Learning to Manipulate Individual Objects in an Image

Yanchao Yang*
UCLA Vision Lab
yanchao.yang@cs.ucla.edu

Yutong Chen†*
Tsinghua University
chen-yt16@mails.tsinghua.edu.cn

Stefano Soatto
UCLA Vision Lab
soatto@cs.ucla.edu

Abstract

We describe a method to train a generative model with latent factors that are (approximately) independent and localized. This means that perturbing the latent variables affects only local regions of the synthesized image, corresponding to objects. Unlike other unsupervised generative models, ours enables object-centric manipulation, without requiring object-level annotations, or any form of annotation for that matter. The key to our method is the combination of spatial disentanglement, enforced by a Contextual Information Separation loss, and perceptual cycle-consistency, enforced by a loss that penalizes changes in the image partition in response to perturbations of the latent factors. We test our method's ability to allow independent control of spatial and semantic factors of variability on existing datasets, and also introduce two new ones which highlight the limitations of current methods.[1]

1. Introduction

Generative models typically aim to capture the natural statistics while isolating independent factors of variation. This can be beneficial if such factors correspond to variables of interest in tasks to be instantiated *post-hoc*, or if the model is to be used for image synthesis where the user wants to independently control the outcome. Generative models learned from large image collections, for instance variational auto-encoders (VAEs) or generative adversarial networks (GANs) do isolate independent factors of variation, but those affect the global statistics of the image. We are interested in spatially-localized factors of variation, so that manipulation of image statistics can occur at the level of *objects*, rather than of the whole image. While one could learn conditional generative models, this usually requires annotation of the independent factors. We aim to learn spatially and semantically independent latent factors without the need for any annotation. We call these *object-centric generative factors*.

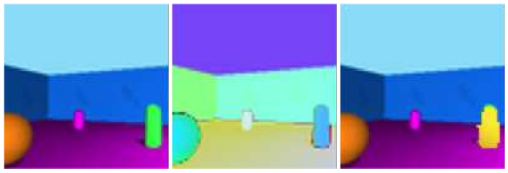

Figure 1. Perturbing the factors learned without knowing objects affects the synthesized scene globally (middle). Object-centric generative factors enable changing the color of the pillar from green (left) to yellow without affecting the other objects (right).

The existing literature on object-centric generative models is restricted to piece-wise constant or smooth images. We introduce two variational constraints, one derived from the Contextual Information Separation (CIS) principle [36], but extended to multiple objects, and one derived by enforcing perceptual cycle-consistency, which is that the partition of the image into independently controlled region is stable with respect to perturbations of the latent factors. We illustrate the characteristics of our model on existing datasets, and introduce two new datasets of increased complexity.

2. Related Work

There are two approaches to representation learning, *task driven*, where the goal is to learn a function of the data that captures all information relevant to the task and discards everything else (sufficient invariant) [1], and *disentangled*, where the goal is to reconstruct the data while separating the independent factors of variation [3, 32]. Technically speaking, the latter is a special case of the former, when the task variable is the data itself, and the independence of latent factor can be framed as the secondary task. However, the literature has largely progressed on separate tracks. An independent taxonomy can be devised based on the level of supervision. While disentangled representations usually refer to unsupervised learning, task-driven representations can be unsupervised (if the task is, for instance, prediction, or reconstruction), semi-supervised, or fully supervised. Among unsupervised representation learning methods, variational autoencoders (VAE) [24] attempt to extract "meaningful"

*Equal contribution.
†Work is done during the author's visit at UCLA.
[1]Code available at: https://github.com/ChenYutongTHU/Learning-to-manipulate-individual-objects-in-an-image-Implementation

latent factors by forcing a variational bottleneck in the generative model. This can also be seen as a special case of task-driven representation, where the task is the data itself. It was shown by [1, 30] that the latent factors tend to be independent, and can be easily manipulated in a generative setting. A different approach uses an adversarial loss [12] to map a known distribution (typically a Gaussian) to an approximation of the data distribution, so the input distribution can be considered a set of independent factors. To encourage the alignment of the learned representation with the underlying generative factors, several constraints have been proposed to enforce the disentanglement of the latent codes [17, 5, 1, 30, 22, 9, 4, 18]. InfoGAN [10] promotes disentanglement by explicitly maximizing the mutual information between a subset of latent variables and the generated images. Also, domain-specific knowledge could be utilized to learn disentangled representations [33, 32, 21].

In all these methods, disentanglement is sought in latent space, with no grounding on the domain where the data is defined. For the case of images, we would like the independent factors to correspond to compact and simply-connected regions of the image, corresponding to *objects*. In all these methods, manipulation of the independent factors typically has global effects on the image, rather than enabling manipulation of one object at a time. We would like to develop models that naturally disentangle the spatial domain, in addition to other latent factors of variation.

