MONet: Unsupervised Scene Decomposition and Representation

Christopher P. Burgess, Loic Matthey, Nicholas Watters,
Rishabh Kabra, Irina Higgins, Matt Botvinick, Alexander Lerchner

DeepMind
London, United Kingdom

{cpburgess, lmatthey, nwatters,
rkabra, irinah, botvinick, lerchner}@google.com

Abstract

The ability to decompose scenes in terms of abstract building blocks is crucial for general intelligence. Where those basic building blocks share meaningful properties, interactions and other regularities across scenes, such decompositions can simplify reasoning and facilitate imagination of novel scenarios. In particular, representing perceptual observations in terms of entities should improve data efficiency and transfer performance on a wide range of tasks. Thus we need models capable of discovering useful decompositions of scenes by identifying units with such regularities and representing them in a common format. To address this problem, we have developed the Multi-Object Network (MONet). In this model, a VAE is trained end-to-end together with a recurrent attention network – in a purely unsupervised manner – to provide attention masks around, and reconstructions of, regions of images. We show that this model is capable of learning to decompose and represent challenging 3D scenes into semantically meaningful components, such as objects and background elements.

1 Introduction

Realistic visual scenes contain rich structure, which humans effortlessly exploit to reason effectively and intelligently. In particular, object perception, the ability to perceive and represent individual objects, is considered a fundamental cognitive ability that allows us to understand – and efficiently interact with – the world as perceived through our senses [Johnson, 2018, Green and Quilty-Dunn, 2017]. However, despite recent breakthroughs in computer vision fuelled by advances in deep learning, learning to represent realistic visual scenes in terms of objects remains an open challenge for artificial systems.

The impact and application of robust visual object decomposition would be far-reaching. Models such as graph-structured networks that rely on hand-crafted object representations have recently achieved remarkable results in a wide range of research areas, including reinforcement learning, physical modeling, and multi-agent control [Battaglia et al., 2018, Wang et al., 2018, Hamrick et al., 2017, Hoshen, 2017]. The prospect of acquiring visual object representations through unsupervised learning could be invaluable for extending the generality and applicability of such models.

Most current approaches to object decomposition involve supervision, namely explicitly labeled segmentations in the dataset [Ronneberger et al., 2015, Jégou et al., 2017, He et al., 2017]. This limits the generalization of these models and requires ground-truth segmentations, which are very difficult to acquire for most datasets. Furthermore, these methods typically only segment an image and don't learn structured object representations. While some unsupervised methods for scene decomposition have been developed, their performance is limited to very simple visual data [Greff et al., 2016, 2017, van Steenkiste et al., 2018, Eslami et al., 2016]. On the other hand, Generative Query Networks [Eslami et al., 2018] have demonstrated impressive modelling of rich 3D scenes, but do not explicitly factor representations into objects and are reliant on privileged view information as part of their training.

Recent progress has been made on learning object representations that support feature compositionality, for which a variety of VAE-based methods have become state-of-the-art [Higgins et al., 2017, Kim and Mnih, 2017, Chen et al., 2016, Locatello et al., 2018]. However, these methods ignore the structural element of objects, hence are limited to simple scenes with only one predominant object.

We propose that good representations of scenes with multiple objects should fulfill the following desiderata:

- A common representation space used for each object in a scene.
- Ability to accurately infer objects in 3-dimensional scenes with occlusion.
- Flexibility to represent visual datasets with a variable number of objects.
- Generalise at test time to (i) scenes with a novel number of objects (ii) objects with novel feature combinations, and (iii) novel co-occurrences of objects.

Here, we introduce an architecture that learns to segment and represent components of an image. This model includes a segmentation network and a variational autoencoder (VAE) [Kingma and Welling, 2014, Rezende et al., 2014] trained in tandem. It harnesses the efficiency gain of operating on an image in terms of its constituent objects to decompose visual scenes.

We call this model the **Multi-Object Network (*MONet*)** and apply it to a variety of datasets, showing that it satisfies all of our aforementioned desiderata. Our key contributions are:

1. An unsupervised generative model for visual scenes.
2. State-of-the-art decomposition performance on non-trivial 3D scenes, including generalisation and occlusion-handling.
3. Ability to learn disentangled representations of scene elements in a common latent code.

2 Method

2.1 The Multi-Object Network

The Multi-Object network (*MONet*) models scenes compositionally, by spatially decomposing a scene into parts and modelling each of those parts individually over a set of 'slots' with

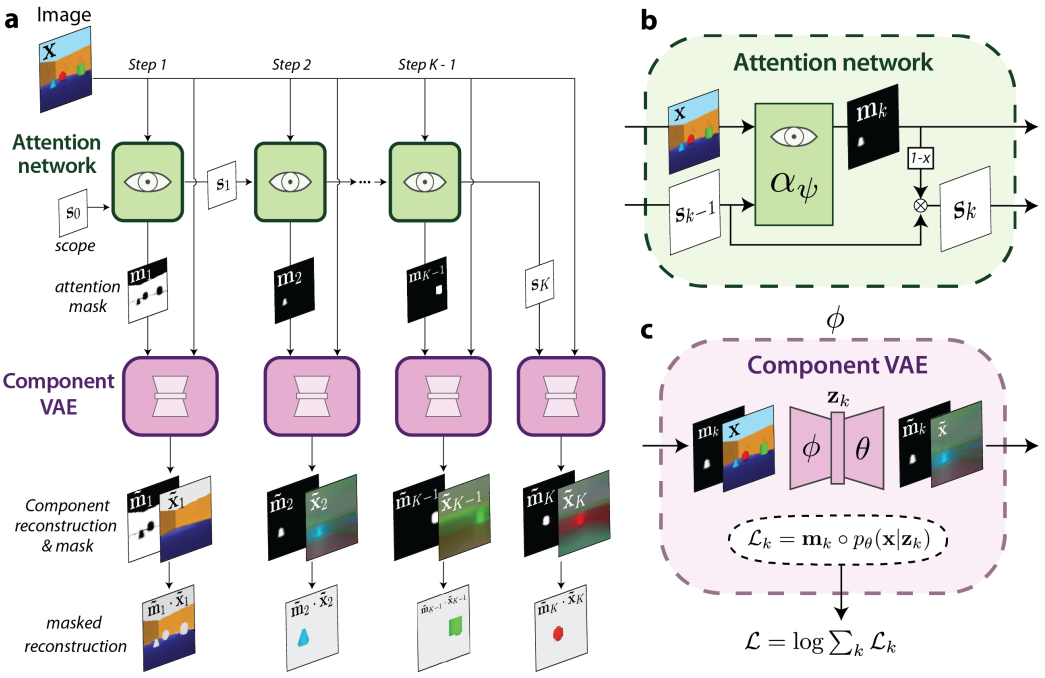

Figure 1: Schematic of *MONet*. **(a)** Overall compositional generative model architecture. The attention net recurrently generates the masks over a sequence of steps to condition the component VAE, labelling which pixels to focus on representing and reconstructing for that component. **(b)** Recursive decomposition process: the attention network at a particular step is conditioned on the image and the current *scope*, which is what currently remains to be explained of the scene (with the initial scope $\mathbf{s}_0 = \mathbf{1}$). The attention mask outputted will be some portion of the scope, and the scope for the next step is then what still remains to be explained after accounting for this attention mask. **(c)**. The component VAE receives both the image and a mask as input, and is pressured only to model the masked region by applying the mask to weight the component likelihood in the loss. Thus the reconstruction component is unconstrained outside of the masked region, enabling it for example to fill in occluded regions. The VAE also models the masks themselves. See main text for more details.

a common representation code (Figure 1). An attention module provides spatial masks corresponding to all the parts for a given scene, while a component VAE independently models each of the parts indicated by the masks.

