Efficient Iterative Amortized Inference for Learning Symmetric and Disentangled Multi-Object Representations

Patrick Emami [1] **Pan He** [1] **Sanjay Ranka** [1] **Anand Rangarajan** [1]

Abstract

Unsupervised multi-object representation learning depends on inductive biases to guide the discovery of object-centric representations that generalize. However, we observe that methods for learning these representations are either impractical due to long training times and large memory consumption or forego key inductive biases. In this work, we introduce EfficientMORL, an efficient framework for the unsupervised learning of object-centric representations. We show that optimization challenges caused by requiring both symmetry and disentanglement can in fact be addressed by high-cost iterative amortized inference by designing the framework to minimize its dependence on it. We take a two-stage approach to inference: first, a hierarchical variational autoencoder extracts symmetric and disentangled representations through bottom-up inference, and second, a lightweight network refines the representations with top-down feedback. The number of refinement steps taken during training is reduced following a curriculum, so that at test time with zero steps the model achieves 99.1% of the refined decomposition performance. We demonstrate strong object decomposition and disentanglement on the standard multi-object benchmark while achieving nearly an order of magnitude faster training and test time inference over the previous state-of-the-art model.

1. Introduction

Deep learning has produced impressive results across multiple domains by taking advantage of enormous amounts of data and compute. However, it has become clear that the *representations* these models learn have fundamental limitations. Consider the problem of inferring a representation for a multi-object scene. Most humans can observe a scene and then manipulate the individual objects in their mind—perhaps imagining that a chair has suddenly flipped upside down. This example illustrates that the common approach of summarizing an entire scene as a *single* distributed representation (Hinton et al., 1986) is likely insufficient. It has been shown that this approach fails on simple generalization tasks such as processing novel numbers of objects (Eslami et al., 2016; Watters et al., 2019a; Mao et al., 2019).

Alternatively, a scene can be encoded as a *set* of distributed *object-centric* representations using object detection (Zhou et al., 2019) or instance segmentation (He et al., 2017). While sets can handle arbitrary numbers of objects, these methods require ground truth supervision which limits reusability and generalization. Unsupervised approaches bring the promise of better generalization at the expense of relying on inductive biases to implicitly define the scene representation. In this paper, we aim to develop such a method that incorporates three key inductive biases that have been argued previously as being essential for object-centric reasoning and addressing the *binding problem* within deep neural networks (Greff et al., 2015; 2017; 2019; Locatello et al., 2020; Huang et al., 2020; Greff et al., 2020).

- **Symmetry** Multiple distributed representations are inferred for a single scene, each sharing a common format with the others (Greff et al., 2019; Locatello et al., 2020). This eases relational and compositional reasoning, e.g. for learning dynamics models (van Steenkiste et al., 2018; Veerapaneni et al., 2019).
- **Unordered** Any latent representation can take responsibility for any single object (Greff et al., 2019). Typically, a randomized iterative process is used to decide the assignment.
- **Disentangled** Manipulating one dimension of an object representation changes a single object property and leaves all else invariant (Schmidhuber, 1992; Higgins et al., 2017; 2018).

Prior attempts to incorporate all three inductive biases have been unsuccessful for a variety of reasons. Some do not enforce symmetry to avoid solving for the assignment (Burgess

[1]University of Florida, Gainesville, FL, USA. Correspondence to: Patrick Emami <pemami@ufl.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

et al., 2019; Engelcke et al., 2020) while others learn symmetric yet *entangled* representations which circumvents the challenge of disentangling latent factors (Locatello et al., 2020). To the best of our knowledge, IODINE (Greff et al., 2019) is the only model that has all three inductive biases. However, IODINE has computational concerns due to its use of iterative amortized inference (IAI) (Marino et al., 2018) to implement the assignment of pixels to symmetric representations. This translates to relatively long training times of over a week given a reasonable compute budget and slow test time inference.

In this work, we show that IAI *can* be used to solve multi-object representation learning while being as efficient as competing approaches *and* without sacrificing representation quality. Our idea is to cast the iterative assignment of inputs to symmetric representations as bottom-up inference in a multi-layer hierarchical variational autoencoder (HVAE). A hierarchical prior regularizes the bottom-up posterior, disentangling the latent space. We use a two-stage inference algorithm to obtain a scene representation; the first stage uses the HVAE, and the second stage uses IAI to simply refine the HVAE posterior. We find this is crucial for the HVAE to achieve reliable convergence to good local minima, particularly early on during training. At test time, IAI can optionally be discarded.

Contributions:

- EfficientMORL, a framework for *efficient* multi-object representation learning consisting of a hierarchical VAE and a lightweight network for iterative refinement
- Our method learns both *symmetric* and *disentangled* representations while being comparably efficient to state-of-the-art methods whose representations miss on at least one of the key inductive biases
- An order of magnitude faster training and test time inference than the closest comparable method

2. Related Work

Unsupervised image segmentation Algorithms for unsupervised multi-object representation learning can be broadly differentiated by the use of hand-crafted or learned features. Unsupervised image segmentation algorithms (Arbelaez et al., 2010; Achanta et al., 2012) predate modern deep approaches and used perceptual grouping ideas to define features for clustering pixels in human-interpretable ways. These algorithms are still used in vision pipelines, and although the community's focus has shifted to learning representations from data, they contain valuable insights for improving neural methods (Bear et al., 2020).

Spatial attention How neural approaches decompose scenes into object-centric representations largely segregates the relevant literature. AIR (Eslami et al., 2016), SPAIR (Crawford & Pineau, 2019), and SPACE (Lin et al., 2020) use spatial attention to discover explicit object attributes such as position and scale similarly to unsupervised object detection. These models excel at decomposing and generating synthetic scenes (Jiang & Ahn, 2020; Deng et al., 2021), but the underlying grid-based scene representation and symbolic bounding-box-like object representations are ill-suited for handling complex real-world scenes.

