The Multi-Entity Variational Autoencoder

Charlie Nash[1,2]*, **S. M. Ali Eslami**[2], **Chris Burgess**[2], **Irina Higgins**[2],
Daniel Zoran[2], **Theophane Weber**[2], **Peter Battaglia**[2]
[1]Edinburgh University [2]DeepMind

Abstract

Representing the world as objects is core to human intelligence. It is the basis of people's sophisticated capacities for reasoning, imagining, planning, and learning. Artificial intelligence typically assumes human-defined representations of objects, and little work has explored how object-based representations can arise through unsupervised learning. Here we present an approach for learning probabilistic, object-based representations from data, called the "multi-entity variational autoencoder" (MVAE), whose prior and posterior distributions are defined over a *set* of random vectors. We demonstrate that the model can learn interpretable representations of visual scenes that disentangle objects and their properties.

1 Introduction

Human intelligence is object-oriented. Infants begin parsing the world into distinct objects within their first months of life [13], and our sophisticated capacities for reasoning, imagining, planning, and learning depend crucially on our representation of the world as dynamic objects and their relations. Though the human notion of an object is rich, and exists in an even richer continuum of non-solids, non-rigids, object parts, and multi-object configurations, here we use the term "object" simply as a discrete visual entity localized in space.

Many important domains of artificial intelligence use representations of objects that were chosen ahead of time by humans, based on subjective knowledge of what constitutes an object (e.g. patches in images that can be categorized, or geometric meshes for physical control). This core object knowledge was learned through evolution and experience, and is very useful. It can be shared across object instances, provides a means for some properties of the world to be highly dependent and others to be relatively independent, and allows objects to be composed to form abstractions and hierarchies whose wholes are greater than the sums of their parts. Given the importance of such representations, and the high cost of manually translating it from the engineer's mind into AI datasets and architectures, this work asks: How can an artificial system learn, without supervision, an object-based representation?

Our contributions are: (1) a probabilistic model that can learn object-based representations from data, (2) a visual attention mechanism for inferring a sparse set of objects from images.

2 Multi-entity VAE

The multi-entity VAE (MVAE) is a latent variable model of data $\mathbf{x}$ in which the latent space is factored into a set of N independent 'object' representations $\{\mathbf{z}_1, \dots, \mathbf{z}_N\}$. The MVAE defines a generative process in which each $\mathbf{z}_n$ $(n = 1, ..., N)$ is sampled independently from a prior distribution, $p(\mathbf{z})$, and data examples are sampled from a decoder distribution $p(\mathbf{x}|\mathbf{z})$.

In our visual experiments, the MVAE model assumes $p(\mathbf{z}_n)$ is a D-dimensional Gaussian with zero mean and unit variance. The conditional data distribution is implemented as a three-step

[*] Work done during an internship at DeepMind

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

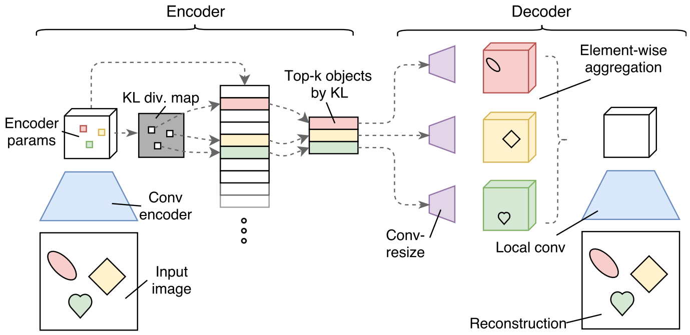


Figure 1: **Multi-entity VAE.** The encoder takes input images and produces a spatial map of posterior parameters. The KL-divergence of the posterior distribution against the prior is computed in each spatial location. The top-N posterior parameters by KL-divergence are selected from the spatial map, removing the spatial structure. These distributions are sampled, and the samples are passed independently through a shared convolution / upsampling network. The resulting object feature maps are aggregated using an element-wise operation, and a final convolutional network produces the output parameters.

deterministic decoding function, $\mathbf{f}$, which first maps each latent object representation to a processed object representation using a shared function, aggregates the processed object representations together, and deterministically transforms the result into the parameters of a Bernoulli distribution over pixel values. Crucially, $\mathbf{f}$ is permutation invariant with respect to the set of object representations. This encourages the model to learn object representations that are consistent and interchangeable.