The component VAE is a neural network, with an encoder parameterised by ϕ and a decoder parameterised by θ (see Figure 1c). The encoder parameterises a distribution over the component latents $\mathbf{z}_k$, conditioned on both the input image $\mathbf{x}$ and an attention mask $\mathbf{m}_k$. The mask indicates which regions of the image the VAE should focus on representing via its latent posterior distribution, $q_\phi(\mathbf{z}_k|\mathbf{x}, \mathbf{m}_k)$. Crucially, during training, the VAE's decoder likelihood term in the loss $p_\theta(\mathbf{x}|\mathbf{z}_k)$ is weighted according to the mask, such that it is unconstrained outside of the masked regions. A complete image is compositionally modelled by conditioning with a complete set of attention masks for the image (i.e. $\sum_{k=1}^{K} \mathbf{m}_k = \mathbf{1}$), over K independent passes through the VAE.

The VAE is additionally required to model the attention masks over the K components,

where their distribution $p(\mathbf{c}|\{\mathbf{m}_k\})$ is the probability that pixels belong to a particular component k, i.e. $\mathbf{m}_k = p(\mathbf{c} = k|\{\mathbf{m}_k\})$. In *MONet*, the mask distribution is learned by the attention module, a neural network conditioned on $\mathbf{x}$ and parameterised by ψ. Thus, we refer to this distribution as $q_\psi(\mathbf{c}|\mathbf{x})$. The VAE's generative model of those masks is denoted as $p_\theta(\mathbf{c}|\{\mathbf{z}_k\})$.

We wanted *MONet* to be able to model scenes over a variable number of slots, so we used a recurrent attention network α_ψ for the decomposition process. In particular, we arranged the recurrence as an autoregressive process, with an ongoing state that tracks which parts of the image have yet to be explained (see Figure 1b). We call this state the scope $\mathbf{s}_k$, which is an additional spatial mask updated after each attention step. Specifically, it signifies the proportion of each pixel that remains to be explained given all previous attention masks, where the scope for the next step is given by:

$$\mathbf{s}_{k+1} = \mathbf{s}_k\big(1 - \alpha_\psi(\mathbf{x}; \mathbf{s}_k)\big) \tag{1}$$

with the first scope $s_0 = \mathbf{1}$. The attention mask for step k is given by:

$$\mathbf{m}_k = \mathbf{s}_{k-1}\alpha_\psi(\mathbf{x}; \mathbf{s}_{k-1}) \tag{2}$$

except for the last step K, where the attention network is not applied, but the last scope is used directly instead, i.e. $\mathbf{m}_K = \mathbf{s}_{K-1}$. This ensures that the entire image is explained, i.e. $\sum_{k=1}^{K} \mathbf{m}_k = \mathbf{1}$.

The whole system is trained end-to-end with a loss given by:

$$\begin{aligned}\mathcal{L}(\phi; \theta; \psi; \mathbf{x}) = & -\log \sum_{k=1}^{K} \mathbf{m}_k p_\theta(\mathbf{x}|\mathbf{z}_k) + \beta D_{KL}\big(\prod_{k=1}^{K} q_\phi(\mathbf{z}_k|\mathbf{x}, \mathbf{m}_k) \parallel p(\mathbf{z})\big) \\ & + \gamma D_{KL}\big(q_\psi(\mathbf{c}|\mathbf{x}) \parallel p_\theta(\mathbf{c}|\{\mathbf{z}_k\})\big)\end{aligned} \tag{3}$$

The first two terms of the loss are derived from the standard VAE loss. The first term is the decoder negative log likelihood, given our mixture of components decoder distribution, as discussed above. The second term is the Kullback–Leibler divergence (KL) divergence of the latent posterior (factorised across slots) with the latent prior, weighted with a hyperparameter β, following Higgins et al. [2017], which can be tuned to encourage learning of disentangled latent representations. The last term to be minimised is the KL divergence between the attention mask distribution $q_\psi(\mathbf{c}|\mathbf{x})$ and the VAE's decoded mask distribution $p_\theta(\mathbf{c}|\{\mathbf{z}_k\})$. This is also weighted by a tuneable hyperparameter γ that here modulates how closely the VAE must model the attention mask distribution.

2.2 Exploiting compositional structure

In this section we aim to motivate the development of the approach described above, specifically exploring the reasons we might expect the loss defined in Eq. 3 to decrease if masks corresponding to semantically meaningful decompositions are learned. This includes an empirical confirmation, for which we construct a visually interesting 3D scenes dataset (also used in our main results) and compare performance under different decomposition regimes. However, the main results for unsupervised scene decomposition with *MONet* are in the next section (Section. 3).

We originally developed this architecture while considering how to learn to meaningfully decompose a scene without supervision. We wanted to identify some general consequences of

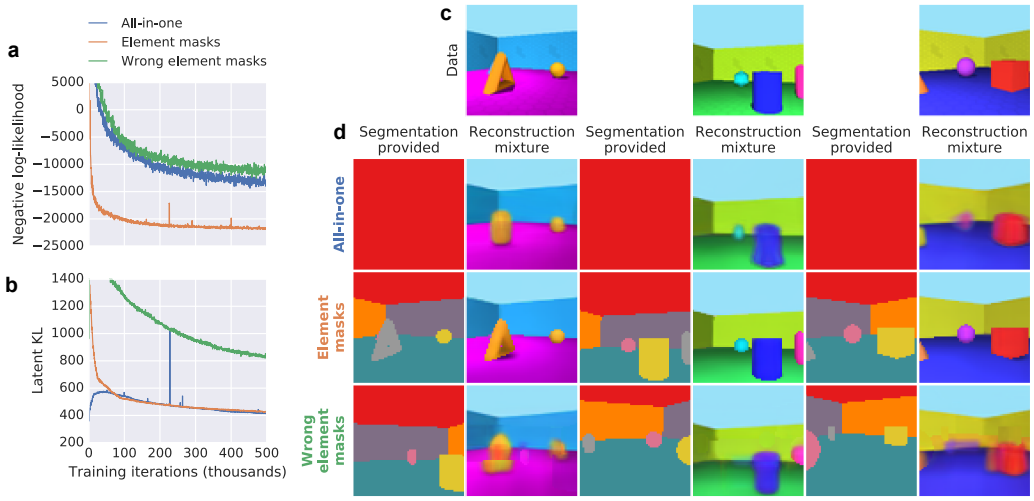

Figure 2: Semantic decomposition improves reconstruction accuracy. These are results from experiments to motivate the *MONet* training objective by replacing the learned masks from the attention network with provided masks (see Section. 2.2). The component VAE is trained to reconstruct regions of *Objects Room* images given the provided masks. Three mask conditions are compared: reconstructing everything in the first pass (*all-in-one*), reconstructing individual scene elements in separate passes using ground-truth masks (*element masks*), and finally a control condition requiring reconstruction of regions given by element masks for the wrong scene (*wrong element masks*). **(a)** and **(b)** show the negative log-likelihood (mask-weighted sum over scene elements) and KL divergence (with the latent prior), respectively over training for the three mask conditions. **(d)** Example masks for each of the conditions (summarised as colour-coded segmentation maps), for example scenes shown in **c**, with corresponding reconstructions after training under the different conditions (reconstructions rendered by mixing the scene reconstruction components according to the masks). The colors used in the segmentation visualization are independent of the colors in the scenes themselves.

the compositional structure of scenes that could push the optimisation towards decomposition. We started from the hypothesis that compositional visual scenes can be more efficiently processed by something like a deep neural network if there is some common repeating structure in scenes that can be exploited. In particular, if a network performing some task can be repeatedly reused across scene elements with common structure (such as objects and other visual entities), its available capacity (limited for example by its architecture and weights) will be more effectively utilised and thus will be more efficient than the same network processing the entire scene at once. This both motivates a key benefit of decomposition – identifying ways to break up the world that can make a range of tasks more efficient to solve – *and* suggests an approach for identifying the compositional structure of data.