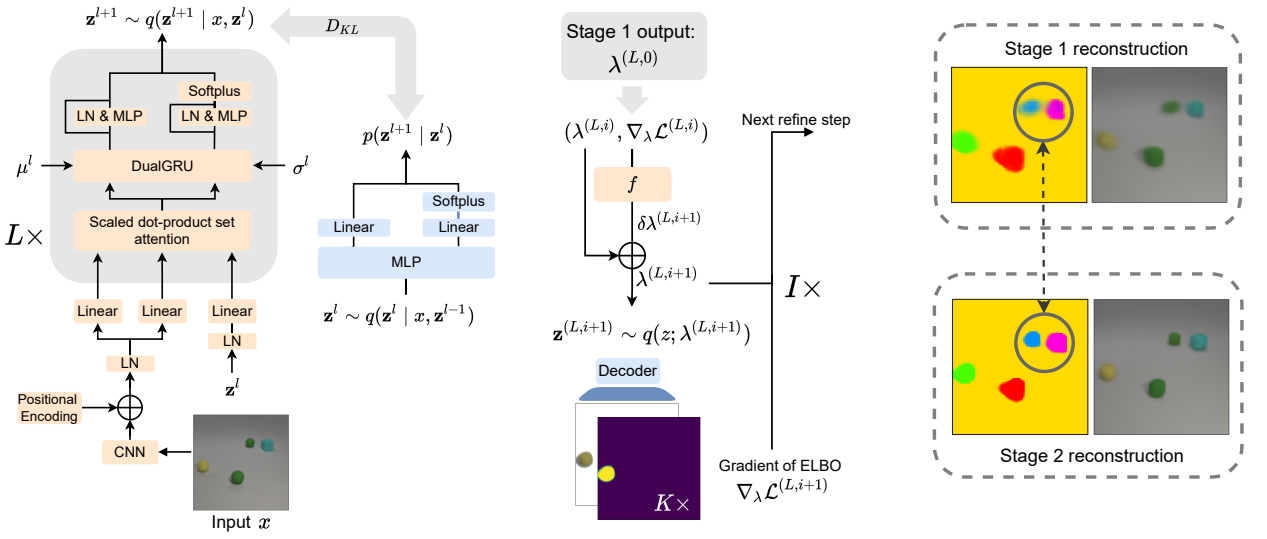
Sequential attention MONet (Burgess et al., 2019) and GENESIS (Engelcke et al., 2020) use sequential attention to bind latent variables to the components of a *segmented* image but as a result learn *ordered* representations. Imposing an ordering on the set of representations for a scene is unnatural, can leak global scene information into the object-centric representations, and biases the decomposition (e.g., the background or large objects are always attended to first).[1] In a follow-up work, MONet was extended with an in-painting network to improve its spatial disentanglement (Yang et al., 2020).

Iterative inference A line of methods (Greff et al., 2016; 2017; van Steenkiste et al., 2018; Yuan et al., 2019; Greff et al., 2019) use iterative inference to bind symmetric latent representations with the components of a segmentation mixture model. IODINE is the state-of-the-art method in this category; our method, described in the next section, presents an efficient alternative without sacrificing representation quality. Slot Attention (Locatello et al., 2020) is a general method for mapping a distributed representation to a symmetric set representation and has been used within a deterministic autoencoder for unsupervised object discovery. However, it tends to learn highly entangled representations, unlike ours, since it can only implicitly encourage disentanglement by adjusting the latent dimension. The SRN (Huang et al., 2020) was published concurrently with Slot Attention and appears to offer a similar mechanism for mapping a single distributed representation to a set representation.

3. EfficientMORL

Our goal is to infer a set $\mathbf{z} := \{z_1, \ldots, z_K\}$, $z_k \in \mathbb{R}^D$ of object-centric representations from a color image $x \in \mathbb{R}^{H \times W \times 3}$. We assume that $\mathbf{z}$ generates the image x and that each element of $\mathbf{z}$ corresponds to a single object in the scene. To solve the inverse problem of obtaining $\mathbf{z}$ from x, we could compute the posterior distribution $p(\mathbf{z} \mid x)$. Bayes rule tells us that we also need the joint distribution $p(x, \mathbf{z}) = p(x \mid \mathbf{z})p(\mathbf{z})$, which describes the image generation process. Since the latent dimension D can be

[1] See Appendix A.3 of Greff et al. (2019) and our Appendix B for a discussion on unordered vs ordered representations.



(a) Stage 1 (b) Stage 2 (c)

Figure 1. **Two-stage inference** a) Bottom-up inference over L stochastic layers is used to iteratively extract K symmetric and disentangled representations from an image x. Disentanglement is achieved via hierarchical prior regularization. b) A *lightweight* refinement network f_ϕ refines the Stage 1 posterior $\boldsymbol{\lambda}^{(L,0)}$ for I steps. Although f_ϕ has low-dimensional inputs and outputs to make it efficient, the decoding step for computing the loss $\mathcal{L}^{(L,i)}$ is still costly. c) Since EfficientMORL learns to use refinement to avoid getting stuck in poor local minima during the early phase of training, we find that we can speed up training by decreasing I once Stage 1 starts converging. After training, the refinement stage can be removed at a small drop in decomposition performance for faster test time inference.

large (e.g., 64), computing $p(\mathbf{z} \mid x)$ requires solving an intractable integral. Instead we use amortized variational inference (Kingma & Welling, 2014) and compute an approximate variational posterior $q(\mathbf{z} \mid x)$. Like IODINE, we make an independence assumption among the K latent variables so that the variational posterior and prior are defined as symmetric products of K multivariate Gaussians with diagonal covariance. Neural nets with weights θ are used to obtain the parameters of the generative distribution; for the variational distributions, we use networks with weights ϕ. Pseudocode for the inference algorithm is provided (Algorithm 1) which we will reference by line number.

3.1. Stage 1: The hierarchical variational autoencoder

Bottom-up inference The variational posterior is designed to perform an L-step iterative assignment of pixels to K latents in a single pass. As such, we split $\mathbf{z}$ across L stochastic layers, creating a bottom-up pathway (Figure 2). The prior $p_\theta(\mathbf{z}^{1:L})$ regularizes the posterior $q_\phi(\mathbf{z}^{1:L} \mid x)$ at intermediate layers, disentangling the scene representation as a result. The multi-layer variational posterior distribution is given by

$$q_\phi(z_k^0) q_\phi(\mathbf{z}^{1:L} \mid x, \mathbf{z}^0) = q_\phi(z_k^0) \prod_{k=1}^{K} q_\phi(z_k^{1:L} \mid x, z_k^0) \quad (1)$$
$$= q_\phi(z_k^0) \prod_{k=1}^{K} \prod_{l=1}^{L} q_\phi(z_k^l \mid x, z_k^{l-1}),$$

where z_k^0 is Gaussian with learned mean $\mu \in \mathbb{R}^D$ and variance $\sigma^2 \in \mathbb{R}^D$ has been introduced as a zeroth layer. Randomness is introduced by sampling K times from $q_\phi(z^0)$ to help initially break symmetry. *Conditional on each sampled z_k^0, the set of K marginal distributions $q_\phi(z_k^{1:L} \mid x, z_k^0)$ are equivariant with respect to permutations applied to their ordering*. Equivalently, shuffling the order of $\mathbf{z}^0$ will likewise shuffle the order of the set of K Gaussian posterior parameters at the L^{th} layer. Each layer must be designed to preserve this symmetry while mapping pixels to latents.