Shared object processing. In the first stage of the decoder a shared function $\mathbf{g} : \mathbb{R}^D \to \mathbb{R}^K$ is applied independently to each latent object representation, resulting in a set of processed object descriptions $\mathbf{o}_n = \mathbf{g}(\mathbf{z}_n)$, $n = 1, \ldots, N$. These deterministic transformations of the prior latent variables are themselves random variables, which have dependencies induced by the prior latents. The K-dimensional object descriptions could be of any shape, but in this work we used 3D tensors as a structural bias towards representations of visual attributes. We implement $\mathbf{g}$ as a network that transforms each latent vector to a 3D tensor via reshaping, convolutions and upsampling.

Aggregation. The processed object descriptions $\mathbf{o}_{1:N}$ are aggregated using a symmetric pooling function, to form $\mathbf{o}_{\text{pool}}$, a tensor with the same shape as each of $\mathbf{o}_{1:N}$. In our experiments we used element-wise sum or max as aggregation functions.

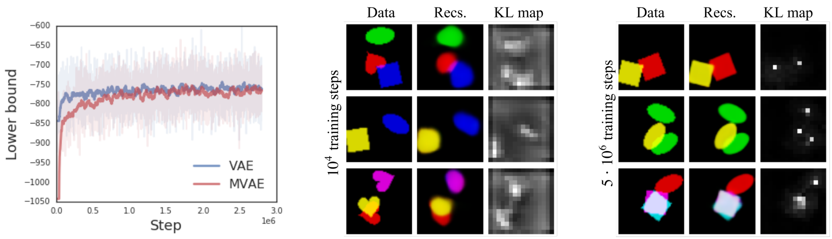
Rendering. After pooling, the resulting $\mathbf{o}_{\text{pool}}$ is mapped (i.e. rendered) to the element-wise parameters of the decoder distribution $\boldsymbol{\theta} = \mathbf{h}(\mathbf{o}_{\text{pool}})$. In our experiments $\mathbf{o}_{\text{pool}}$ is a 3D tensor, and $\mathbf{h}$ is a convolutional, upsampling network which outputs pixel-wise Bernoulli logits.

2.1 Maximal information attention

We employ amortized variational inference and learn a parameterized approximate posterior $q(\mathbf{z}_n|\mathbf{x})$ for each latent object representation. Unlike prior work [3, 4], we do not employ a learned attention mechanism in order to localise objects, but instead generate a large collection of candidate object inferences, from which N objects are selected. This inference method has the advantage that it circumvents the need for an explicitly learned attention mechanism, which may require a large number of recurrent passes over the image. This enables us to model scenes with large numbers of objects, something that was challenging in prior work.

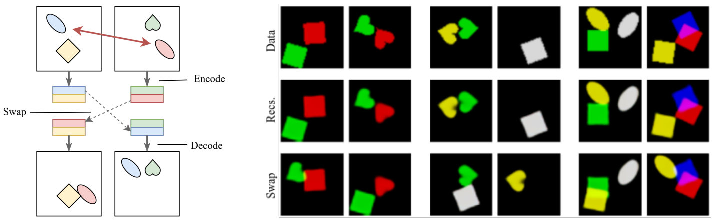
Candidate generation. We generate candidate object inferences for visual scenes using a convolutional-network which maps input images to a grid of posterior parameters. Each spatial location in this output feature map is treated as an object, and we perform candidate sub-selection of this set as described in the next section. After sub-selection the spatial structure present in the convolutional grid is destroyed, so we tag each object with its relative spatial coordinates at an intermediate feature map in the convolutional network.

Candidate sub-selection. Given a collection of candidate posteriors $\{q(\mathbf{z}_s|\mathbf{x})\}_s$ we compute the KL divergence $D^s_{\text{kl}} = D_{\text{kl}}[q(\mathbf{z}_s|\mathbf{x})|p(\mathbf{z})]$ for each candidate s. Approximate posteriors for the N



(a) (b)

Figure 2: **Training.** (a) Training curves with exponential moving averages for the MVAE and a standard convolution VAE. (b) Model reconstructions and KL-divergence spatial maps at early and late stages of training. The KL maps are diffuse early in training and become increasingly sharp during optimization.