Of course, one could partition the image and independently represent each region, but doing so would require the ability to detect objects in the first place. Recent methods on structured representation learning have been proposed to enable reasoning about objects in the scene. AIR [11] proposes a structured generative model for the image generation process. SQAIR [25] proposes an additional state-space model to enforce temporal consistency such that the decomposition within a sequence is consistent. The capacity of their structured models and the assumption of known/trivial background restrict both AIR and SQAIR. DRAW [15] employs attention to generate images with objects in a structured manner. In [37], an image is decomposed into semantic mask, texture, and geometry, requiring heavy manual supervision. And [26] proposes an auto-encoder that de-renders images into graphics code, which is trained by explicitly specifying the variations. Similarly, [35] employs a graphics engine as the decoder to enforce an interpretable representation, which is, however, not backward-differentiable. NEM [14, 34] construct spatial mixture models to cluster pixels into objects, but only for grayscale images. IODINE [13] also employs a spatial mixture model to jointly infer the segmentation and object representation. [13] does not perform well on textured or cluttered scene, culprit the assumption that pixels can be grouped into objects according to the low dimensional spatial mixture models.

Semantic segmentation has improved considerably since the advent of deep neural networks [28, 8, 38] and so has instance segmentation, where the network has to distinguish between different instances of the same semantic class [31, 19, 27, 7]. However, these methods depend on densely annotated ground-truth segmentation masks. On the other hand, unsupervised object segmentation has been a long-standing problem, but most of the methods require complicated optimization during inference. Recently, unsupervised learning methods for object segmentation have shown promise: UMODCIS [36] proposes contextual information separation for binary moving object detection, with end-to-end training without manual supervision or pseudo masks. Later, [2] proposes Copy-Pasting GAN to discover binary object masks in images; however, special care has to be taken to prevent trivial solutions.

There is only a handful of work dealing with both segmentation and object-centric representation learning in a unified framework. Besides the few employing variational inference with a structured model, MONet [6] introduces a recurrent segmentation network within the VAE framework, and trains them jointly to provide segmentation and learned representation.

In the next section, we describe our method and in the following Sect. 4 we test the model's ability to capture the statistics of the data while enabling independent control of latent factors corresponding to objects in the scene.

3. Method

Let $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ be a color image, and $\mathbf{z} \in \mathbb{R}^N$ be the generative factor of $\mathbf{x}$, which represents different characteristics of the data. Our model uses as an inference criterion the Information Bottleneck when the task is the data itself, whereby a representation (encoder) $q_\phi(\mathbf{z}|\mathbf{x})$ describes the latent factors (bottleneck) $\mathbf{z}$, and a decoder $p_\theta(\mathbf{x}|\mathbf{z})$ allows sampling images from the latent factors $\mathbf{z}$. The encoder and decoder are trained by minimizing the Information Bottleneck Lagrangian (IBL) [1]:

$$\mathcal{L}(\phi, \theta; \mathbf{x}, \beta) = -\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] + \beta \mathbb{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) \quad (1)$$

where $\mathbb{KL}$ is the Kullback–Leibler divergence, and $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, I)$ is usually a zero mean unit variance Gaussian. When $\beta = 1$ the IBL reduces to the Evidence Lower Bound (ELBO). For $\beta > 1$, [1] shows analytically and [17] validates empirically, that the latent factors are disentangled. However, the factors always affect the generated images globally as shown in Fig. 1. Our goal here is to infer object-centric generative factors, such that we can perturb the factors associated with a single object, and the perturbation will not affect other objects or entities in the scene.

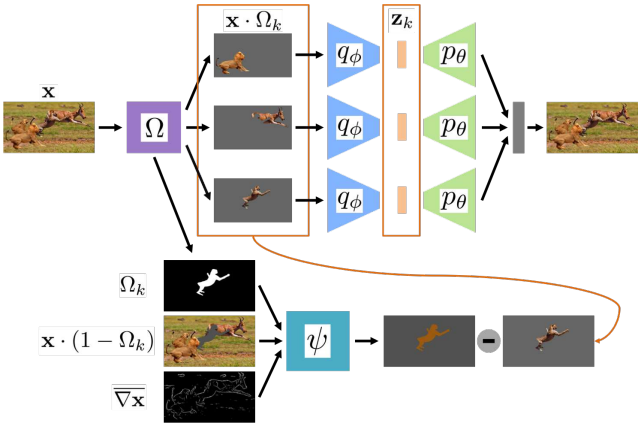

Figure 2. *System Overview.* Our method works by partitioning the image domain into mutually independent regions using the Contextual Information Separation criterion, which entails an inpainting network, and then extracting the generative factors disentangled both spatially and statistically with the identity consistency enforced by the perceptual cycle-consistency constraint. We omit the masks for simplicity.

In other words, our goal is to make the generative factors disentangled not only statistically, but also spatially.

To this end, we construct a segmentation network to map a color image to K segmentation masks Ω_k's, with K the maximum number of objects (including background) in the scene:

$$\Omega : \mathbb{R}^{H\times W\times 3} \to [0,1]^{H\times W\times K}; \sum_k \Omega(i,j,k) = 1, \forall i,j \tag{2}$$