We achieve this by adapting the attention-based image-to-set mapping introduced by Slot Attention. In detail, the l^{th} stochastic layer uses scaled dot-product set attention to attend over $N = HW$ tokens derived from a flattened embedding of an image augmented with a positional encoding. The **key** k and **value** v are the embedded image, and the set-structured **query** q is the *stochastic* sample $\mathbf{z}^{l-1}$ from the previous layer's posterior. The query is used to output set-structured features $\boldsymbol{\Theta} \in \mathbb{R}^{K \times D}$ as a function of the attention $\boldsymbol{\alpha} \in [0, 1]^{K \times HW}$ (Lines 2-11). Two GRUs (Cho et al., 2014), each with hidden dimension D, fuse the previous layer posterior's mean and variance with $\boldsymbol{\Theta}$ before an additive update to the predicted mean and variance with separate MLPs (Lines 14-15). We justify the introduction of a second GRU by noting in an ablation study that the model struggled to learn to map the shared feature $\boldsymbol{\Theta}$ to the posterior mean and variance using a single GRU with hidden dimension $2D$. For a similar reason the MLPs predicting the

Algorithm 1 Two-stage inference Linear attention maps k, q, v with D output units and LayerNorms (LN) are trainable. SP := Softplus. ϵ is for numerical stability. $N = HW$.

Input: image x
$x = \texttt{image\_encoder}(x)$
$x = \texttt{LayerNorm}(x + \texttt{pos\_encoding}(x))$
/* Stage 1: Bottom-up inference */
$\mathbf{z}^0 \sim \mathcal{N}(\mu^0, (\sigma^0 I)^2)$
$\boldsymbol{\lambda}^0 = (\mu^0, \sigma^0)$
for stochastic layer $l = 1 \dots L$ **do**
  $\mathbf{z}^{l-1} = \texttt{LayerNorm}(\mathbf{z}^{l-1})$
  $\boldsymbol{\alpha} = \texttt{softmax}_K\left(\frac{1}{\sqrt{D}} k(x) q(\mathbf{z}^{l-1})\right)$
  $\boldsymbol{\alpha} = (\boldsymbol{\alpha} + \epsilon) / \sum_N (\boldsymbol{\alpha} + \epsilon)$
  $\boldsymbol{\Theta} = \sum_N \boldsymbol{\alpha} \cdot v(x)$
  $\boldsymbol{\lambda}^l = \texttt{DualGRU}([\boldsymbol{\Theta}; \boldsymbol{\Theta}], \boldsymbol{\lambda}^{l-1})$
  $(\boldsymbol{\mu}^l, \boldsymbol{\sigma}^l) = \boldsymbol{\lambda}^l$
  $\boldsymbol{\mu}^l \mathrel{+}= \texttt{MLP}(\texttt{LayerNorm}(\boldsymbol{\mu}^l))$
  $\boldsymbol{\sigma}^l \mathrel{+}= \texttt{SP}(\texttt{MLP}(\texttt{LayerNorm}(\boldsymbol{\sigma}^l)))$
  $\mathbf{z}^l \sim \mathcal{N}(\boldsymbol{\mu}^l, (\boldsymbol{\sigma}^l I)^2)$ /* $q(\mathbf{z}^l \mid x, \mathbf{z}^{l-1})$ */
end for
$\mathcal{L}^{(L,0)} = \mathcal{L}_{\text{NLL}} + D_{KL}\left(q(\mathbf{z}^{1:L} \mid x) \parallel p(\mathbf{z}^{1:L})\right)$
/* Stage 2: Iterative refinement */
$\boldsymbol{\lambda}^{(L,0)} = (\boldsymbol{\mu}^L, \boldsymbol{\sigma}^L)$
for refinement iter $i = 1 \dots I$ **do**
  $\nabla_{\boldsymbol{\lambda}} \bar{\mathcal{L}}^{(L,i-1)} = \texttt{LN}(\texttt{stop\_grad}(\nabla_{\boldsymbol{\lambda}} \mathcal{L}^{(L,i-1)}))$
  $\boldsymbol{\lambda}^{(L,i)} = \boldsymbol{\lambda}^{(L,i-1)} + f(\boldsymbol{\lambda}^{(L,i-1)}, \nabla_{\boldsymbol{\lambda}} \bar{\mathcal{L}}^{(L,i-1)})$
  $\mathbf{z}^{(L,i)} \sim q(\mathbf{z}; \boldsymbol{\lambda}^{(L,i)})$
  $\boldsymbol{\pi}, \mathbf{y} = \texttt{decoder}(\mathbf{z}^{(L,i)})$
  $\mathcal{L}^{(L,i)} = \mathcal{L}^{(L,i)}_{\text{NLL}} + D_{KL}\left(q(\mathbf{z}; \boldsymbol{\lambda}^{(L,i)}) \parallel p(\mathbf{z}^L \mid \mathbf{z}^{L-1})\right)$
end for
return $\boldsymbol{\lambda}^{(L,I)}$ /* The image representation */

Gaussian parameters are not shared, which is standard for VAEs. In practice, we implement the two GRUs as a single GRU with *block-diagonal* weight matrices that takes in the concatenated features $[\boldsymbol{\Theta}; \boldsymbol{\Theta}] \in \mathbb{R}^{K \times 2D}$ to parallelize the computation (DualGRU in Algorithm 1 and Figure 1). Finally, we sample from the posterior which provides the next query (Line 16). See Appendix A for verification that the permutation equivariance of the K marginal distributions of the posterior are preserved during inference and Appendix F for more details on the DualGRU implementation.

Hierarchical prior Multi-layer priors $p_\theta(\mathbf{z}^{1:L})$ for mean field HVAEs are often designed so that correlations among the latent variables can be captured to facilitate learning highly-expressive priors and posteriors. Recently, a particular approach to accomplish this—top-down priors with bidirectional inference—has achieved impressive results for unconditional image generation (Sønderby et al., 2016; Kingma et al., 2016; Vahdat & Kautz, 2020; Child, 2020).

However, we found it non-trivial to adapt such top-down priors for our setting. First, it is not obvious how to re-use our unique bottom-up inference network within a top-down prior to combine their pathways (Sønderby et al., 2016). Second, it is also unclear how to design a suitable prior that

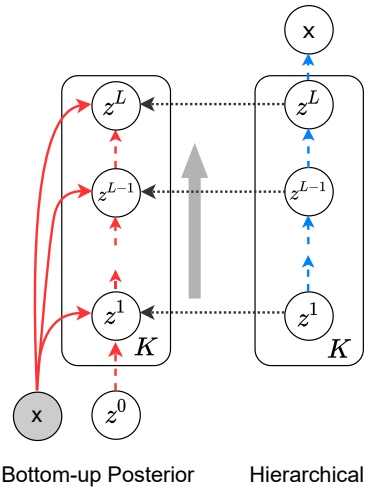


Figure 2. The HVAE's graphical model. Dashed arrows show reparameterized sampling, dotted arrows show prior regularization, and solid arrows show deterministic connections. White circles are random variables and gray are inputs. To try to mitigate posterior collapse, we also consider a variant of the prior where the arrows between $\mathbf{z}^1, \dots, \mathbf{z}^L$ are reversed (see Figure 12).

uses global information to generate coherent multi-object scenes without removing the equivariance property. We leave investigating these complex priors for future work and instead simply take the image likelihood and multi-layer prior to be

$$p_\theta(x, \mathbf{z}^{1:L}) = p_\theta(x \mid \mathbf{z}^L) p_\theta(\mathbf{z}^{1:L}) \tag{2}$$
$$= p_\theta(x \mid \mathbf{z}^L) \prod_{k=1}^{K} p(z_k^1) \prod_{l=2}^{L} p_\theta(z_k^l \mid z_k^{l-1}).$$

See Figure 2 for the graphical model. The bottom-level prior $p(z_k^1)$ is a standard Gaussian distribution and we implement each layer $p_\theta(z_k^l \mid z_k^{l-1})$ as an MLP with one hidden layer followed by two linear layers for the mean and variance respectively. Note that our simple image likelihood distribution only depends on $\mathbf{z}^L$.