In the context of our scene representation learning goal, this hypothesis would predict that a network tasked with autoencoding visual scenes would perform better, *if* it is able to build up scenes compositionally by operating at the level of the structurally similar scene elements. Specifically, such a network should have a lower reconstruction error than if the same network were instead trained to reconstruct entire scenes in a single pass.

Empirical validation. To test this hypothesis, we constructed a dataset of rendered 3D scene images of non-trivial visual complexity, the *Objects Room* dataset (see Figure 10 for example images). We specifically designed the scenes to be composed of multiple scene elements that share varying degrees of common structure. The *Objects Room* images were generated as randomised views inside a cubic room with three randomly shaped, coloured and sized objects scattered around the room. For any given image, 1-3 of those objects are in view. The wall and floor colour of the room in each image are also randomised (see Section. C in the appendix for more details). For each image $\mathbf{x}$, a set of 7 ground-truth spatial masks $\hat{\mathbf{m}}$ was also generated, indicating the pixel extents of visual elements comprising the scene (the floor, sky, each of two adjoining wall segments, and three objects).

We tested the hypothesis by training the model outlined above on this dataset, but instead of learning the masks as in *MONet* we provided a set of seven attention masks $\{\mathbf{m}_k\}$ masks under three different conditions of interest. In the *all-in-one* condition, the entire image was always segmented into the first mask (i.e. $\mathbf{m}_1 = \mathbf{1}$ with the remaining masks being all zeros). In the *element masks* condition, the scene was segmented according to ground-truth visual element masks, i.e. $\{\mathbf{m}_k\} = \{\hat{\mathbf{m}}_k\}$. Finally, the *wrong element masks* is a control condition with the mask in the same format as *element masks*, but always for a different, random scene (i.e. always an incorrect segmentation, but with the same statistics as *element masks*).

Figure 2 shows the results of training the component VAE under each condition. As predicted, the reconstructions for the model trained with *element masks* are significantly better than the reconstructions from the model trained with *all-in-one* masks. This can be seen by the large gap between their reconstruction errors over the course of training (see the negative log-likelihoods in Figure 2a). The *element masks* reconstructed images are also markedly better visually (see the mixtures of the reconstruction components in Figure 2d), with objects from *all-in-one* in particular being more blurred, and sometimes missing (e.g. in the middle example in Figure 2d, the *all-in-one* model fails to reconstruct the magenta object). In contrast, the latent KLs for the two conditions were very similar (see Figure 2b).

However, the kind of segmentation is important: the model trained with *wrong element masks* (where the masks do not correspond to structurally meaningful visual elements for the relevant scene) performs worse than both of the other conditions. Reconstruction error and latent KL are significantly higher (gold curves in Figure 2a and Figure 2b), and the reconstructed images have clear edge artifacts where the provided masks are misaligned with the structural elements in the scene (bottom row in Figure 2d).

Overall, these results support the hypothesis that processing elements of scenes in a way that can exploit any common structure of the data makes more efficient use of a neural network's capacity. Furthermore, the significant improvement seen in the loss suggests this corollary can be used to discover such structurally aligned decompositions without supervision. In the next section, we show the results from the full *MONet* setup, in which the attention masks are learned in tandem with the object representations, in a fully unsupervised manner.

3 Results

For all experiments we used the same basic architecture for *MONet*. The attention network used an architecture inspired by the U-Net [Ronneberger et al., 2015]. For the VAE, we used an encoder with convolutional followed by fully connected layers to parameterise a diagonal

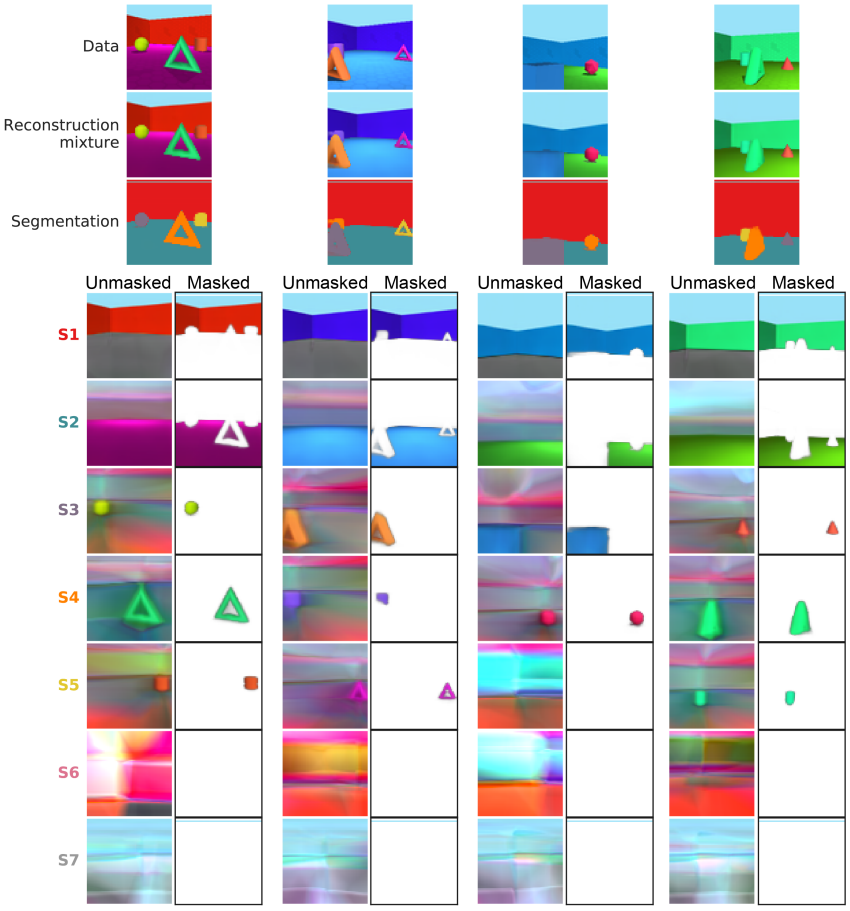

Figure 3: Unsupervised decomposition results on the *Objects Room* dataset. Results from *MONet* trained on a dataset of 3D scenes of a room with 1-3 objects in view. Each example shows the image fed as input data to the model, with corresponding outputs from the model. Reconstruction mixtures show sum of components from all slots, weighted by the learned masks from the attention network. Colour-coded segmentation maps summarising the attention masks $\{\mathbf{m}_k\}$ are shown in the next row. Rows labelled S1-7 show the reconstruction components of each slot. Unmasked versions are shown side-by-side with corresponding versions that are masked with the VAE's reconstructed masks $\tilde{\mathbf{m}}_k$. In the third example, note the correct handling of the large blue cube with the same colour as the background.

Gaussian latent posterior (with a unit Gaussian prior), together with a spatial broadcast decoder [Watters et al., 2019] to encourage the VAE to learn disentangled features. The decoder parameterised the means of pixel-wise independent Gaussian distributions, with fixed scales. All results are shown after training for 1,000,000 iterations. See Section. B in the appendix for more details on the architecture and hyperparameters used.