Note that $\sum_{i,j} \Omega_k(i,j) = 0$ simply means that there is no object in the k-th channel. If all the non-zero channels in Ω represent exactly the segmentation masks of the objects, then we can presumably learn the object-centric generative factors $\mathbf{z}_k$'s as follows:

$$\mathcal{L}(\phi^\alpha, \phi^s, \theta; \Omega, \mathbf{x}, \beta, \lambda) = \sum_k \mathcal{L}(\phi^\alpha, \phi^s, \theta; \Omega_k, \mathbf{x}, \beta, \lambda) \tag{3}$$

with

$$\begin{aligned}\mathcal{L}(\phi^\alpha, \phi^s, \theta; \Omega_k, \mathbf{x}, \beta, \lambda) = \\ - \mathbb{E}_{\mathbf{z}_k \sim q_{\phi^\alpha}\cdot q_{\phi^s}} \log p_\theta([\mathbf{x}\cdot\Omega_k, \Omega_k] \mid \mathbf{z}_k) \\ + \beta \mathbb{KL}(q_{\phi^\alpha}(\mathbf{z}^\alpha_k \mid \mathbf{x}\cdot\Omega_k) \| p(\mathbf{z}^\alpha_k)) \\ + \lambda \mathbb{KL}(q_{\phi^s}(\mathbf{z}^s_k \mid \Omega_k) \| p(\mathbf{z}^s_k))\end{aligned} \tag{4}$$

where q_{ϕ^α} and q_{ϕ^s} are the encoders for appearance and shape related factors of objects in $\mathbf{x}$ respectively. Then the joint decoder θ reconstructs the appearance and mask for each object using $\mathbf{z}_k = \{\mathbf{z}^\alpha_k, \mathbf{z}^s_k\}$, which is the union of the appearance and shape related factors. Note that, Eq. (3) is a summation of the Information Bottleneck Lagrangians Eq. (4) defined on individual segments or "objects". A similar loss is also used in [6] to learn object related representations in an unsupervised manner. However, a question arises from Eq. (3): *Why would minimizing the above loss yield a segmentation network Ω that partitions the image domain into objects?* Given a small enough encoding capacity, it may be true that Ω will be biased to partition the image $\mathbf{x}$ into pieces that are easier to encode and decode than the full image, while minimizing the first term in Eq. (4), which represents the reconstruction error. Then Ω may succeed when objects happen to be constant color blobs, as they appear in some datasets, *e.g.*, Multi-dSprites and Objects Room [20], used for experiments in [6, 13]. However, what if we want to apply our method on textured objects or cluttered scenes, which are simply not color blobs?

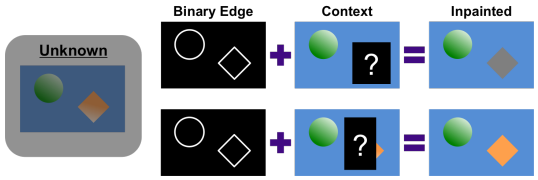

Figure 3. *The average inpainting error using context conditioned on binarized edge map is a good measure of the contextual information computed with the edge conditionals*: Given the binary edge map of an unknown image, the average inpainting error of the masked out region (question mark) will be larger when the context contains less mutual information (the first row), smaller with more mutual information in the context (second row).

Spatial Disentanglement. To endow our method with the ability to learn object-centric generative factors in realistic scenarios, we adapt the Contextual Information Separation (CIS) criterion of [36], which obviates the shortcomings of Eq. (4). Instead of a binary segmentation, we extend CIS to multiple objects with the number of objects unknown, and also combine it with the representation learning loss in Eq. (3), which in turn imposes additional regularization through the representational bottleneck. This way, statistical and spatial disentangling of the generative factors occur simultaneously during learning.

The basic idea of CIS is that, when the context contains no information about a sub-region of an image, the (conditional) reconstruction or "inpainting" error will be maximized, as shown in Fig. 3. In order to measure the mutual information, a joint distribution between a region and its context has to be specified. Here, we choose to use the conditional distribution $p(\mathbf{x} \mid \overline{\nabla \mathbf{x}})$ of an image $\mathbf{x}$ on the binarized edge $\overline{\nabla \mathbf{x}}$. Note that, one could also use the marginal distribution of images $p(\mathbf{x})$, which may result in degraded performance in general since the mutual information between pixels computed using $p(\mathbf{x})$ depends on the spatial proximity instead of the structure of the scene. By instantiating a conditional inpainting network ψ, our CIS-based

spatial disentanglement loss becomes:

$$\mathcal{L}_{SD}(\psi;\Omega,\mathbf{x}) = \sum_k \frac{\langle \Omega_k, \|\psi(\Omega_k, \mathbf{x}\cdot(1-\Omega_k); \overline{\nabla \mathbf{x}}) - \mathbf{x}\| \rangle}{\langle \Omega_k, \|\mathbf{x}\| \rangle + \epsilon} \tag{5}$$

where $\langle \cdot,\cdot \rangle$ represents the dot product, and $\|\cdot\|$ is the element-wise L_1 norm, which keeps the dimension of the input, ϵ is a small positive constant that prevents division by zero. In [36], it is shown that, under the assumption of Gaussian conditionals, the mutual information can be approximated with the inpainting error. Note the conditional inpainting network ψ takes in the condition $\overline{\nabla \mathbf{x}}$ and the context $\mathbf{x}\cdot(1-\Omega_k)$ of the image being masked out by Ω_k, and outputs the inpainted image. If Ω_k's perfectly separate the context from each object, this spatial disentanglement loss will be maximized, thus minimizing the mutual information between the inside and outside of Ω_k's.