Image likelihoods We implement two image likelihood models. For both, we use a spatial broadcast decoder (Watters et al., 2019b) to map samples from the posterior or prior to K assignment masks $\boldsymbol{\pi}$ normalized by softmax and K RGB images $\mathbf{y}$. The first model (*Gaussian*) uses $\boldsymbol{\pi}$ to compute a weighted sum over K predicted RGB values for each pixel, then places a Gaussian over the weighted sum with fixed variance σ^2. The second is a pixel-wise *Mixture of Gaussians* where $\boldsymbol{\pi}$ weighs each Gaussian in the sum (Burgess et al., 2019; Greff et al., 2019; Engelcke et al., 2020). See Appendix F for formal descriptions. We found the discussion in the literature lacking on the different inductive biases imbued by each image model so we explored this empirically (Section 4.2). We observed that

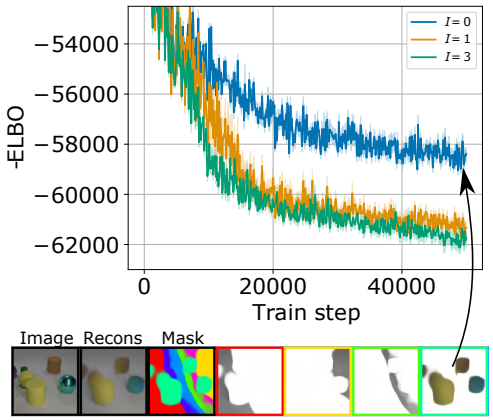


Figure 3. **Without iterative amortized inference the HVAE achieves a poor ELBO.** Training loss using $I = 0, 1, 3$ refinement steps. Results for each I are averaged across 10 random seeds with 95% C.I. shown. Without refinement, the HVAE consistently gets stuck in poor local minima early on during training and does not recover. For example, all foreground objects get placed in a single component. With just one refinement step we find that the HVAE can better avoid such minima.

the Gaussian model tends to split the background across the K components whereas the mixture model consistently places the background into a single component.

3.2. Stage 2: Iterative amortized inference

Given the bottom-up posterior, prior, and decoder, it is possible to train the HVAE with just the single-sample approximation of the Evidence Lower Bound (ELBO) without IAI. We define the negative log-likelihood as

$$\mathcal{L}_{\text{NLL}} = -\mathbb{E}_{\mathbf{z}^L \sim q_\phi(\mathbf{z}^L \mid x)}\big[\log p_\theta(x \mid \mathbf{z}^L)\big], \tag{3}$$

where the expectation is computed using ancestral sampling. The KL divergence factorized by layer between the prior and posterior is

$$\begin{aligned} &D_{KL}\big(q_\phi(\mathbf{z}^{1:L} \mid x) \parallel p_\theta(\mathbf{z}^{1:L})\big) \\ &= \mathbb{E}\big[D_{KL}\big(q_\phi(\mathbf{z}^1 \mid x, \mathbf{z}^0) \parallel p(\mathbf{z}^1)\big)\big] \\ &\quad + \sum_{l=2}^{L} \mathbb{E}\big[D_{KL}\big(q_\phi(\mathbf{z}^l \mid x, \mathbf{z}^{l-1}) \parallel p_\theta(\mathbf{z}^l \mid \mathbf{z}^{l-1})\big)\big], \end{aligned} \tag{4}$$

then we can minimize the negative ELBO

$$\mathcal{L}^{(L,0)} = \mathcal{L}_{\text{NLL}} + D_{KL}\big(q_\phi(\mathbf{z}^{1:L} \mid x) \parallel p_\theta(\mathbf{z}^{1:L})\big). \tag{5}$$

Finding good local minima is challenging because the posterior is highly multi-modal due to the symmetric latent structure. Penalizing the model for learning entangled representations through prior regularization makes optimization more challenging as well. We hypothesize that IAI can address this since it uses top-down feedback—the negative ELBO—to refine the posterior. In Figure 3, we show that without IAI the HVAE consistently converges to a poor ELBO. It also learns the optimal step size for the refinement update making it highly effective. *The question remains of how to use it without incurring a large increase in computation since evaluating the negative ELBO requires decoding K image-sized masks and RGB components.*

Stage 2 efficiency As shown in Figure 1 we use a two-stage approach to inference. First we use ancestral sampling and the HVAE's bottom-up inference pathway to compute $\boldsymbol{\lambda}^L := \{\boldsymbol{\mu}^L, \boldsymbol{\sigma}^L\}$—the parameters of the marginal distribution $q_\phi(\mathbf{z}^L \mid x)$. The goal of the second stage is to use a *lightweight* refinement network f_ϕ for I steps of IAI (Lines 21-27) to refine $\boldsymbol{\lambda}^L$ with top-down feedback. As Figure 1 shows, the HVAE primarily uses IAI to make small refinements to $\boldsymbol{\lambda}^L$, particularly during the early training stages. This means that with only a small number of steps we can see a large improvement in final performance (Figure 3) and suggests employing training strategies such as reducing the number of steps after the HVAE starts to converge to reduce overall training time (Section 4.1). Unlike IODINE, our refinement network f_ϕ does not take in image-sized inputs, which greatly reduces the number of model parameters and makes refinement even faster during training.

Stage 2 refinement network At refinement step i, we use a simple network f_ϕ that encodes the concatenated Gaussian parameters $\boldsymbol{\lambda}^{(L,i-1)}$ and the gradient of the refinement loss $\nabla_{\boldsymbol{\lambda}}\mathcal{L}^{(L,i-1)}$. For the latter, we use layer normalization (Ba et al., 2016) and stop gradients from passing through (Line 22) (Marino et al., 2018). The network f_ϕ is an MLP that encodes the input, a GRU with hidden dimension D, and two linear layers to compute an additive update:

$$\delta\boldsymbol{\lambda}^{(L,i-1)} = f_\phi(\boldsymbol{\lambda}^{(L,i-1)}, \nabla_{\boldsymbol{\lambda}}\mathcal{L}^{(L,i-1)}) \tag{6}$$

$$\boldsymbol{\lambda}^{(L,i)} = \boldsymbol{\lambda}^{(L,i-1)} + \delta\boldsymbol{\lambda}^{(L,i-1)}. \tag{7}$$

The refinement loss $\mathcal{L}^{(L,i)}$ (Line 26) is the negative ELBO, defined as the KL divergence between the refined posterior and $p_\theta(\mathbf{z}^L \mid \mathbf{z}^{L-1})$ plus the negative log-likelihood.