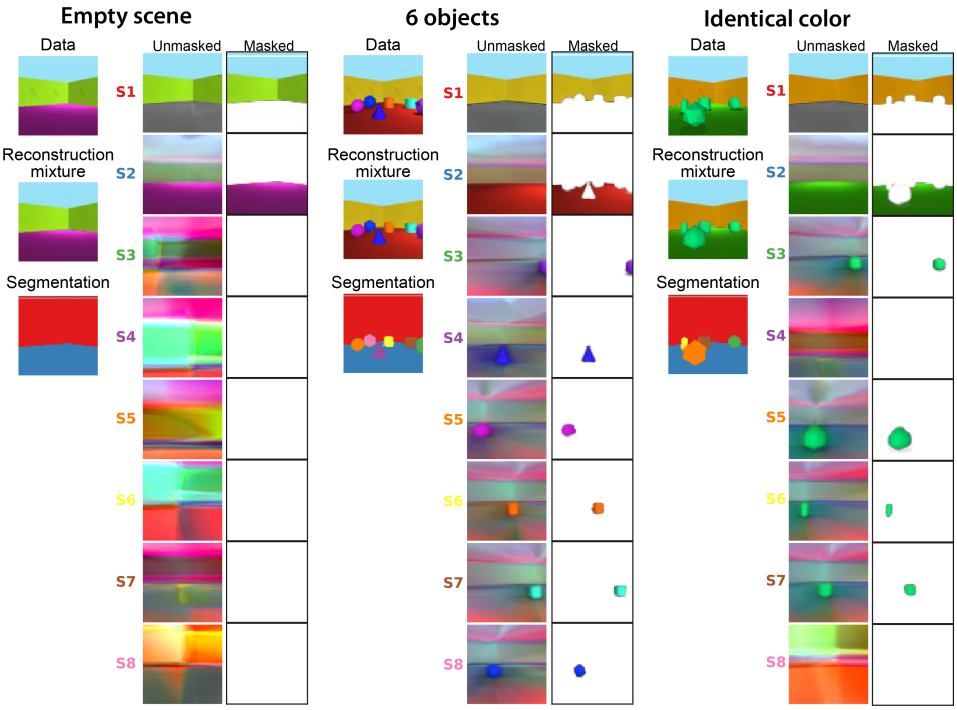

Figure 4: *MONet* generalisation outside of training regime and data distribution. Results presented in a similar format and using the same trained model as as Figure 3 but now run for 9 slots (of which only slots 1-8 are shown). The model robustly segments novel scenes with no objects (left), or with 6 objects (middle) – double the number of objects seen at training time (utilising extra test time slots). It also correctly handles scenes with 4 identically coloured and visually overlapping objects (right).

3.1 Results on *Objects Room*

We trained *MONet* with $K = 7$ slots on the *Objects Room* dataset with 1-3 objects in view (introduced in Section. 2.2); results are shown in Figure 3. *MONet* learns to generate distinct attention masks for the floor, wall + sky (combined), and the individual objects (including some of their shadows), across a variety of scenes as can be seen in the segmentation maps (*Segmentation* row in Figure 3a).

The unmasked reconstruction components of each slot are shown side-by-side with masked versions (see rows labelled S1-S7 Figure 3 show the respective unmasked reconstruction components). The latter are masked using the VAEs mask reconstructions $\tilde{\mathbf{m}}_k$, and show accurate reconstructions within those regions as well as demonstrating the VAEs modelling of the attention mask distribution. The unmasked components reveal coherent filling in over regions otherwise occluded by objects (e.g. floors and walls). Where slots have empty masks, their reconstructions components tend to contain unspecific background patterns. Combining reconstruction components according to their attention masks yields high quality reconstructions of the full scenes (*Reconstruction mixtures* in Figure 3).

We also assessed the generalisation capabilities of *MONet*, by testing with extra decomposition steps on novel scenes (Figure 4). Using the recurrent autoregressive process of

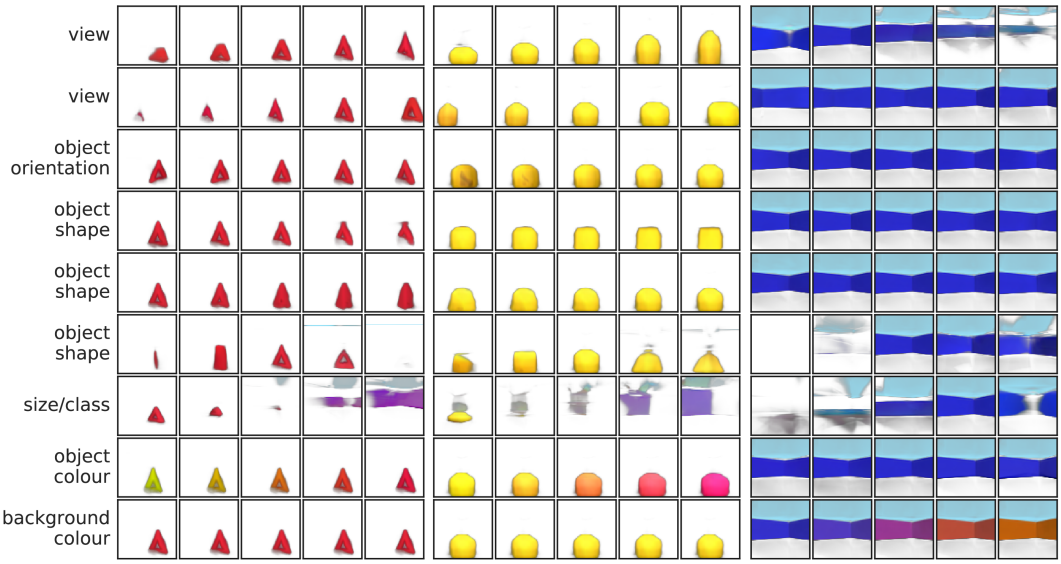

Figure 5: *MONet* object representations learned on *Objects Room*. Each plot (left, middle, and right) shows how the reconstruction component of a particular slot (with its reconstructed mask applied) changes as each single latent $z_{k,i}$ is traversed from -1.0 to +1.0. Same trained model as in Figure 3. Results from different seed images and slots shown in each column group. The left and middle column groups show components from slots representing scene objects, and the right column shows a wall + sky component. A subset of latents were picked based on visual inspection and are consistent across column groups. Labels show their intuitive feature interpretations. Note here we used reconstructed masks that were not normalised across other components to generate the components in isolation.

decomposition, we were able to run the same trained model but for more attention steps at test time, here extending it to a 9 slot model (although only the first 8 are shown, with more extensive results shown in supplementary Figure 8). This model generalised well to scene configurations not observed during training. We presented empty room scenes, scenes with 6 objects, and scenes with 4 objects of the same colour on a similarly coloured floor. In each case, the model produced accurate segmentations and filled up extra test time slots with the excess objects as appropriate.

3.2 Disentangled representations

An important design feature of *MONet* is that it learns a latent representation unified across slots. As discussed above, we utilised a weight modulating the latent KL in the loss [Higgins et al., 2017, Burgess et al., 2018] and a broadcast decoder [Watters et al., 2019] in *MONet* to encourage disentangling of the latent representation at the feature level.

We assessed the feature-level disentangling of the latent representations $\mathbf{z}_k$ by seeding input images and inspecting how the reconstruction components changed as single latents $\mathbf{x}_{k,i}$ were traversed (Figure 5). These results are generated from the same trained model as above. We show the components with their reconstructed masks applied to incorporate how the latent traversals also affect them (note in this figure we use masks not normalised