Perceptual Cycle-Consistency. Given that the decoder p_θ generates images from the object-centric generative factors $\mathbf{z}_k$'s, we can perturb the factors of an appointed object $\hat{\mathbf{z}_k} \sim \mathcal{N}(\mathbf{z}_k, I)$ (Eq. (7)), and synthesize the perturbed image $\hat{\mathbf{x}}$ with $\{\mathbf{z}_k\}_{\{1,..K\}\setminus k} \cup \hat{\mathbf{z}_k}$ (Eq. (8)), and then extract the object-centric generative factors of the perturbed image (Eq. (9),(10)). If not only the factors are well disentangled statistically and spatially, but also the identities of the disentangled factors are robust to local perturbations, we would expect that the factors extracted from the perturbed image will be unchanged in $\mathbf{z}_k$'s of the other objects, and, also synchronize well with $\hat{\mathbf{z}_k}$, which suggests the following perceptual cycle-consistency loss to further promote disentanglement and identity consistency:

$$k \sim \text{Uniform}(1, \text{K}) \tag{6}$$

$$\hat{\mathbf{z}_k} \sim \mathcal{N}(\mathbf{z}_k, I) \tag{7}$$

$$\hat{\mathbf{x}} \leftarrow p_\theta([\mathbf{x}, \Omega] | \{\mathbf{z}_k\}_{\{1,..K\}\setminus k} \cup \hat{\mathbf{z}_k}) \tag{8}$$

$$\hat{\Omega} = \Omega(\hat{\mathbf{x}}) \tag{9}$$

$$\{\bar{\mathbf{z}_k}\} \xleftarrow{q_{\phi^\alpha}, q_{\phi^s}} \hat{\mathbf{x}}, \hat{\Omega} \tag{10}$$

$$\mathcal{L}_{PC}(\phi^\alpha, \phi^s, \theta, \Omega, \mathbf{x}) = \sum_{l \neq k} \|\mathbf{z}_l - \bar{\mathbf{z}}_l\| + \|\hat{\mathbf{z}_k} - \bar{\mathbf{z}_k}\| \tag{11}$$

Note this characteristic is also desired when we need to track the status of different objects for temporal analysis. By combining Eq. (3), (5) and (11) we have the final training loss for our model:

$$\arg\max_{\psi} \min_{\phi^\alpha, \phi^s, \theta, \Omega} \mathcal{L}(\phi^\alpha, \phi^s, \theta; \Omega, \mathbf{x}, \beta, \lambda) - \gamma \mathcal{L}_{SD}(\psi; \Omega, \mathbf{x}) + \eta \mathcal{L}_{PC}(\phi^\alpha, \phi^s, \theta, \Omega, \mathbf{x}) \tag{12}$$

Note that, the segmentation network Ω appears now in three terms, which encourage Ω to partition the image (first term) while minimizing contextual information (second term) to prevent over-segmentation; *i.e.*, pixels belonging to the same object should be grouped together. Moreover, it has to be robust to perturbations introduced by the third term, which not only imposes perceptual consistency, but can also prevent identity switching; *i.e.*, the object assigned to the k-th mask Ω_k (identified as k) should be assigned to Ω_k again after the perturbations, especially the spatial ones. This is particularly useful for applications involving video, since temporal consistency will be automatically achieved after training, and we will show its effectiveness in the experimental section.

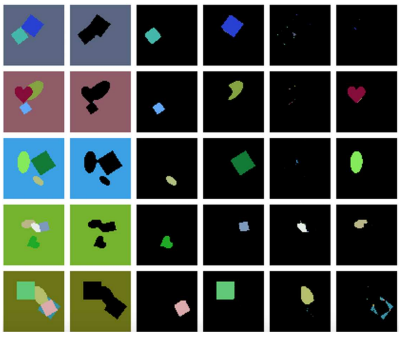

Figure 4. *Spatial Disentanglement on Multi-dSprites*: Our method can segment images containing various numbers of constantly colored objects with heavy occlusions (last two rows).

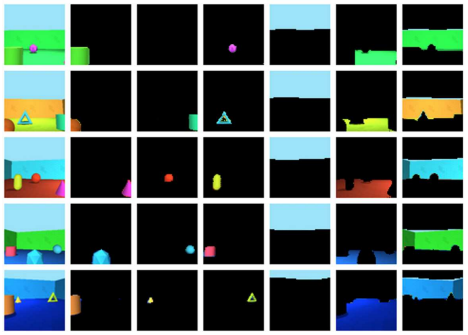

Figure 5. *Spatial Disentanglement on Objects Room*: Our method works on 3D scenes with smoothly colored objects, complex shapes, and different lighting conditions.

4. Experiments

We first describe the datasets used for evaluation and then elaborate on the implementation details and the training procedure, after which qualitative and quantitative comparisons are provided.