(a) (b)

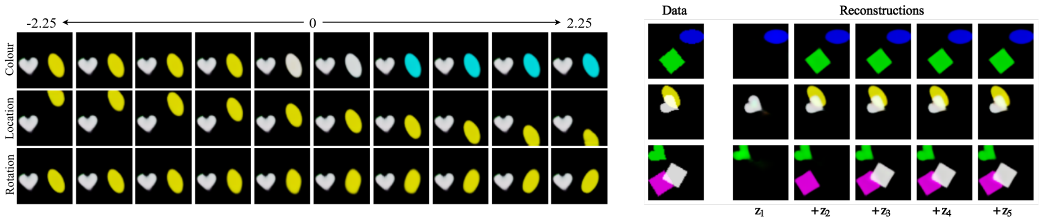
Figure 3: **Entity exchange.** (a) For a pair of input images, we encode to a set of latent objects before exchanging the representations of one objects in each pair. (b) Input pairs, model reconstructions, and reconstruction with exchanged entities. Here the objects in each input image with highest KL divergence are swapped.

latent objects are obtained by choosing the top-N objects by KL divergence. The intuition for this process is as follows: In order to reconstruct the input the network must encode lots of information in locations where objects exist like their shapes, colours, etc., whereas much less information is needed to encode background information; simply that there is no object present there. As such the "object" and "non-object" locations will have high and low KL-divergence respectively, and by choosing the top locations by KL-divergence we encode information only in the most informative regions of the image. We call this process maximal-information attention, and note that it can be used for any data modality where a superset of candidate inferences can be generated.

The candidate generation and sub-selection results in approximate posteriors for the N object representations, which we then sample from and pass to the decoder as in a standard VAE.

3 Related work

Our MVAE builds on previous neural probabilistic generative models, especially variational autoencoders (VAEs) [6, 10]. The DC-IGN [7], beta-VAE [5], and InfoGAN [1] are extensions and alternatives that promote learning latent codes whose individual features are “disentangled” [14, 2], i.e. correlated exclusively with underlying axes in the data-generating process. Other work has developed attention mechanisms for sampling subsets of visual scenes [9, 4, 15], which promotes learned representations that are spatially localized. Recurrent neural networks have been used in combination with spatial attention to allow for unsupervised learning of object locations and counts [3]. And several recent approaches allow object-like representations to be learned in supervised settings for visual question-answering and physical prediction [12, 16]. Another related strand of work focuses on explicit representations of multiple objects in visual scenes, making use of graphics engines as a known decoding function [17, 11].



(a)

(b)

Figure 4: (a)**Within-object latent traversals.** Decoded scenes for traversals from -2.25 to 2.25 for a selection of latent dimensions for a single object. The latent variables associated with different objects are held fixed during the process. (b) **Decoding objects sequentially.** (left) Data examples. (right) Reconstructions of the data where one latent object is added at each step. Here the latent objects $\mathbf{z}_1, \ldots, \mathbf{z}_5$ are ordered by the KL-divergence of the encoding distributions.

4 Experiments

We evaluate our model on a multiple object variant of the dSprites dataset [8]. This dataset consists of 64×64 images of sprite shapes with random colours, orientations and locations. Figure 2 shows training curves, model reconstructions and the development of KL-divergence attention maps over the course of training. For more experimental detail including model architectures see appendix A.

4.1 Between-object disentangling

In order to check whether the MVAE has learned a factored representation of objects we use the following qualitative tasks:

Entity exchange. One of the main motivations for learning disentangled representations of objects in a scene is that it facilitates compositional reasoning. We should be able to imagine an object in different contexts, independent of the original context in which we observed it. We examine our model's capacity to transfer objects into new contexts by encoding a pair of scenes, swapping the representations for one one object in each of the scenes, and then decoding both of the altered scenes. Figure 3 shows some example results for the MVAE. We note that entire objects are cleanly swapped between scenes, even in the presence of overlap or clutter, indicating that objects in the input image are cleanly partitioned into separate object representations.

Sequential decoding. Another qualitative indicator of the representations learned by the MVAE is to encode an input scene, then decode one object a time. As the MVAE's decoder performs object-wise aggregation, we can decode variable numbers of latent object representations. Figure 4a shows example decoded scenes in which we sequentially add one latent object representation at a time to the decoder. At each step a single object is introduced to the scene until all the objects present in the input scene have been decoded, and beyond this point the reconstructions are unaltered by the additional latent object representations. This indicates that the surplus latent object slots encode a 'Null' object, which decodes to nothing.

4.2 Within-object disentangling

To investigate the extent to which MVAE learns a representation that is disentangled within particular objects, we perform latent traversals for one object at a time, while keeping the other latent variables fixed. If the model has been successful we should expect to see that the underlying generative factors of the data are captured by single latent variables. Figure 4b shows a number of example latent traversals. The figure shows that the MVAE achieves a good degree of disentangling, e.g. with location factored from colour. This contrasts with the representations learned by a standard VAE (Appendix B), which are entangled across objects, with latent traversals causing multiple objects to deform and change colour simultaneously.

