Unsupervised Learning of Compositional Scene Representations from Multiple Unspecified Viewpoints

Jinyang Yuan, Bin Li*, Xiangyang Xue*

Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University
{yuanjinyang, libin, xyxue}@fudan.edu.cn

Abstract

Visual scenes are extremely rich in diversity, not only because there are infinite combinations of objects and background, but also because the observations of the same scene may vary greatly with the change of viewpoints. When observing a visual scene that contains multiple objects from multiple viewpoints, humans are able to perceive the scene in a compositional way from each viewpoint, while achieving the so-called “object constancy” across different viewpoints, even though the exact viewpoints are untold. This ability is essential for humans to identify the same object while moving and to learn from vision efficiently. It is intriguing to design models that have the similar ability. In this paper, we consider a novel problem of learning compositional scene representations from multiple unspecified viewpoints without using any supervision, and propose a deep generative model which separates latent representations into a viewpoint-independent part and a viewpoint-dependent part to solve this problem. To infer latent representations, the information contained in different viewpoints is iteratively integrated by neural networks. Experiments on several specifically designed synthetic datasets have shown that the proposed method is able to effectively learn from multiple unspecified viewpoints.

Introduction

Vision is an important way for humans to acquire knowledge about the world. Due to the diverse combinations of objects and background that constitute visual scenes, it is hard to model the whole scene directly. In the process of learning from the world, humans are able to develop the concept of object (Johnson 2010), and is thus capable of perceiving visual scenes compositionally, which in turn leads to more efficient learning compared with perceiving the entire scene as a whole (Fodor and Pylyshyn 1988). Compositionality is one of the fundamental ingredients for building artificial intelligence systems that learn efficiently and effectively like humans (Lake et al. 2017). Therefore, instead of learning a single representation for the entire visual scene, it is desirable to build compositional scene representation models which learn object-centric representations (i.e., learn separate representations for different objects and background), so that the combinational property can be better captured.

*Corresponding author

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
(Greff et al. 2019), and Slot Attention (Locatello et al. 2020), however, are unsupervised methods that learn from only a single viewpoint. Only few methods, including MulMON (Li, Eastwood, and Fisher 2020) and ROOTS (Chen, Deng, and Ahn 2020), have considered the problem of learning from multiple viewpoints. These methods assume that the viewpoint annotations (under a certain global coordinate system) are given, and aim to learn viewpoint-independent object-centric representations conditioned on these annotations. Viewpoint annotations play fundamental roles in the object-centric representations from multiple viewpoints. These methods assume that the viewpoint annotations (under a certain global coordinate system) are given, and aim to learn viewpoint-independent object-centric representations conditioned on these annotations. Viewpoint annotations play fundamental roles in the initialization and updates of object-centric representations in MulMON, and in the computations of perspective projections in ROOTS. Therefore, without nontrivial modifications, existing methods cannot be applied to the novel problem of learning compositional scene representations from multiple unspecified viewpoints without any supervision.

The problem setting considered in this paper is very challenging, as the object-centric representations that are shared across viewpoints and the viewpoint representations that are shared across objects both need to be learned. More specifically, there are two major reasons. Firstly, the object constancy needs to be achieved without the guidance of viewpoint annotations, which are the only variable among images observed from different viewpoints and can be exploited to reduce the difficulty of learning the common factors. Secondly, the representations of images need to be disentangled into object-centric representations and viewpoint representations, even though there are infinitely many possible solutions, e.g., due to the change of global coordinate system.

In this paper, we propose a deep generative model called Object-Centric Learning with Object Constancy (OCLOC) to learn object-centric representations from multiple viewpoints without any supervision (including viewpoint annotations), under the assumptions that 1) objects in the visual scenes are static, and 2) different visual scenes may be observed from different sets of unordered viewpoints. The proposed method models viewpoint-independent attributes of objects/background (e.g., 3D shapes and appearances in the global coordinate system) and viewpoints with separate latent variables, and adopts an amortized variational inference method that iteratively updates parameters of the approximate posteriors by integrating information of different viewpoints with inference neural networks.

To the best of the authors’ knowledge, no existing object-centric learning method can learn from multiple unspecified viewpoints without viewpoint annotations. Thus, the proposed OCLOC cannot be directly compared with existing ones in the considered problem setting. Experiments on several specifically designed synthetic datasets have shown that OCLOC can effectively learn from multiple unspecified viewpoints without supervision, and competes with or slightly outperforms a state-of-the-art method that uses viewpoint annotations in the learning. Under an extreme condition that visual scenes are observed from one viewpoint, the proposed OCLOC is also comparable with the state-of-the-arts.

Related Work

Object-centric representations are compositional scene representations that treat object or background as the basic entity of the visual scene and represent different objects or background separately. In recent years, various methods have been proposed to learn object-centric representations in an unsupervised manner, or using only scene-level annotations. Based on whether learning from multiple viewpoints and whether considering the movements of objects, these methods can be roughly divided into three categories.