4.1. Datasets

Multi-dSprites: dSprites [29] consists of binary images of a single object that varies in shape (square, ellipse, heart), scale, orientation, and position. In Multi-dSprites [20], 1-4 shapes are randomly selected from dSprites, randomly colorized and composed on a randomly colored background

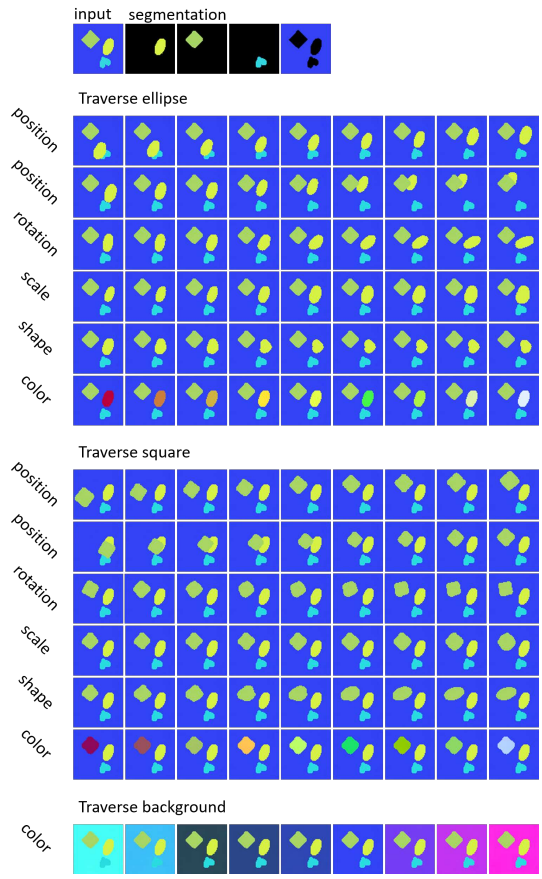

Figure 6. *Traversing the object-centric generative factors on Multi-dSprites*: the input image and the corresponding spatial disentanglement (first row). The rest: traversing the statistically disentangled generative factors of a specific object. Note that the perturbations only affect the object been targeted, and the color of the background can also be modified.

with occlusions and independent variations in position, scale, and rotation; see sample images in Fig. 4.

Objects Room [20] contains rendered images of 3D scenes consisting of 1 to 3 randomly chosen 3D objects that vary in shape, color, size, and pose independently. The wall and the floor of the 3D scene are also colorized randomly. Once projected, objects can exhibit significant appearance variability, depending on lighting and viewpoint. Examining the images from Objects Room in Fig. 5, we can still see that the images are far from realistic, even though the objects are not uniformly colorized.

Multi-Texture: To test whether our proposed method works on complex appearance, for example, textured objects, we create the Multi-Texture dataset. To generate this dataset, 1 to 4 shapes are randomly selected, and independently textured using randomly colorized chessboard patterns. Then, these textured objects are randomly placed on a randomly colorized wooden texture background (Fig. 8).

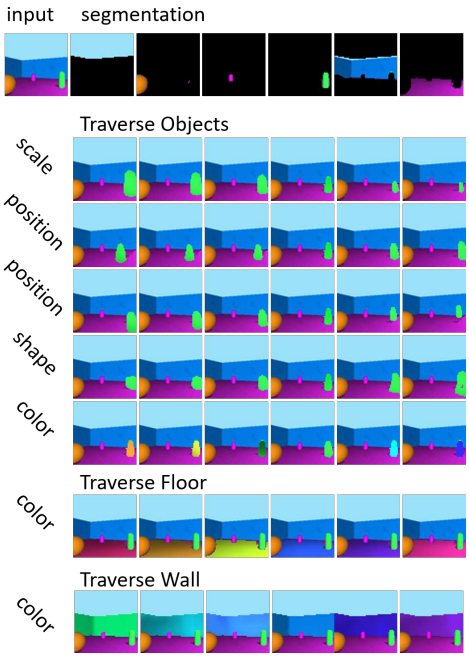

Figure 7. *Traversing the object-centric generative factors on Objects Room*: We can change the scale, position, and color of the green pillar continuously without affecting the others. Also, its shape can deform from a circle to a triangle.

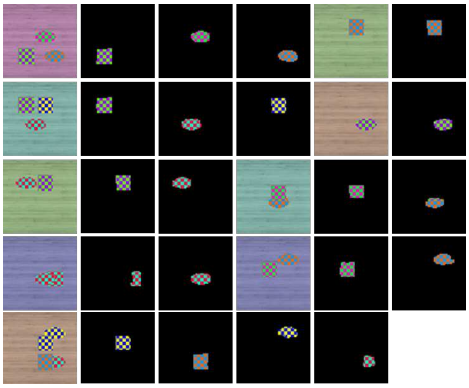

Figure 8. *Spatial Disentanglement on Multi-Texture*: with explicit spatial disentanglement via contextual information separation, our method can segment objects with complex textures.

Flying Animals: Although the Multi-Texture is more complex in the object appearance compared to the Objects Room dataset, the homogeneously textured objects still look unnatural in the image statistics, far from the images that would be seen in the real world. For this reason, we come up with the Flying Animals dataset. We collect two sets of natural images. One contains background images from 10 different landscapes, e.g., mountain, desert, and forest, each with 10 different instances; the other set contains clean foreground images of 24 different kinds of animals, each with 10 different instances. We select 1 to 5 objects, randomly scale and position them on a random

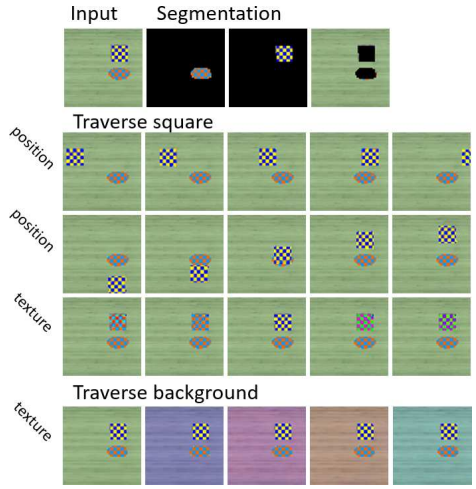

Figure 9. *Traversing the object-centric generative factors on Multi-Texture*: Although the objects are not constantly colored, disentangled object-centric generative factors can still be learned with explicit modeling of the spatial disentanglement.

background image with occlusions. Moreover, we perturb the intensity of each component to simulate different lighting conditions. For sample images please refer to Fig. 10.