3.3. Training

Training loss The loss that gets minimized is

$$\mathcal{L} = \mathcal{L}^{(L,0)} + \sum_{i=1}^{I} \frac{I-(i-1)}{I+1}\mathcal{L}^{(L,i)}, \tag{8}$$

where the discount factor $\frac{I-(i-1)}{I+1}$ emphasizes the loss near $i = 0$ to place more weight on the encoder than on the refinement network; this has the opposite effect of the discount factor used by IODINE which places more weight on later refinement terms (large i).

Posterior collapse HVAEs can suffer from posterior collapse (Sønderby et al., 2016), which is when each layer of

the approximate posterior collapses to the prior early on during training and never recovers. We applied mitigation strategies for this in three parts of the framework: the graphical model of the hierarchical prior, the prior used in the refinement loss KL term, and the training objective.

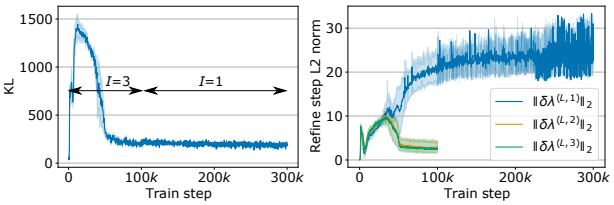
We created a variant of the hierarchical prior where the arrows between $\mathbf{z}^1, \ldots, \mathbf{z}^L$ are reversed. This *reversed prior* now has $p(\mathbf{z}^L)$ as the standard Gaussian and each intermediate layer is given by $p_\theta(\mathbf{z}^l \mid \mathbf{z}^{l+1})$. Intuitively, penalizing the posterior at layer L by its deviation from a standard Gaussian imposes a stricter constraint on the posterior's expressiveness. By exploring this alternative prior, we can determine whether a better match between the flexibility of the posterior and our simple prior helps address collapse. We then considered replacing the prior in the refinement loss KL term with $p_\theta(\mathbf{z}^1 \mid \mathbf{z}^2)$ (e.g., *reversed prior++*). We hypothesized that implicitly pushing the posteriors at lower layers to match the posterior at layer L during refinement should help keep them from collapsing to the prior. See Figure 12 for a comparison of the variants. The loss (Equation 8) was modified to use GECO (Rezende & Viola, 2018). GECO reformulates the ELBO to initially allow the KL to grow large so that a predefined reconstruction threshold can first be attained. We chose GECO since tuning its hyperparameters was easier than for deterministic warm-up (Sønderby et al., 2016).

Our ablation studies analyzing the contribution of each strategy can be found in Appendix D. We note that GECO was needed for CLEVR6 and Tetrominoes but not on Multi-dSprites. We use the best-performing variant, *reversed prior++*, for the experiments in Section 4.

4. Experiments

The evaluation of EfficientMORL is organized as follows. We first analyze the refinement updates to suggest a principled justification for our training strategy (Section 4.1). Then, we evaluate object decomposition (Section 4.2) and disentanglement performance (Section 4.3). Finally, we compare run time and memory costs with competing models (Section 4.4). Additional qualitative results and the ablation studies on the hierarchical prior, refinement steps I, and the DualGRU are in Appendix D.

Datasets We use the Multi-Object Dataset (Kabra et al., 2019) for all experiments. This benchmark has two sprites-based environments (Tetrominoes and Multi-dSprites) and a synthetic 3D environment (CLEVR) with ground truth object segmentation masks. We follow the same training and evaluation protocol as Greff et al. (2019); Locatello et al. (2020). For both sprites datasets, we split the data by using the first 60K samples for training and then hold out the next 320 images for testing. We filter all CLEVR images with less than seven objects to create a 50K training set (CLEVR6). For testing, we also use 320 held-out images. To evaluate generalization we use a test set of 320 images containing $7-10$ objects (CLEVR10). Note that after center cropping the CLEVR images, we resize them to 96×96 instead of 128×128, unlike Locatello et al. (2020); Greff et al. (2019), to make replicating our CLEVR experiments across multiple random seeds practical.



(a) (b)

Figure 4. **The L2 norm of the refinement updates are good indicators for when we can reduce I, speeding up training**. a) Training curves of the final KL across five CLEVR6 runs. The early spike in KL is due to GECO. b) As the KL starts to converge around 100K steps, the L2 norm of the updates for $I > 1$ also decreases. This suggests they are no longer contributing much to the final posterior.

Hyperparameters All models use the Adam (Kingma & Ba, 2015) optimizer, a learning rate of 4e-4 with warm-up and exponential decay, gradient norm clipping to 5.0, and a mini-batch size of 32. Following (Greff et al., 2019; Locatello et al., 2020) we use $K = 7$ components for CLEVR6, $K = 6$ for Multi-dSprites, and $K = 4$ for Tetrominoes (note that K can be set to any number for any given image if desired). Complete model architecture, optimizer, and evaluation details are provided in Appendix F.

4.1. IAI steps analysis

To better understand the role of IAI in EfficientMORL and to justify decreasing I during training to reduce training time, we analyze the KL curves from five training runs on CLEVR6 and plot the L2 norm of each refinement update $\delta\boldsymbol{\lambda}^{(L,i)}$ (Figure 4). Across runs, we observe that as the KL starts to converge, the L2 norm of updates for steps $I > 1$ became small. Our interpretation is that at this point, the bottom-up posterior from stage one is sufficiently good such that more than one refinement step is not needed. When we decrease I from three to one during training, we notice a slight drop in reconstruction quality that the model quickly recovers from. We attribute the continued increase in L2 norm for $\delta\boldsymbol{\lambda}^{(L,1)}$ to the observation that a single refinement step at test time has a larger effect on the KL than on the reconstruction quality (Figure 7d, Figure 14).

In our experiments we did not decrease I to one from three on Tetrominoes due to the small image size and fast convergence in $200K$ steps. On CLEVR6 and Multi-dSprites we

Table 1. Multi-object Benchmark results. Adjusted Rand Index (ARI) scores (mean ± stddev for five seeds). **We achieve comparable performance to state-of-the-art baselines.** We outperform IODINE on Multi-dSprites. We replicated Slot Attention's CLEVR6 results over five random seeds (**) but one run failed—without it, the ARI improves from 93.3 to 98.3. Tetrominoes (*) was reported with only 4 seeds (Locatello et al., 2020).