4.3 Unsupervised object counting

Here we demonstrate that the MVAEs spatial KL-divergence maps are a good proxy for object counting. We searched over KL-divergence thresholds on a training set of size 6400, and using the best training threshold tested on a 12800 newly sampled data examples. The chosen threshold gets 74.4% and 72.6% object count accuracy on the training and test sets respectively.

5 Discussion

Here we introduced a probabilistic model for learning object-based representations of visual scenes, which performs efficient inference using a novel one-shot informational attention mechanism that scales to large numbers of objects. Our results showed that our MVAE model can learn discrete, self-contained, interchangeable representations of multiple objects in a scene. We also found that the learned latent features of each object representation formed a disentangled code that separated the underlying factors of variation.

One limitation of this work is that the latent objects are assumed to be independent, which is obviously inconsistent with the tremendous statistical structure among objects in real scenes. Future work will tackle the task of learning not just about objects, but about their relations, interactions, and hierarchies in a unsupervised setting.

This work opens new directions for learning efficient and useful representations of complex scenes that may benefit question-answering and image captioning systems, as well as reinforcement learning agents. Moreover, this work may also help cognitive science explain why object-based representations are central to biological intelligence, as well as their evolutionary and experiential origins.

References

[1] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2172–80, 2016.

[2] J. J. DiCarlo and D. D. Cox. Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341, 2007.

[3] S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *NIPS*, pages 3225–3233, 2016.

[4] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. DRAW: A recurrent neural network for image generation. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1462–1471. JMLR.org, 2015.

[5] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2016.

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

[7] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. B. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, pages 2539–2547, 2015.

[8] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[9] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[10] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[11] L. Romaszko, C. K. I. Williams, P. Moreno, and P. Kohli. Vision-as-inverse-graphics: Obtaining a rich 3d explanation of a scene from a single image. In *ICCV Workshops*, pages 940–948. IEEE Computer Society, 2017.

[12] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. P. Lillicrap. A simple neural network module for relational reasoning. *NIPS*, abs/1706.01427, 2017.

[13] E. S. Spelke and K. D. Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007.

[14] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[16] N. Watters, A. Tacchetti, T. Weber, R. Pascanu, P. Battaglia, and D. Zoran. Visual interaction networks. *arXiv preprint arXiv:1706.01433*, 2017.

[17] J. Wu, J. B. Tenenbaum, and P. Kohli. Neural scene de-rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Appendix

A Experimental details

Data generation: Sprites are sampled independently, including whether or not they exist, up to some fixed upper bound on the number of sprites. We blend the colours of overlapping sprites by summing their RGB values and clipping at one. During training we generate these images dynamically, such that the MVAE always sees newly sampled data.

Encoder: The MVAE encoder is a convolutional network that starts with four convolutional layers with max-pooling. The resulting feature maps are concatenated with two channels of relative x,y co-ordinates followed by two further convolutional layers with a final channel dimensionality of 24. The output channels are split to form the means and log-variances of 12-dimensional diagonal Gaussian posteriors.

Decoder: Latent object representations are processed with a shared network $\mathbf{g}$. This network has two fully connected layers, the outputs of which are reshaped into a feature maps of shape $8 \times 8 \times 16$, followed by two convolutional layers with bilinear up-sampling. We aggregate across object representations using max-pooling, and then process the resulting feature map with a convolutional network $\mathbf{h}$. This network consists of three convolutional layers and one bilinear up-sampling layer. The network outputs a $64 \times 64 \times 3$ image of Bernoulli logits.

Standard VAE: We trained a baseline VAE with a comparable architecture to the MVAE. We match the latent dimensionality, using 60 latent dimensions. The encoder is identitical to the MVAE encoder, but the output feature maps are projected to the parameters of a 60-dimensional Gaussian using a linear layer. The decoder is equivalent to applying the shared object network $\mathbf{g}$ from the MVAE to the latent variables, and then passing the outputs into the rendering network $\mathbf{h}$.

Training: We train the MVAE using first-order gradient methods to maximize the variational lower bound on the log-probability of the data as in a standard VAE. All Models used elu non-linearities and were trained with Adam optimizer with scheduled learning rate annealing from 10^{-3} to 10^{-5} over the course of $3 * 10^6$ training steps.

B VAE latent traversals

As a comparison to the MVAE latent traversals in figure 4, we show traversals of latent dimensions for a trained VAE baseline in figure 5. Latent dimensions were hand-chosen as examples of variables that have a significant impact on the decoded results.