4.2. Training Details

Segmentation Network Ω: Similar to the DeepLabV2 architecture [8], we use ResNet50 [16] as the backbone for our segmentation network, which is followed by four dilated convolution layers in parallel, whose responses are aggregated to generate the K segmentation masks. The total number of trainable parameters is 24M.

Inpainting Network ψ: We adapt the inpainting network from [36]. It consists of two symmetric encoders that encode binarized edge map and masked image (context), respectively; and a joint decoder with skip-connections from the two encoders. The total number of parameters in the inpainting network is 13M.

Encoder and Decoder $\phi^\alpha, \phi^s, \theta$: We adapt the VAE structure proposed in [22]. Instead of a single encoder for images, we instantiate two symmetric encoders ϕ^s and ϕ^α, where ϕ^s encodes the one-channel object mask which is the output of the segmentation network, and ϕ^α encodes the masked object to get the appearance-related generative factors. The decoder takes in the object-centric generative factors and generates the objects' appearance and masks, which are then concatenated and fused through four convolutional layers with relu and sigmoid activations to synthesize images over the whole image domain. The total number of parameters in our encoder-decoder is 1.7M.

Training: Adam is used [23] for all modules with initial learning rate 1e-4, epsilon 1e-8, and beta (0.9,0.999). As in [36], we find that a pretrained inpainting network will stabilize the training. We randomly crop the input images using rectangular masks with varying height, width, and position, and train the inpainting network to minimize the inpainting error (L_1) within the masked region. The training stops after 50K steps. Then, we update the segmentating network and the inpainting network adversarially to speed up the spatial disentanglement before the joint training of all modules that is performed in an adversarial manner as shown in Eq. (12) and stops after 4M iterations. The capacity constraint is adjusted during training following the scheme proposed in [5].

4.3. Results

The closest method to ours is MONet [6], which is, to the best of our knowledge, the only one to learn segmentation and representation in a unified framework for non-constantly colored objects. Since we do not have access to the native implementation, we re-implemented MONet by training the same K-way segmentation network in our framework but using the same loss as in [6]. This also eliminates the structural bias that could prevent a fair comparison. Note that we set $K = 6$, which is larger than the maximum number of objects that could appear in the training of MONet. In the following, we show the segmentation and learned object-centric generative factors on each dataset. We will also show quantitative evaluations of the jointly learned object segmentation masks.

Multi-dSprites: As shown in Fig. 4, our approach manages to separate different objects such that the VAE can learn representations for every single object and background. Note that the unsupervised segmentation works well with an unknown number of objects and heavy occlusions. Given that spatial disentanglement is achieved through segmentation, we forward each masked object and background into the encoder-decoder and obtain the object-centric factors at the bottleneck. We observe that some dimensions diverge from the prior Gaussian distribution during the training process. As explained in [5], these dimensions exhibit semantic meanings aligned with the independent generative factors of dSprites. Fig. 6 displays the segmentation and object-centric disentanglement for an image with three objects. Objects can be manipulated one at a time by perturbing one's latent factors while keeping other objects' representation unchanged. For each object, we can control its independent factors, including positions along two orthogonal axes, rotation, scale, shape, and color, by traversing one dimension in the latent space one at a time.

Objects Room: Our approach also performs well on the Objects Room dataset, even with shading effects on different objects under various lighting conditions, as shown in Fig. 5. Object-centric statistical disentanglement is presented in Fig. 7. Similarly, we can edit the scene by changing the position, shape, and color of different objects individually, which shows the applicability on 3D scene editing.

Figure 10. *Spatial Disentanglement on Flying Animals.* Our method can spatially disentangle images with natural statistics, where the objects and background are highly non-homogeneous, and also the shape of the objects exhibits more variations than squares or ellipses.

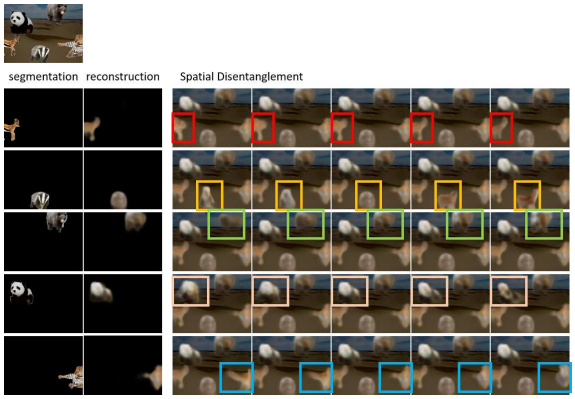

Figure 11. *Traversing the object-centric generative factors on Flying Animals*: The spatial disentanglement of the input image (top-left) is shown in the first column and the second column displays the reconstructed objects by the decoder. The other columns show the traversal on each object. Still, we can change the shape or appearance of each spatially disentangled object individually. For example, in the second row, when perturbing the badger's representation, the appearance of the animal in the yellow box interpolates from owl-like to fox-like while the other four animals and the background remain unchanged.