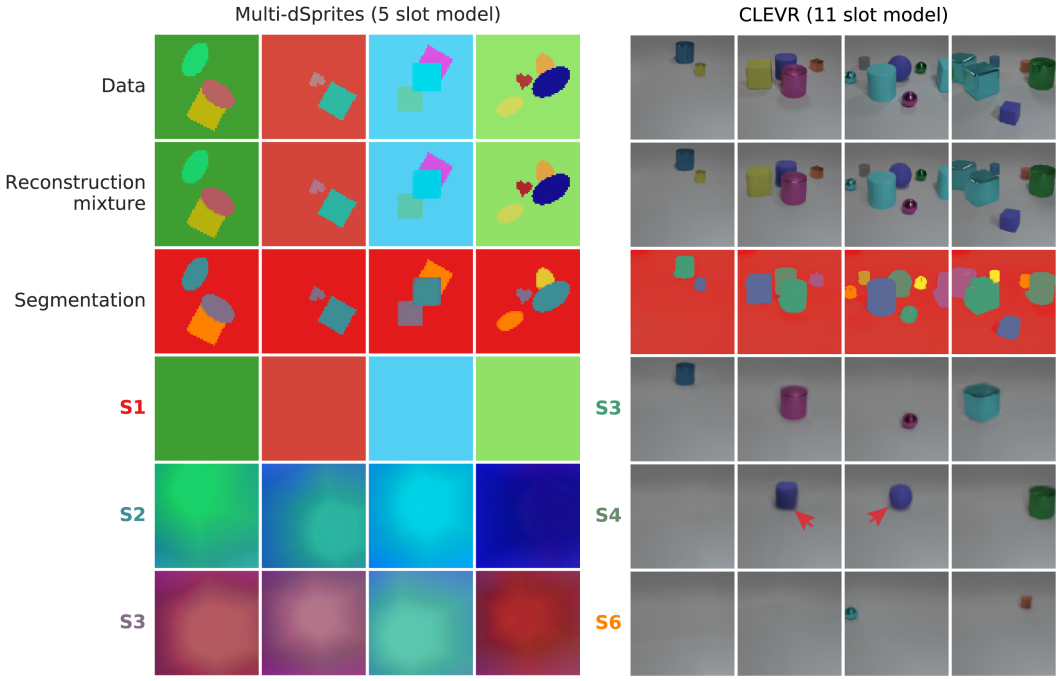

Figure 6: *MONet* decomposition on *Multi-dSprites* and *CLEVR* images. Format as in Figure 3. **Left panel** shows results from model with five slots trained on *Multi-dSprites* (with 1-4 sprites per image). Unmasked component reconstructions from first three slots are shown. Note the correct segmentation. **Right panel** shows results from 11 slot model trained on images from the *CLEVR* dataset [Johnson et al., 2017]. Unmasked component reconstructions from three representative slots are shown. Red arrows highlight occluded regions of shapes that are completed as full shapes. Rightmost column demonstrates accurate delineation of two very similar visually overlapping shapes.

by other components). From visual inspection, we identified many latents independently controlling interpretable features of each visual element, indicative of disentangling. Some latents only controlled specific features in slots containing scene objects (those prefixed with 'object', e.g. the shape latents), or were specific to slots containing the wall + sky component (e.g. the distinct 'background' and 'object' colour latents). Others controlled features across all visual element classes (e.g. the 'view' latents), or switched between them (the 'size/class' labelled latent).

3.3 Results on *Multi-dSprites*

MONet was also able to accurately decompose 2D sprite scenes into the constituent sprites (see the left panel Figure 6). To create this dataset, we adapted the dSprites [Matthey et al., 2017] dataset by colourising 1-4 randomly chosen dSprites and compositing onto a single image with a uniform randomly coloured background (with occlusions; see Section. C in the appendix for more details). The same model architecture and hyperparameters as before were used (except with $K = 5$ slots), enabling us to verify robustness across very

different datasets. After training, the model was able to distinguish individual sprites and the background robustly, even those sprites behind multiple occlusions or very difficult to distinguish by eye. As individual unmasked reconstruction components generally consisted of the entire image taking the relevant sprite's colour, we do not show the masked versions for brevity. More examples can be seen in supplementary Figure 7

3.4 Results on *CLEVR*

We tested *MONet* on images from *CLEVR* [Johnson et al., 2017], a dataset of simple 3D rendered objects (see right panel in Figure 6). We down-sampled the images to 128x128, and cropped them slightly to ensure more of the frame was occupied by objects (see Section. C in the appendix for more details), resulting in between 2-10 visible objects per image. To handle the increased resolution of this dataset, we added extra blocks to the attention network (see Section. B in the appendix for more details) and used $K = 11$ slots, but left all other hyperparameters unchanged.

The model robustly segmented scenes across the diversity of configurations into single objects and generated high quality reconstructions (right panel in Figure 6; and also see more extensive examples in supplementary Figure 9).

The model is also robust to the relatively frequent occlusions between objects in this dataset. Interestingly, the unmasked reconstruction components of occluded objects showed very coherent filling in of occluded regions, demonstrating how *MONet* is learning from and constrained by the structure of the data (red arrows in Figure 6 highlights some examples). The model was also able to correctly distinguish overlapping shapes with very similar appearances (see rightmost column in Figure 6).

4 Related Work

4.1 Supervised Approaches

Semantic segmentation using supervision from pixel-wise class labels is a well-studied problem. Neural architectures such as the U-Net [Ronneberger et al., 2015], Fully Convolutional DenseNet [Jégou et al., 2017], and DeepLabv3 [Chen et al., 2018]) have successively advanced the state of the art on this domain, furnishing strong inductive biases for image segmentation. *MONet* can utilize these architectures in place of its attention module (to propose viable object masks) even without a supervised training signal.

While *MONet* is partly about instance segmentation, its recurrent decomposition process bears little similarity to state-of-the-art instance segmentation models like Mask R-CNN [He et al., 2017] and PANet [Liu et al., 2018]. These are trained via supervision to propose bounding boxes for all objects in a scene at once. Separate modules then process the bounding boxes one at a time to yield object masks and classify the objects. The overarching distinction with our approach is that features learnt by instance segmentation methods only help identify object classes, not explain away the complexity of the scene. *MONet* also benefits from learning segmentations and representations jointly, which allows it to discover fine-grained object masks directly and without supervision. Admittedly, Mask R-CNN and PANet can work with larger, natural images with $\sim$100 objects.

4.2 Unsupervised Approaches

A principled approach to deconstruct scenes into underlying components is encapsulated by the vision-as-inverse-graphics paradigm. Methods in this camp [Tian et al., 2019, Yao et al., 2018] make domain-specific assumptions about the latent code [Kulkarni et al., 2015] or generative process [Wu et al., 2017] to keep inference tractable, but these assumptions tend to be too strong for the methods to be broadly useful. *MONet* demonstrates more general scene understanding by learning object features in an unstructured Gaussian latent space via end-to-end learning.

Methods apart from probabilistic and variational inference have also made some headway toward scene understanding. Adversarially trained generative models can demonstrate an implicit understanding of missing parts of a scene by inpainting over them [Pathak et al., 2016]. "Self-supervised" models can also show impressive results on object discovery [Doersch et al., 2015] or tracking [Vondrick et al., 2018]). But rather than learning from the structure and statistics of the data, these methods often rely on heuristics (such as region masks, neighboring patches, or reference frames for colorization), and occasionally on explicit supervision (e.g. to recognize the presence of a type of object in Shetty et al. [2018]). They largely abstract away the problem of representation learning and have not yielded general principles for representing objects.

That leaves us with a small class of models to which *MONet* is immediately comparable. The Attend-Infer-Repeat framework [Eslami et al., 2016] is the closest to our work in spirit. Like *MONet*, AIR highlights the representational power of object-wise inference, and uses a recurrent network to attend to one object at a time. Unlike *MONet*, it explicitly factors an object representation into 'what', 'where', and 'presence' variables. The 'what' is modelled by a standard VAE. A spatial transformer module [Jaderberg et al., 2015] then scales/shifts the generated component to match the 'where'. Finally, the components are added together (if 'present') to form the final image. This additive interaction is restrictive, and as a consequence, neither AIR nor its successor SQAIR [Kosiorek et al., 2018] can model occluded objects or background pixels. These models have not been shown to scale to larger number of objects.

Another line of work spanning Tagger [Greff et al., 2016], Neural Expectation Maximization [Greff et al., 2017], and Relational Neural Expectation Maximization [van Steenkiste et al., 2018] makes use of iterative refinement to decompose scenes into groups of pixels and assign them to individual components. These works draw their intuitive appeal from classical clustering and EM algorithms. But they have not been shown to work beyond small, binarized images or videos. We anticipate *MONet* can be readily extended to incorporate some form of iterative refinement across components in future work.