Single-Viewpoint Static Scenes: CST-VAE (Huang and Murphy 2016), AIR (Eslami et al. 2016), and MONet (Burgess et al. 2019) extract the representation of each object sequentially based on the attention mechanism. GMIOO (Yuan, Li, and Xue 2019a) initializes the representation of each object sequentially and iteratively updates the representations, both with attentions on objects. SPAIR (Crawford and Pineau 2019) and SPACE (Lin et al. 2020) generate object proposals with convolutional neural networks and are applicable to large visual scenes containing a relatively large number of objects. N-EM (Greff, van Steenkiste, and Schmidhuber 2017), LDP (Yuan, Li, and Xue 2019b), IO-DINE (Greff et al. 2019), Slot Attention (Locatello et al. 2020), and EfficientMORL (Emami et al. 2021) first initialize representations of all the objects, and then apply some kinds of competitions among objects to iteratively update the representations in parallel. GENESIS (Engelcke et al. 2020) and GNM (Jiang and Ahn 2020) consider the structure of visual scene in the generative models in order to generate more coherent samples. ADI (Yuan, Li, and Xue 2021) considers the acquisition and utilization of knowledge. These methods provide mechanisms to separate objects, and form the foundations of learning object-centric representations with the existences of object motions or from multiple viewpoints.

Multi-Viewpoint Static Scenes: MulMON (Li, Eastwood, and Fisher 2020) and ROOTS (Chen, Deng, and Ahn 2020) are two methods proposed to learn from static scenes from multiple viewpoints. MulMON extends the iterative amortized inference (Marino, Yue, and Mandt 2018) used in IODINE (Greff et al. 2019) to sequences of images observed from different viewpoints. Object-centric representations are first initialized based on the first pair of image and viewpoint annotation, and then iteratively refined by processing the rest pairs of data one by one. At each iteration, the previously estimated posteriors of latent variables are used as the current object-wise priors in order to guide the inference. ROOTS adopts the idea of using grid cells like SPAIR (Crawford and Pineau 2019) and SPACE (Lin et al. 2020), and generates object proposals in a bounded 3D region. The 3D center position of each object proposal is estimated and projected into different images with transformations that are computed based on the annotated viewpoints. After extracting crops of images corresponding to each object proposal, a type of GQN (Eslami et al. 2018) is applied to infer object-centric representations. As with our problem setting, different visual scenes are not assumed to be observed from the same set of viewpoints. However, because both methods heavily rely on the viewpoint annotations, they cannot be trivially applied to the fully-unsupervised scenario that the viewpoint annotations are unknown.

Dynamic Scenes: Inspired by the methods proposed for learning from single-viewpoint static scenes, several
methods, such as Relational N-EM (van Steenkiste et al. 2018), SQAIR (Kosiorek et al. 2018), R-SQAIR (Stanic and Schmidhuber 2019), TBA (He et al. 2019), SILO (Crawford and Pineau 2020), SCALOR (Jiang et al. 2020), OP3 (Veerapaneni et al. 2020), and PROVIDE (Zablotskaia et al. 2021), have been proposed for learning from video sequences. The difficulties of this problem setting include modeling object motions and relationships, as well as maintaining the identities of objects even if objects disappear and reappear after full occlusion (Weis et al. 2021). Although these methods are able to identify the same object across adjacent frames, they cannot be directly applied to the problem setting considered in this paper for two major reasons: 1) images observed from different viewpoints are assumed to be unordered, and the positions of the same object may differ significantly in different images; and 2) viewpoints are shared among objects in the same visual scene, while object motions in videos do not have such a property.

Generative Modeling

Visual scenes are assumed to be independent and identically distributed. For simplicity, the index of visual scene is omitted, and the procedure to generate images of a single visual scene is described. Let $M$ denote the number of images observed from different viewpoints (may vary in different visual scenes), $N$ and $C$ denote the respective numbers of pixels and channels in each image, and $K$ denote the maximum number of objects that may appear in the visual scene. The image of the $m$th viewpoint $x_m \in \mathbb{R}^{N \times C}$ is assumed to be generated via a pixel-wise weighted summation of $K + 1$ layers, with $K$ layers ($1 \leq k \leq K$) describing the objects and 1 layer ($k = 0$) describing the background. The pixel-wise weights $s_{m,0;K} \in [0, 1]^{(K+1) \times N}$ as well as the images of layers $a_{m,0;K} \in \mathbb{R}^{(K+1) \times N \times C}$ are computed based on latent variables. In the following, we first describe the latent variables and the likelihood function, and then express the generative model in the mathematical form.

Viewpoint-Independent Latent Variables

Viewpoint-independent latent variables are the ones that are shared across different viewpoints, and are introduced in the generative model to achieve object constancy. These latent variables include $z_{\text{attr}}, \rho, z_{\text{prs}}$.

- $z_{\text{attr}}^{m,k}$ characterize the viewpoint-independent attributes of objects ($1 \leq k \leq K$) and background ($k = 0$). These attributes include the 3D shapes and appearances of objects and background in an automatically chosen global coordinate system. The dimensionalities of all the $z_{\text{attr}}$ with $1 \leq k \leq K$ are identical, and are in general different from the dimensionality of $z_{\text{attr}}^{m,k}$. The priors of all the $z_{\text{attr}}^{m,k}$ with $0 \leq k \leq K$ are standard normal distributions.