Multi-Texture: We experiment on the Multi-Texture dataset to demonstrate that our proposed method enables spatial disentanglement on textured images. As shown in Fig. 8, our approach can accurately segment out squares and ellipses with chessboard texture, confirming that the Contextual Information Separation constraint prevents the network from naively splitting the chessboard into two different colors which correlate with each other. Fig. 9 shows the disentanglement and the object-centric manipulation results. Even with complex textured objects, our method still enables learning the factors that allow us to change the object consistently, including the background.

Flying Animals: To verify that our method is not restricted to synthesized images but can also deal with natural ones, we further test our method on the Flying Animals dataset with real landscapes and animals. As shown in Fig. 10, even with complex appearance and shape, our approach can again segment out animals from the natural landscapes, which is far more challenging than segmenting uniformly colored objects as in Multi-dSprites and Objects Room. Similarly, our method can learn disentangled representations for every single animal in the scene and then edit the animals one at a time, shown in Fig. 11. However, due to the complexity of the appearance and shape, and the trade-off between reconstruction quality and bottleneck capacity for disentanglement, the β-VAE framework is not powerful enough to conduct statistical disentanglement for each animal while maintaining the details of the object. We will discuss this further in the next section.

Dataset	M-dSprites	Obj-Room	M-Texture	F-Animals
MONet	$0.84 \pm 6.4\delta$	$0.80 \pm 8.3\delta$	$0.37 \pm 0.3\delta$	$0.18 \pm 2.8\delta$
Ours	$0.92 \pm 6.6\delta$	$0.85 \pm 5.6\delta$	$0.88 \pm 2.6\delta$	$0.81 \pm 5.5\delta$

Table 1. *Quantitative evaluation of the segmentation quality between MONet [6] and our method.* Performance measured in the mean intersection-over-union score, reported with mean and variance, where $\delta = 10^{-3}$. MONet performs well on Multi-dSprites and Objects Room (constantly or smoothly colored), but its performance drops significantly on Multi-Texture and Flying Animals (textured or complex natural appearance). Our method performs robustly well across different datasets.

Perceptual Cycle-Consistency	No	Yes
rate of identity switching	21/255	0/255

Table 2. *Quantitative evaluation on identity switching*: Note the identity switching decreases to 0 out of 255 by imposing the perceptual cycle-consistency constraint.

Quantitative Evaluation: We compare our method with MONet [6] re-implemented on the four datasets mentioned above. We report the mean-intersection-over-union score (mean$\pm$variance). As shown in Table 1, our approach achieves better scores than MONet on all four datasets. Particularly, in Multi-Texture and Flying Animals, where objects have complex texture, without explicit information separation (CIS) constraint, MONet tends to segment the images based mainly on color information naively. At the same time, CIS enables our model to "see" objects as entities with different parts correlated with each other. To illustrate this, we present the segmentation results in Fig. 12. Note that MONet dissects the black-and-white panda into different channels based on color, while our method successfully detects it without being biased by its color.

Perceptual Cycle-Consistency We expect our model to exhibit the perceptual consistency mentioned in Section 3,

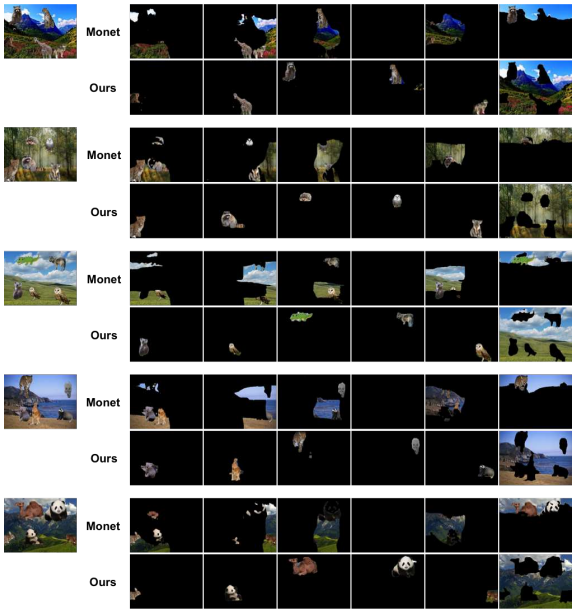

Figure 12. *Comparison of the segmentation* between our approach and MONet, which fails to capture objects with natural statistics due to the lack of explicit spatial disentanglement.

which means that in a temporally coherent sequence, each segmentation channel should keep track of the same object without identity switching. To verify the effectiveness of the perceptual cycle-consistency constraint, we train two segmentation networks on the Multi-Texture dataset, one with perceptual cycle-consistency but not the other. Then, we generate 256 sample sequences with varying positions and occlusions. Fig. 13 compares the two networks' behavior by visualizing their first output channels on the sequence in the top row. Without perceptual cycle-consistency, the first output channel mainly detects the ellipse but can switch to the square occasionally, especially when the two objects come close. However, with the perceptual cycle-consistency constraint enabled, the segmentation network can have each output channel focus on a fixed target throughout the whole sequence, with no identity switching. We evaluate the two network's performance by counting the number of target switches, shown in Table 2, which verifies that the proposed consistency constraint improves the temporal coherence of the generative factors.