5 Conclusions and future work

We presented *MONet*, a compositional generative model for unsupervised scene decomposition and representation learning. Our model can learn to decompose a scene without supervision, and learns to represent individual components in a common disentangled code. The approach we take is motivated by a gain in efficiency from autoencoding scenes compositionally, where they consist of simpler coherent parts with some common structure. To the best of our knowledge, *MONet* is the first deep generative model that can perform meaningful unsupervised decomposition on non-trivial 3D scenes with a varying number of objects such as CLEVR.

We found our model can learn to generate masks for the parts of a scene that a VAE prefers to reconstruct in distinct components, and that these parts correspond to semantically meaningful parts of scene images (such as walls, floors and individual objects). Furthermore, *MONet* learned disentangled representations, with latents corresponding to distinct interpretable features, of the scene elements. Interestingly, *MONet* also learned to complete partially occluded scene elements in the VAE's reconstructions, in a purely unsupervised fashion, as this provided a more parsimonious representation of the data. This confirms that our approach is able to distill the structure inherent in the dataset it is trained on. Furthermore, we found that *MONet* trained on a dataset with 1-3 objects could readily be used with fewer or more objects, or novel scene configurations, and still behave accurately.

This robustness to the complexity of a scene, at least in the number of objects it can contain, is very promising when thinking about leveraging the representations for downstream tasks and reinforcement learning agents.

Although we believe this work provides an exciting step for unsupervised scene decomposition, there is still much work to be done. For example, we have not tackled datasets with increased visual complexity, such as natural images or large images with many objects. As the entire scene has to be 'explained' and decomposed by *MONet*, such complexity may pose a challenge and warrant further model development. It would be interesting, in that vein, to support *partial* decomposition of a scene, e.g. only represent some but not all objects, or perhaps with different precision of reconstruction.

We also have not explored decomposition of videos, though we expect temporal continuity to amplify the benefits of decomposition in principle, which may make *MONet* more robust on video data. Finally, although we showed promising results for the disentangling properties of our representations in Figure 5, more work should be done to fully assess it, especially in the context of reusing latent dimensions between semantically different components.

Acknowledgments

We thank Ari Morcos, Daniel Zoran and Luis Piloto for helpful discussions and insights.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan

Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint*, June 2018.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv*, 2018.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv*, 2016.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3225–3233. Curran Associates, Inc., 2016.

S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. ISSN 0036-8075. doi: 10.1126/science.aar6170. URL http://science.sciencemag.org/content/360/6394/1204.

E J Green and Jake Quilty-Dunn. What is an object file? *Br. J. Philos. Sci.*, December 2017.

Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hao, Harri Valpola, and Jürgen Schmidhuber. Tagger: Deep unsupervised perceptual grouping. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4484–4492. Curran Associates, Inc., 2016.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. *arXiv*, 2017.

J. Hamrick, A. Ballard, R. Pascanu, O. Vinyals, N. Heess, and P. Battaglia. Metacontrol for adaptive imagination-based optimization. In *ICLR*, 2017.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

Yedid Hoshen. Vain: Attentional multi-agent predictive modeling. In *Advances in neural information processing systems*, pages 2698–2708, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1175–1183. IEEE, 2017.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Scott P Johnson. Object perception. In *Oxford Research Encyclopedia of Psychology*. Oxford University Press, July 2018.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv*, 2017.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

Adam R Kosiorek, Hyunjik Kim, Ingmar Posner, and Yee Whye Teh. Sequential attend, infer, repeat: Generative modelling of moving objects. *arXiv preprint arXiv:1806.01794*, 2018.

Tejas D Kulkarni, William F. Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2539–2547. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5851-deep-convolutional-inverse-graphics-network.pdf.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset, 2017. URL https://github.com/deepmind/dsprites-dataset/.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, 32(2):1278–1286, 2014.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Rakshith R Shetty, Mario Fritz, and Bernt Schiele. Adversarial scene editing: Automatic object removal from weak supervision. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7717–7727. Curran Associates, Inc., 2018.

Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. Learning to infer and execute 3d shape programs. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rylNH20qFQ.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. URL http://arxiv.org/abs/1607.08022.

Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. *arXiv preprint arXiv:1802.10353*, 2018.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *The European Conference on Computer Vision (ECCV)*, September 2018.

T. Wang, R. Liao, J. Ba, and S. Fidler. Nervenet: Learning structured policy with graph neural networks. In *ICLR*, 2018.

Nicholas Watters, Loic Matthey, Christopher P. Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture forlearning disentangled representations in vaes. *arxiv*, 2019.

Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. Neural scene de-rendering. In *Proc. CVPR*, volume 2, 2017.

Shunyu Yao, Tzu Ming Hsu, Jun-Yan Zhu, Jiajun Wu, Antonio Torralba, Bill Freeman, and Josh Tenenbaum. 3d-aware scene manipulation via inverse graphics. In *Advances in Neural Information Processing Systems*, pages 1891–1902, 2018.

Supplementary material

A Supplementary figures

See Figures 7–9 for additional examples of *MONet* decomposition results on the different datasets we considered.

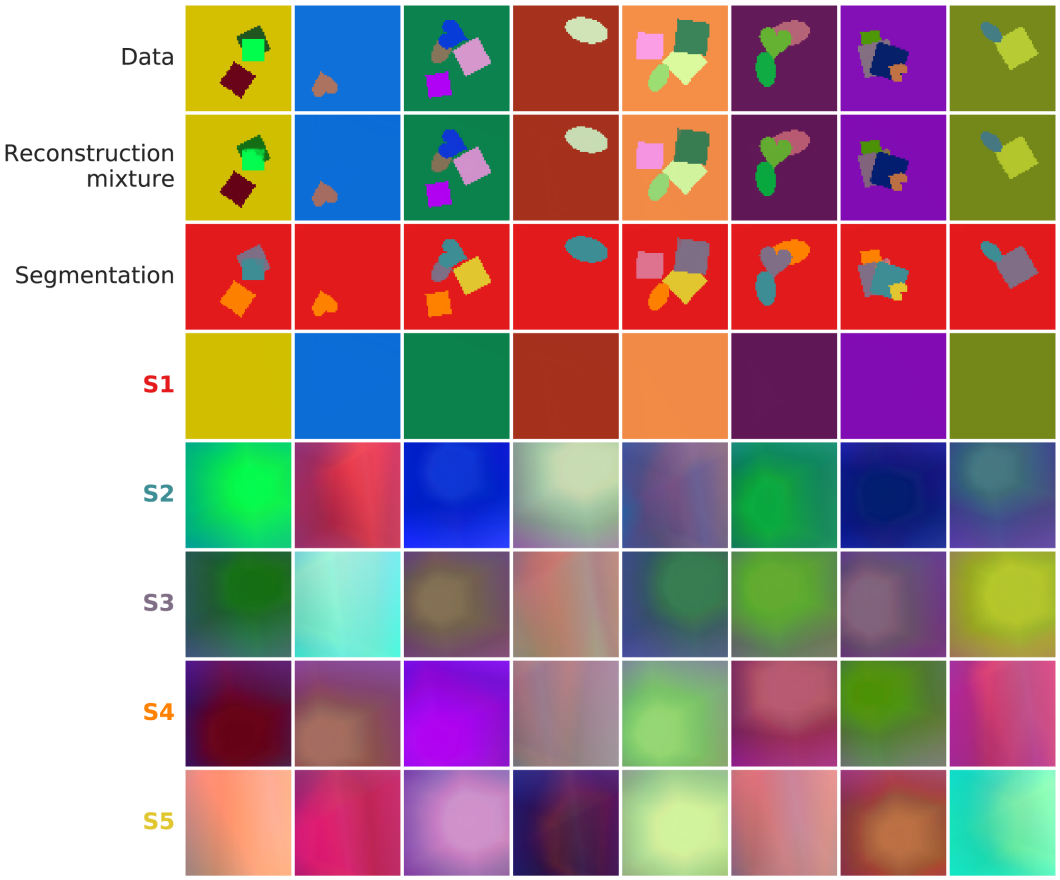

Figure 7: Unsupervised *Multi-dSprites* decomposition with *MONet*. Same trained model and format as Figure 6 but showing all 5 slots on randomly selected data samples.