	CLEVR6	Multi-dSprites	Tetrominoes
Slot Attention	98.8 ± 0.3	$\mathbf{91.3 \pm 0.3}$	$99.5 \pm 0.2^*$
Slot Attention (**)	93.3 ± 11.1	—	—
IODINE	$\mathbf{98.8 \pm 0.0}$	76.7 ± 5.6	$\mathbf{99.2 \pm 0.4}$
MONet	96.2 ± 0.6	90.4 ± 0.8	—
Slot MLP	60.4 ± 6.6	60.3 ± 1.8	25.1 ± 34.3
EfficientMORL	96.2 ± 1.6	$\mathbf{91.2 \pm 0.4}$	98.2 ± 1.8

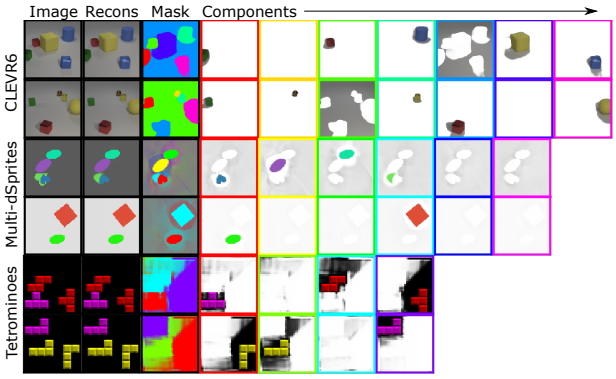


Figure 5. Visualization of scene decompositions for CLEVR6 (top), Multi-dSprites (middle) and Tetrominoes (bottom). The Mixture of Gaussians model used for CLEVR6 places the background into a single component, whereas the Gaussian model splits simple backgrounds across all components.

train for 100K steps with $I = 3$, then train for another 200K steps with I decreased to one. One step is used at test time for evaluating these two environments.

4.2. Object decomposition

Baselines and metrics The baselines are Slot Attention (Locatello et al., 2020), IODINE (Greff et al., 2019), MONet (Burgess et al., 2019), and the Slot MLP baseline from (Locatello et al., 2020) which maps an embedded image to an *ordered* set of K slots. To measure decomposition quality we use the adjusted rand index (ARI) (Rand, 1971; Hubert & Arabie, 1985) and do not include the background mask in the ARI computation following standard practice for this benchmark. We also compute pixel mean squared error (MSE), which takes into account the background.

Analysis on the number of stochastic layers We vary the number of stochastic layers L in the prior and posterior (Figure 8) during training and measure the ARI. Best results are obtained with $L = 3$, which we use for all experiments. Note that when $L = 0$, EfficientMORL extracts the scene representation with only I refinement steps.

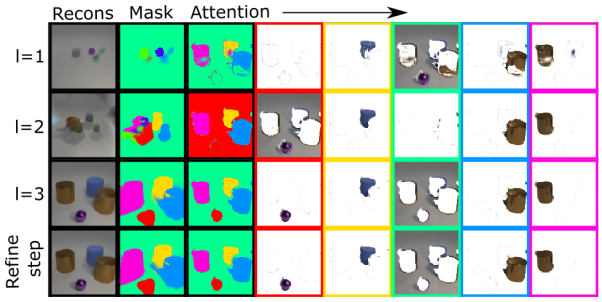
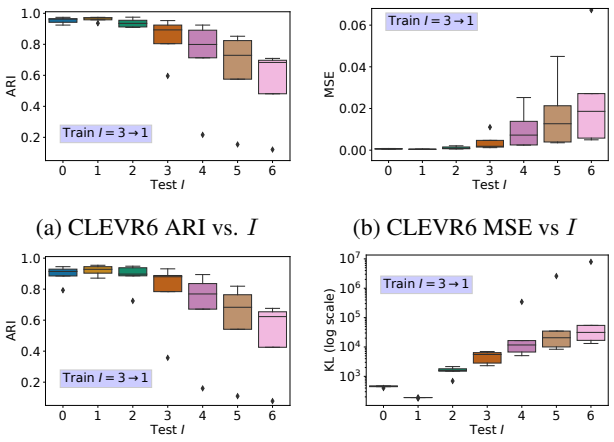


Figure 6. The first two columns show reconstructions/masks from the posteriors, column three shows the hard assignment of the softmax attention $\boldsymbol{\alpha}$ (Algorithm 1, Line 9), and columns 4-8 show the same attention drawn over the input image.



(a) CLEVR6 ARI vs. I (b) CLEVR6 MSE vs I

(c) CLEVR10 ARI vs. I (d) CLEVR6 KL vs. Test I

Figure 7. a-b) **Refinement marginally improves segmentation and reconstruction at test time**. We show ARI and MSE when varying the number of test refinement steps I on CLEVR6. c) **The model generalizes to larger numbers of objects at test time**. We increase components K to 11 for CLEVR10 (7-10 objects). d) Test time refinement impacts the KL more strongly than segmentation as evidenced by the increased KL at $I = 0$.

Main results ARI scores are in Table 1 and qualitative examples for each environment are in Figure 5. Overall, EfficientMORL's decomposition performance is comparable to Slot Attention, MONet, and IODINE on all three environments. EfficientMORL uses the Gaussian image likelihood for the two sprites environments because it quickly and reliably converges, whereas the Mixture of Gaussians had difficulty discovering the sprites. The Mixture of Gaussians is used for CLEVR6 and biases the model towards assigning the background to a single component in each training run, which may be desirable. Since the Gaussian likelihood biases the model to split the background across all components, it appears to be better suited to handle simple single-color backgrounds (note that IODINE uses the mixture model on all environments, which may explain its

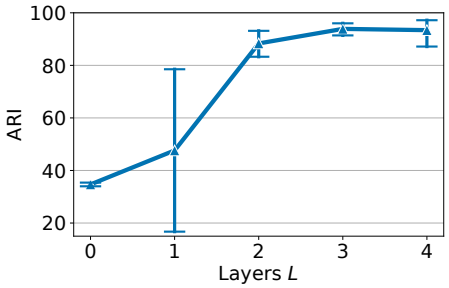


Figure 8. Sensitivity analysis on the number of layers L in the HVAE. Results are averaged over five CLEVR6 training runs. Best performance is achieved at $L = 3$.

lower scores on Multi-dSprites).

Intermediate posteriors We reconstruct samples drawn from the intermediate posteriors and visualize the layer-wise masks and attention over the input image in Figure 6. The single refinement step applied to $\boldsymbol{\lambda}^{(3,0)}$ imperceptibly changes the reconstructed image and segmentation. The intermediate posterior reconstructions suggest they have not collapsed to the prior and that the top-level posterior fits an expressive non-Gaussian distribution.