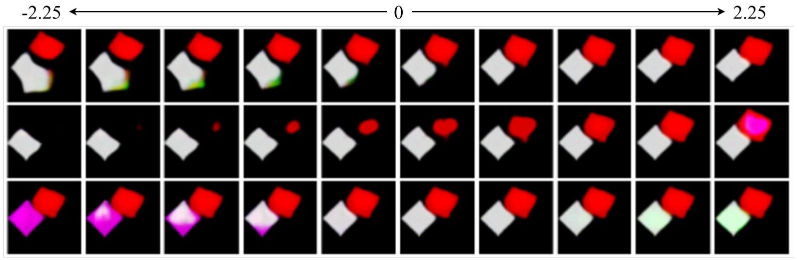


Figure 5: **VAE latent traversals.** Decoded scenes for traversals from -2.25 to 2.25 for a selection of latent dimensions for a standard VAE. All other latent variables are held fixed during the process.

C Reconstructions and samples

Figure 6 shows reconstructions for 3-sprite and 8-sprite datasets. We note that in the 3-sprite data that both the VAE and MVAE achieve good reconstructions, however the MVAE samples are of a higher quality, with more distinctly defined and coherent objects. The MVAE achieves good reconstructions and samples for the highly cluttered scenes in the 8-sprite data.

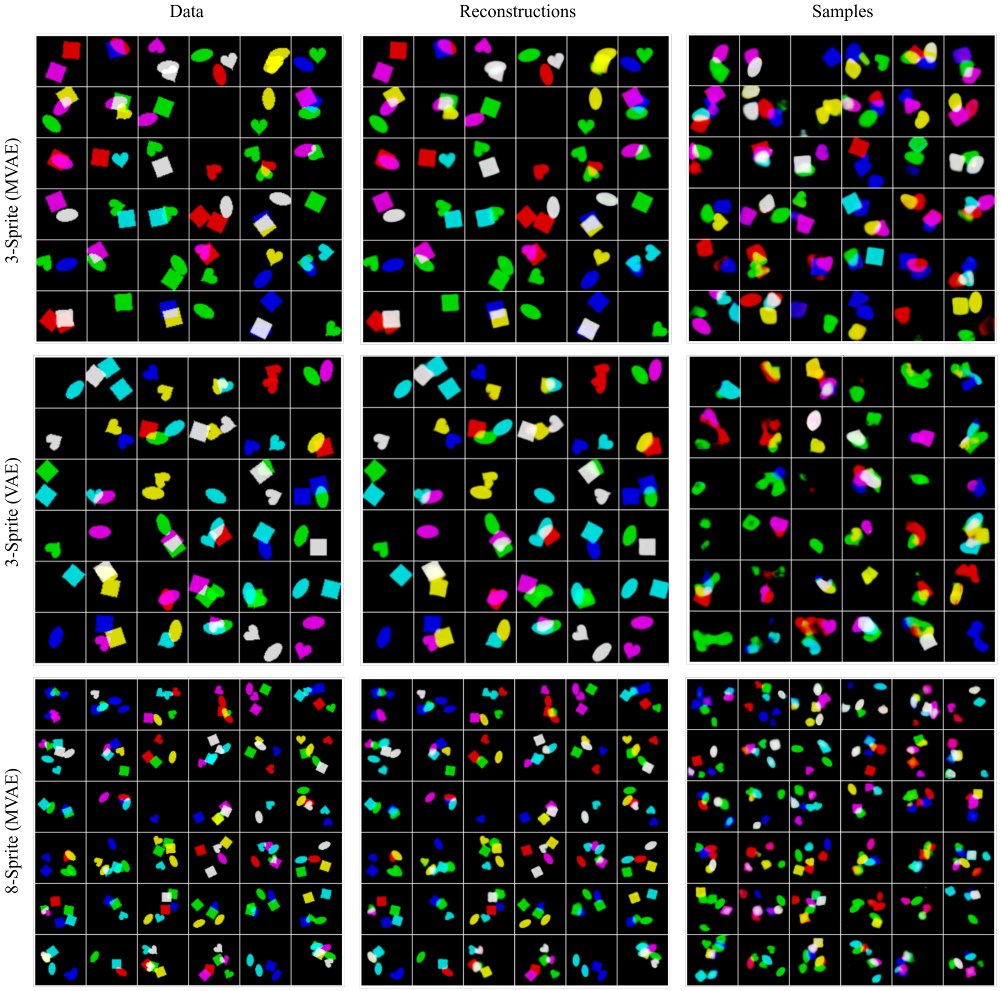


Figure 6: **Reconstructions and samples.** Data images, model reconstructions and model samples on the 3-sprite and 8-sprite datasets.