5. Discussion

The evaluation of a method that aims at "disentanglement" is subjective, since we do not know what the model will be used for: It is common to hope that the hidden variables correspond to known components of the image-formation process, such as pose, scale, color, and shape. However, making that a quantitative benchmark may be misleading since, if that were the goal, we would simply capture those factors explicitly, for instance, via a conditional generative model. What we do observe is that the perceptual cycle-consistency, explicitly enforced in our model, enables the persistence of the representation, so identities of objects are not switched in different views. This would enable temporal consistency when the model is used as a prior in a sequential setting as shown in Fig. 13.

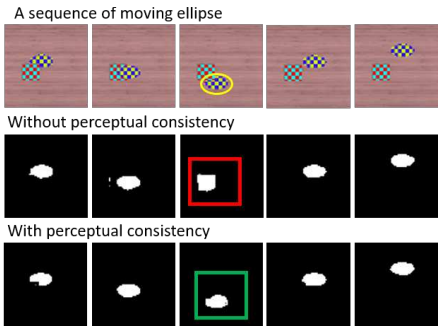

Figure 13. *Effectiveness of the Perceptual Cycle-Consistency*: Note that an object will be assigned to different channels of the segmentation network from time to time (second row), showing temporal incoherence in the spatial disentanglement, however, the proposed perceptual cycle-consistency eliminates this incoherence, making the status of objects trackable.

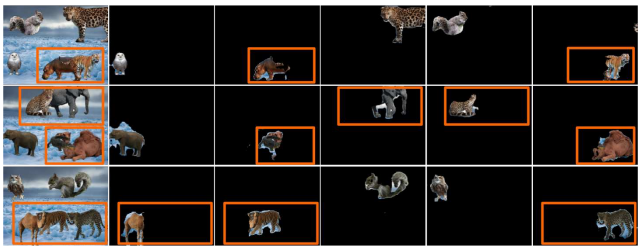

Figure 14. *Occlusion affects the accuracy of the spatial disentanglement on the Flying Animals dataset.* Orange boxes highlight the regions where occlusion happens and the affected objects.

Our model has limitations. The use of a VAE forces a hard trade-off between capacity, which affects the quality of the reconstructed image, and disentanglement, which is forced by the bottleneck. For complex scenes, there may not be a broad range of the trade-off parameter over which the model both captures the image statistics faithfully, and separates the hidden factors. Another limitation is the power of the inpainting model. For highly textured or complex scenes, the un-occluded region requires capturing the fine-grained context at a level of granularity higher than what our model affords, which may make it difficult for the segmentation network to learn perfect segmentation when occlusion happens, as shown in Fig. 14.

Acknowledgements

Research supported by ONR N00014-17-1-2072 and N00014-19-1-2229.

References

[1] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018, https://arxiv.org/abs/1706.01350. 1, 2

[2] Relja Arandjelović and Andrew Zisserman. Object discovery with a copy-pasting gan. *arXiv preprint arXiv:1905.11369*, 2019. 2

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 1

[4] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[5] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018. 2, 6

[6] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. 2, 3, 6, 7

[7] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2018. 2

[8] 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, 2017. 2, 6

[9] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018. 2

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016. 2

[11] SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pages 3225–3233, 2016. 2

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[13] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 2424–2433, 2019. 2, 3

[14] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, pages 6691–6701, 2017. 2

[15] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. DRAW: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, pages 1462–1471, 2015. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 6

[17] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017. 2

[18] Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. In *Advances in neural information processing systems*, pages 1878–1889, 2017. 2

[19] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. Maskrnn: Instance level video object segmentation. In *Advances in Neural Information Processing Systems*, pages 325–334, 2017. 2

[20] Rishabh Kabra, Chris Burgess, Loic Matthey, Raphael Lopez Kaufman, Klaus Greff, Malcolm Reynolds, and Alexander Lerchner. Multi-object datasets. https://github.com/deepmind/multi-object-datasets/, 2019. 3, 4, 5

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2

[22] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, pages 2654–2663, 2018. 2, 6

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR*, 2015. 6

[24] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014. 1

[25] Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, pages 8606–8616, 2018. 2

[26] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547, 2015. 2

[27] 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. 2

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[29] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017. 4

[30] Ori Press, Tomer Galanti, Sagie Benaim, and Lior Wolf. Emerging disentanglement in auto-encoder based unsupervised image content transfer. In *7th International Conference on Learning Representations, ICLR*, 2019. 2

[31] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *European conference on computer vision*, pages 312–329. Springer, 2016. 2

[32] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1415–1424, 2017. 1, 2

[33] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 2

[34] Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *6th International Conference on Learning Representations, ICLR*, 2018. 2

[35] Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. Neural scene de-rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 699–707, 2017. 2

[36] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 879–888, 2019. 1, 2, 3, 4, 6

[37] 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 1887–1898, 2018. 2

[38] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018. 2