Varying test time IAI steps Our earlier analysis (Figure 3) demonstrated that during *training*, using $I > 0$ stabilizes convergence to achieve a better ELBO, with $I = 3$ slightly outperforming $I = 1$. Notably, *zero* IAI steps at test time can achieve 99.1% of the refined ARI and MSE (Figures 7a, 7b). We see a larger gap in the KL between zero and one step (Figures 7d, 14f, 14e), which suggests that refinement plays a larger role in achieving this aspect of the extracted high-quality representation at test time. The decrease in test ARI as I is increased to six is due to the refinement GRU ignoring sequential information after reducing I to one during training. If I is held at three, we find this is no longer the case (Figure 14 in the appendix).

Systematic generalization We evaluate whether EfficientMORL generalizes when seeing more objects at test time by increasing K to 11 and varying I on CLEVR10. As expected due to the equivariance property, we only see a slight drop in ARI (Figures 7c).

4.3. Disentanglement

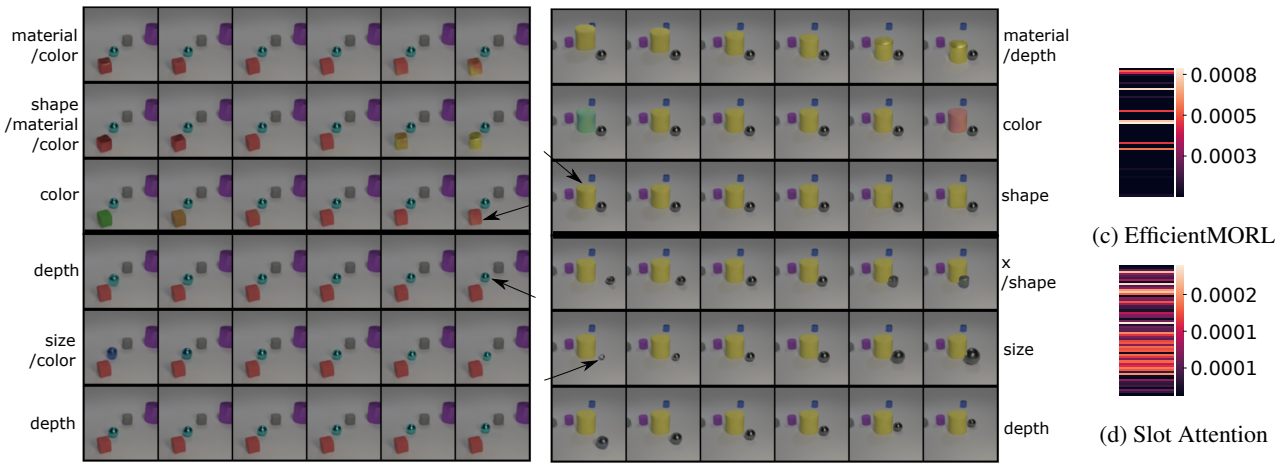
Baselines and metrics The goal of this experiment is to compare the disentanglement quality of EfficientMORL's representations against the current state-of-the-art *efficient* and *equivariant* model Slot Attention on CLEVR6. We recall that Slot Attention is a deterministic autoencoder without regularization, so we expect that it learns entangled scene representations. Quantifying disentanglement for multi-object scenes is challenging since widely accepted metrics like DCI (Eastwood & Williams, 2018) require access to the oracle matching—which is unknown— between the K inferred representations and the set of ground truth factors. Apart from heuristically estimating DCI scores (see Appendix E.2 for details), we visualize latent dimension interpolations and compute the *activeness* (Peebles et al., 2020) of the latent dimensions. Activeness is the mean image variance when changing the i^{th} dimension of one uniformly sampled latent z_k across 100 test images. Intuitively, a disentangled model should have many deactivated latent dimensions that do not change the image when perturbed. While a similar comparison against other models like IODINE and GENESIS would be interesting, their authors did not provide disentanglement scores such as DCI, and we had just enough resources to train Slot Attention with multiple random seeds. Although we leave a broader comparison future work, see Appendix B.1 for a specific case study comparing GENESIS and our model.

Results Figure 9b contains the DCI scores and examples of varying latent dimensions for two different objects in a single scene. Our method's better disentanglement is verified by much higher DCI scores. Many of Slot Attention's latent dimensions change multiple factors (for example, one changes the shape, material, and color of a single object), while we observe this in ours for only a small number. Moreover, by examining Slot Attention's activeness heatmap (Figure 9d), we see that the majority of latent dimensions contribute to the mean variance, whereas the majority of latent dimensions in ours are deactivated (Figure 9c).

4.4. Efficiency

Setup We measure the time taken for the forward and backward passes on 1 2080 Ti GPU with a mini-batch size of 4, images sizes 64×64, 96×96, and 128×128, and $I \in \{0, 1, 3\}$. For comparison we use our own implementation of IODINE ($K = 7$ and 5 inference steps), GENESIS with $K = 7$, and Slot Attention with $K = 7$ and $L = 3$ in PyTorch (Paszke et al., 2019). We checked implementation details against the official releases to ensure a fair comparison. Also, to compare memory consumption we record the largest mini-batch size able to fit on 1 12 GB 2080 Ti GPU.

Results Figure 10 shows the key results with the remainder in the appendix. We show that our model with $I = 0$ has $10\times$ faster forward pass (e.g., at test time) and a $6\times$ faster forward + backward pass than IODINE. In general, our model has better memory consumption than IODINE up until $I = 3$. However, if we replace the decoder with Slot Attention's deconvolutional decoder, we can double the maximum mini-batch size that fits on a single GPU (E-X-S in Figure 10). The impact of increased run time and memory between $I = 1$ and $I = 3$ on wall clock training time is mostly offset by decreasing I after the model starts



(a) Slot Attention (DCI: 0.46 / 0.38 / 0.34)

(b) EfficientMORL (DCI: **0.63 / 0.63 / 0.46**)

(c) EfficientMORL

(d) Slot Attention

Figure 9. **EfficientMORL outperforms Slot Attention at disentangling object attributes.** (a-b) We traverse latent dimensions for two objects in one scene (top three, bottom three rows). Rows are labeled by the attributes we observe to change, with entangled dimensions annotated by multiple attributes. DCI scores are in the captions (higher is better). The latent dimensions of EfficientMORL's representation ($\mathbf{z}^{(l=3,i=1)}$) have less correlation and redundancy (multiple dimensions controlling the same attribute). (c-d) **EfficientMORL has fewer *active* latent dimensions than Slot Attention**. Perturbing most of the 64 dimensions has no effect on the reconstructed image.

