as in steps two, three, and four for this input image.