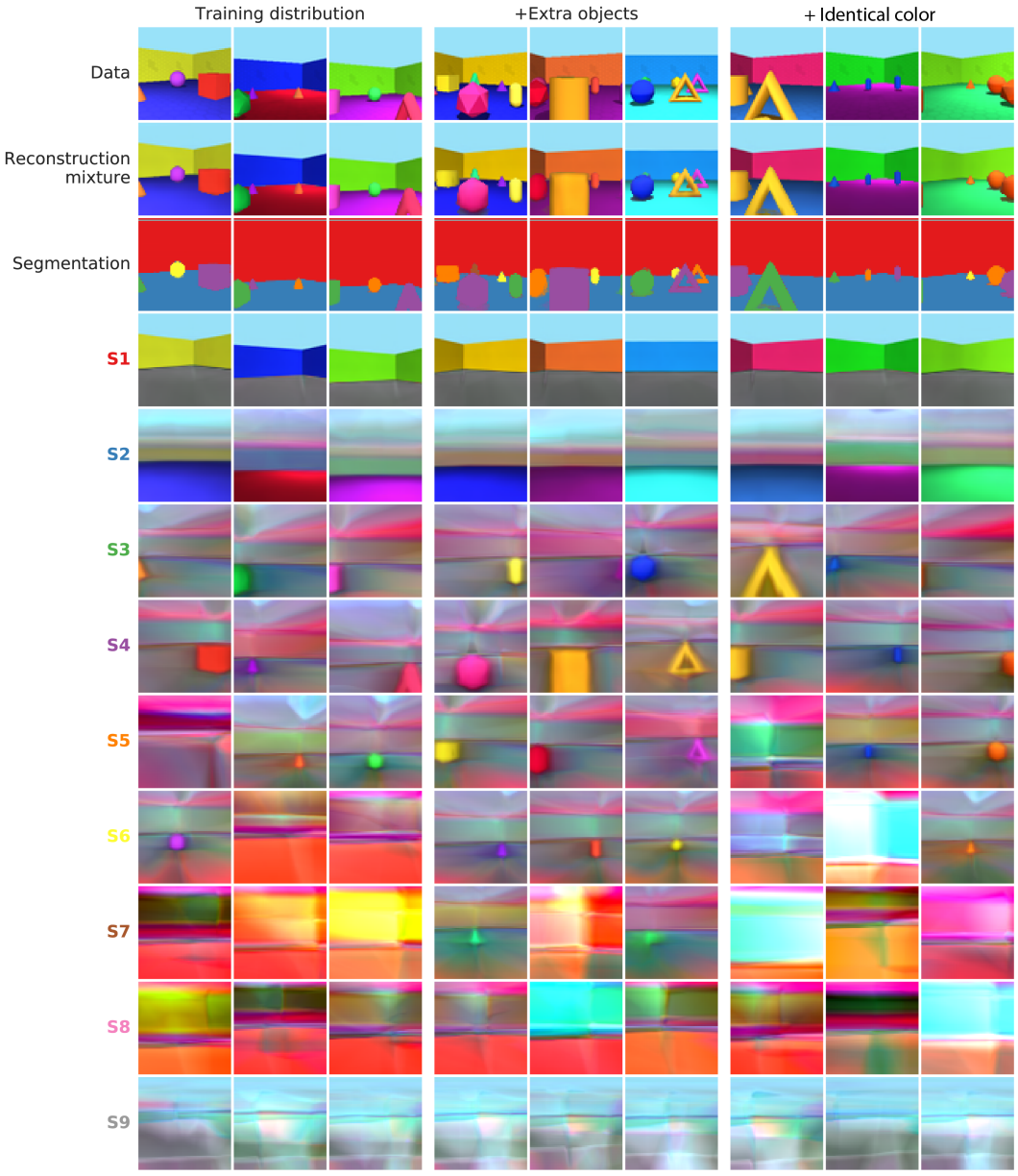

Figure 8: Unsupervised *Objects Room* decomposition with *MONet*. Same trained model and format as Figure 3 but showing all 9 slots (model trained on 7 slots) on randomly selected data samples. Left three columns show results on the training distribution. Remaining columns shown generalisation results outside of the training distribution: middle three columns on scenes with extra objects; last three columns on scenes with extra objects and all objects the same colour.

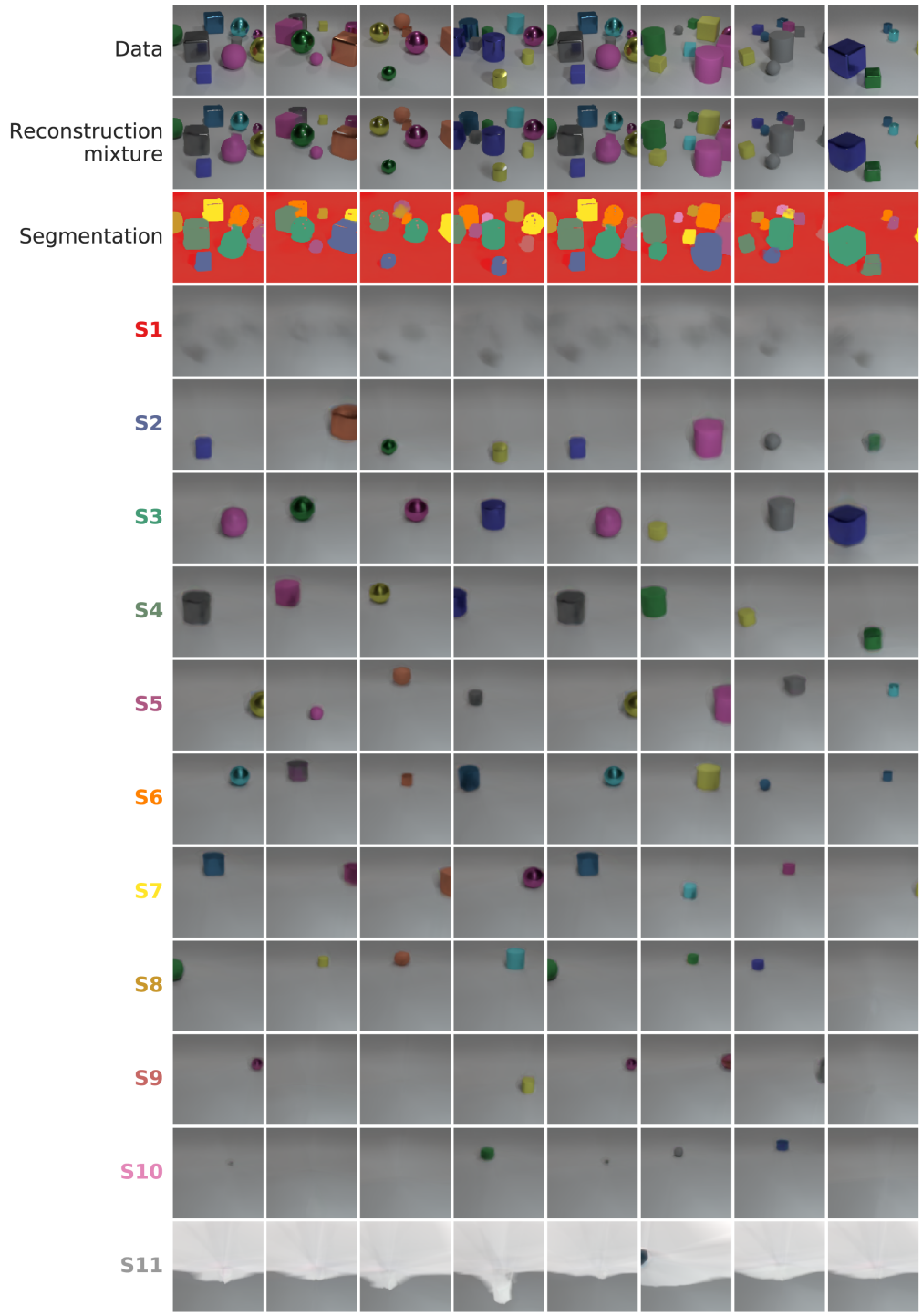

Figure 9: Unsupervised *CLEVR* decomposition with *MONet*. Same trained model and format as Figure 6 but showing all 11 slots on randomly selected data samples.