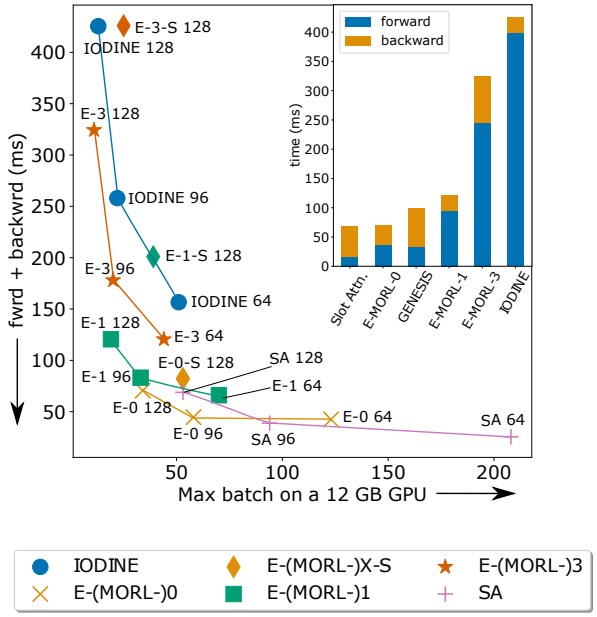


Figure 10. Lines connect results for each image size. E-X-S is ours with the memory-efficient decoder, SA := Slot Attention. Across all settings except E-3 we have better memory consumption than IODINE. Top right) **With $I = 0$ we have a $10\times$ faster forward pass than IODINE and with $I \leq 1$ we are comparable to Slot Attention/GENESIS**. Times are for 128×128 images.

to converge early on, a unique aspect of our model.

The wall clock time to train EfficientMORL on CLEVR6 with a mini-batch size of 32 split evenly across 8 GPUs is about 17 hours. Slot Attention reports a wall clock training time of 24 hours on CLEVR6, but they train using the full 128×128 images. Controlling for that, our model takes slightly longer than Slot Attention to train on CLEVR. IODINE reportedly trains for 1M steps, which would take $10\times$ longer than our model took to converge.

5. Discussion

We introduced EfficientMORL, a generative modeling framework for learning object-centric representations that are symmetric and disentangled without the intense computation typically required when using iterative amortized inference and without sacrificing important aspects of the scene representation. EfficientMORL's relatively fast training times and test time inference make it useful for exploring topics such as object-centric representation learning for video. While these models still require some engineering to work well on new environments, further study will likely lead to more general approaches.

Acknowledgements

We thank Hadi Abdullah for providing valuable comments and suggested revisions on an early draft. This work is supported in part by NSF Awards 1446813 and 1922782. Patrick Emami is also supported in part by a FEF McKnight fellowship and a UF CISE graduate research fellowship. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. doi: 10.1109/TPAMI.2012.120.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bear, D., Fan, C., Mrowca, D., Li, Y., Alter, S., Nayebi, A., Schwartz, J., Fei-Fei, L., Wu, J., Tenenbaum, J., and Yamins, D. L. Learning physical graph representations from visual scenes. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, 2020.

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

Child, R. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-4012.

Clevert, D., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). In Bengio, Y. and LeCun, Y. (eds.), *International Conference on Learning Representations*. OpenReview.net, 2016.

Crawford, E. and Pineau, J. Spatially invariant unsupervised object detection with convolutional neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 3412–3420. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33013412.

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

Eastwood, C. and Williams, C. K. I. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*. OpenReview.net, 2018.

Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. GENESIS: generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*. OpenReview.net, 2020.

Engelcke, M., Jones, O. P., and Posner, I. Genesis-v2: Inferring unordered object representations without iterative refinement. *arXiv preprint arXiv:2104.09958*, 2021.

Eslami, S. M. A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., and Hinton, G. E. Attend, infer, repeat: Fast scene understanding with generative models. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, pp. 3225–3233, 2016.

Greff, K., Srivastava, R. K., and Schmidhuber, J. Binding via reconstruction clustering. *arXiv preprint arXiv:1511.06418*, 2015.

Greff, K., Rasmus, A., Berglund, M., Hao, T. H., Valpola, H., and Schmidhuber, J. Tagger: Deep unsupervised perceptual grouping. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, pp. 4484–4492, 2016.

Greff, K., van Steenkiste, S., and Schmidhuber, J. Neural expectation maximization. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, pp. 6691–6701, 2017.

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. Multi-object representation learning with iterative variational inference. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2424–2433. PMLR, 2019.

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

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 2980–2988. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.322.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained

variational framework. In *International Conference on Learning Representations*. OpenReview.net, 2017.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Hinton, G. E., McClelland, J. L., and Rumelhart, D. E. *Distributed Representations*, pp. 77–109. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X.

Huang, Q., He, H., Singh, A., Zhang, Y., Lim, S., and Benson, A. R. Better set representations for relational reasoning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, 2020.

Hubert, L. and Arabie, P. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Jiang, J. and Ahn, S. Generative neurosymbolic machines. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, 2020.

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

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *International Conference on Learning Representations*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *International Conference on Learning Representations*, 2014.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934*, 2016.

Li, N., Eastwood, C., and Fisher, R. B. Learning object-centric representations of multi-object scenes from multiple views. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, 2020.

Lin, Z., Wu, Y., Peri, S. V., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. SPACE: unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*. OpenReview.net, 2020.

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4114–4124. PMLR, 2019.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11525–11538. Curran Associates, Inc., 2020.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*. OpenReview.net, 2019.

Marino, J., Yue, Y., and Mandt, S. Iterative amortized inference. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3400–3409. PMLR, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035. Curran Associates, Inc., 2019.

Peebles, W., Peebles, J., Zhu, J.-Y., Efros, A., and Torralba, A. The hessian penalty: A weak prior for unsupervised disentanglement. *arXiv preprint arXiv:2008.10599*, 2020.

Rand, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

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

Schmidhuber, J. Learning factorial codes by predictability minimization. *Neural computation*, 4(6):863–879, 1992.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder variational autoencoders. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, pp. 3738–3746, 2016.

Vahdat, A. and Kautz, J. Nvae: A deep hierarchical variational autoencoder. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19667–19679. Curran Associates, Inc., 2020.

van Steenkiste, S., Chang, M., Greff, K., and Schmidhuber, J. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *International Conference on Learning Representations*. OpenReview.net, 2018.

van Steenkiste, S., Kurach, K., Schmidhuber, J., and Gelly, S. Investigating object compositionality in generative adversarial networks. *Neural Networks*, 130:309–325, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.07.007.

Veerapaneni, R., Co-Reyes, J. D., Chang, M., Janner, M., Finn, C., Wu, J., Tenenbaum, J. B., and Levine, S. Entity abstraction in visual model-based reinforcement learning. *arXiv preprint arXiv:1910.12827*, 2019.

Watters, N., Matthey, L., Bosnjak, M., Burgess, C. P., and Lerchner, A. Cobra: Data-efficient model-based rl through unsupervised object discovery and curiosity-driven exploration. *arXiv preprint arXiv:1905.09275*, 2019a.

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

Yang, Y., Chen, Y., and Soatto, S. Learning to manipulate individual objects in an image. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 6557–6566. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00659.

Yuan, J., Li, B., and Xue, X. Spatial Mixture Models with Learnable Deep Priors for Perceptual Grouping. *arXiv preprint arXiv:1902.02502*, 2019.

Zhou, X., Wang, D., and Krähenbühl, P. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.