B Architecture and hyperparameter details

B.1 Component VAE

The VAE encoder is a standard CNN with 3x3 kernels, stride 2, and ReLU activations. It receives the concatenation of the input image $\mathbf{x}$ and the attention mask in logarithmic units, $\log \mathbf{m}_k$ as input. The CNN layers output (32, 32, 64, 64) channels respectively. The CNN output is flattened and fed to a 2 layer MLP with output sizes of (256, 32). The MLP output parameterises the μ and $\log \sigma$ of a 16-dim Gaussian latent posterior.

The VAE uses a broadcast decoder [Watters et al., 2019] to transform the sampled latent vector $\mathbf{z}_k$ into the reconstructed image component and mask distributions. The input to the broadcast decoder is a spatial tiling of $\mathbf{z}_k$ concatenated with a pair of coordinate channels – one for each spatial dimension – ranging from -1 to 1. These go through a four-layer CNN with no padding, 3x3 kernels, stride 1, 32 output channels and ReLU activations. The height and width of the input to this CNN were both 8 larger than the target output (i.e. image) size to arrive at the target size (i.e. accommodating for the lack of padding). A final 1x1 convolutional layer transforms the output into 4 channels: 3 RGB channels for the means of the image components $\hat{\mathbf{x}}_k$, and 1 for the logits used for the softmax operation to compute the reconstructed attention masks $\hat{\mathbf{m}}_k$. For all experiments, the output component distribution was an independent pixel-wise Gaussian with fixed scales.

For the *MONet* experiments, the first "background" component scale was fixed at $\sigma_{bg} = 0.09$, and for the $K - 1$ remaining "foreground" components, the scale was fixed at $\sigma_{fg} = 0.11$. The loss weights were $\beta = 0.5$, $\gamma = 0.5$:

For the component VAE experiments in Section. 2.2, a single scale $\sigma = 0.05$ was used for all K components, and we used $\beta = 0.5$, $\gamma = 0.25$.

B.2 Attention network

At the kth attention step, the attention network receives the concatenation of the input image $\mathbf{x}$ and the current scope mask in log units, $\log \mathbf{s}_k$, as input.

We used a standard U-Net [Ronneberger et al., 2015] blueprint with five blocks each on the downsampling and upsampling paths (except for the *CLEVR* experiments which used six blocks on each path). Each block consists of the following: a 3x3 bias-free convolution with stride 1, followed by instance normalisation [Ulyanov et al., 2016] with a learned bias term, followed by a ReLU activation, and finally downsampled or upsampled by a factor of 2 using nearest neighbour-resizing (no resizing occurs in the last block of each path).

Skip tensors are collected from each block in the downsampling path after the ReLU activation function. These are concatenated with input tensors along the upsampling blocks before the convolutional layer.

A 3-layer MLP serves as the non-skip connection between the downsampling and upsampling paths with its final output dimensionally matching that of the last skip tensor. The intermediate hidden layers were sized (128, 128). The input to the MLP is the last skip tensor collected from the downsampling path (after flattening). A ReLU activation is applied after all three output layers. The final output is then reshaped to match that of the last skip tensor, concatenated with it, and finally fed into the upsampling path.

Following the upsampling path, a final 1x1 convolution with stride 1 and a single output channel transforms the U-Net output into the logits for $\boldsymbol{\alpha}_k$. Both $\log \boldsymbol{\alpha}_k$ and $\log(1 - \boldsymbol{\alpha}_k)$ are computed directly in log units from the logits (using the log softmax operation). Each

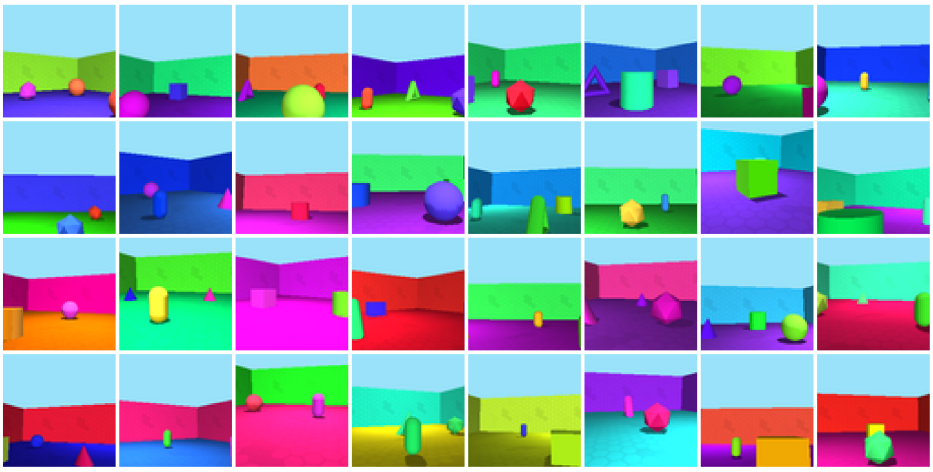

Figure 10: The *Objects Room* dataset. 32 random examples shown. See Section. C for more details.

are added to the current scope (also maintained in log units) $\log \mathbf{s}_{k-1}$ to compute the next (log) attention mask $\log \mathbf{m}_k$ and next (log) scope $\log \mathbf{s}_k$, respectively. Also see Section 2.1 for equations describing the recurrent attention process.

B.3 Optimisation

All network weights were initialized by a truncated normal (see [Ioffe and Szegedy, 2015]), and biases initialized to zero. All experiments were performed in TensorFlow [Abadi et al., 2015], and we used RMSProp for optimisation with a learning rate of 0.0001, and a batch size of 64.

C Datasets

The *Objects Room* dataset was created using a Mujoco environment adapted from the Generative Query Networks datasets [Eslami et al., 2018]. It consists of 64x64 RGB static scene images of a cubic room, with coloured walls, floors and a number of objects visible (see examples in Figure 10). A blue sky is visible above the wall. The camera is randomly positioned on a ring inside the room, always facing towards the centre but randomly oriented vertically, uniformly in $(-25°, 22°)$. The wall, the floor and the objects are all coloured randomly, with colours each sampled uniformly in HSV colour-space, with H in $(0, 1)$, S in $(0.75, 1)$ and V always 1.0. The objects are randomly sized, shaped (six different shapes), and arranged around the room(avoiding overlap). In the training dataset there are 3 objects in the room (making 1-3 objects visible).

The *Multi-dSprites* experiments used a dataset of 64x64 RGB images of 1-4 random coloured sprites. These were generated by sampling 1-4 images randomly from the 64x64 dSprites dataset [Matthey et al., 2017], colourising the sprites with a uniform random RGB colour, and compositing those (with occlusion) onto a uniform random RGB background.

For the *CLEVR* experiments we used images from the *CLEVR* dataset [Johnson et al., 2017]. The standard images are 320x240, and we crop those at y-coordinates (29, 221), bottom and top, and at x-coordinates (64, 256) left and right (creating a 192x192 image). We then resize that crop using bilinear interpolation to 128x128.