- $\rho_{1;K}$ and $z_{\text{prs}}^{m,k}$ are used to model the number of objects in the visual scene, considering that different visual scenes may contain different numbers of objects. The binary latent variable $z_{\text{prs}}^{m,k} \in \{0, 1\}$ indicates whether the $k$th object is included in the visual scene (i.e., the number of objects is $\sum_{k=1}^{K} z_{\text{prs}}^{m,k}$), and is sampled from a Bernoulli distribution with the latent variable $\rho_k$ as its parameter. The priors of all the $\rho_k$ with $1 \leq k \leq K$ are beta distributions parameterized by hyperparameters $\alpha$ and $K$.

Viewpoint-Dependent Latent Variables

Viewpoint-dependent latent variables may vary as the viewpoint changes. These latent variables include $z_{\text{view}}^{m}$ and $z_{\text{shp}}^{m,1;K.1:N}$.

- $z_{\text{view}}^{m}$ determines the viewpoint (in an automatically chosen global coordinate system) of the $m$th image, and is drawn from a standard normal prior distribution.

- $z_{\text{shp}}^{m,1;K.1:N} \in \{0, 1\}^{K \times N}$ consist of binary latent variables that indicate the complete shapes of objects in the image coordinate system determined by the $m$th viewpoint. Each element of $z_{\text{shp}}^{m,1;K.1:N}$ is sampled independently from a Bernoulli distribution, whose parameter is computed by transforming latent variables $z_{\text{view}}^{m}$ and $z_{\text{attr}}^{m,k}$ ($1 \leq k \leq K$) with a neural network $f_{\text{shp}}$ that captures the spatial dependencies among pixels. The sigmoid activation function in the last layer of $f_{\text{shp}}$ is explicitly expressed to clarify the output range of the neural network.

Likelihood Function

All the pixels of the images $x_{1:M,1:N}$ are assumed to be conditional independent of each other given all the latent variables $\Omega$, and the likelihood function $p(x|\Omega)$ is assumed to be factorized as the product of several normal distributions with varying mean vectors and constant covariance matrices, i.e., $\prod_{m=1}^{M} \prod_{n=1}^{N} N(\sum_{k=0}^{K} s_{m,k,n} a_{m,k,n}, \sigma^2 I)$. To compute the mean vectors, intermediate variables $\sigma$, $s$, and $a$ need to be computed by transforming the sampled latent variables with deterministic functions.

- $o_{m,1;K}$ characterize the depth ordering of objects in the image observed from the $m$th viewpoint. If multiple objects overlap, the object with the largest value of $o_{m,k}$ is assumed to obscure the others in a soft and differentiable way. To compute $o_{m,k}$, latent variables $z_{\text{view}}^{m}$ and $z_{\text{attr}}^{m,k}$ are first transformed by a neural network $f_{\text{ord}}$, and then the exponential function is applied to the output of $f_{\text{ord}}$ divided by $\lambda$. The exponential function ensures that the value of $o_{m,k}$ is greater than 0, and the hyperparameter $\lambda$ controls the softness of object occlusions.

- $s_{m,0;K.1:N}$ indicate the perceived shapes of objects ($1 \leq k \leq K$) and background ($k = 0$) in the $m$th image, and satisfy the constraints that $(\forall m,k,n) 0 \leq s_{m,k,n} \leq 1$ and $(\forall m,n) \sum_{k=0}^{K} s_{m,k,n} = 1$. These latent variables are computed based on $z_{\text{prs}}^{m,k}$, $z_{\text{shp}}^{m,1;K.1:N}$, and $o_{m,1;K}$. Because $z_{\text{prs}}^{m,k}$ and $z_{\text{shp}}^{m,1;K.1:N}$ are binary variables, the perceived shape $s_{m,0;1:N}$ of background is also binary, and equals 1 at the pixels that are not covered by any object. The computation of perceived shapes $s_{m,1;K.1:N}$ of objects at each pixel can be interpreted as a masked softmax operation that only considers the objects covering that pixel. As the hyperparameter $\lambda$ in the computation of $o$ approaches 0, the perceived shapes $s_{m,0;K.1:N}$ of all the objects and background at each pixel approach a one-hot vector.
• \( a_{m,0:K,1:N} \) contain information about the complete appearances of objects (1 ≤ k ≤ K) and the background appearance (k = 0) in the mth image, which are computed by transforming latent variables \( z_{\text{view}} \) and \( z_{\text{attr}} \) with neural networks \( f_{\text{shp}} \) (for k = 0) and \( f_{\text{ape}} \) (for 1 ≤ k ≤ K). Appearances of objects and the background image are computed differently because the dimensionality of \( z_{\text{0:attr}} \) is in general different from \( z_{\text{k:attr}} \) with 1 ≤ k ≤ K.

### Generative Model

The mathematical expressions of the generative model are

\[
z_{\text{view}}^m \sim \mathcal{N}(0, I);
\]

\[
z_{\text{attr}}^m \sim \mathcal{N}(0, I);
\]

\[
\rho_k \sim \text{Beta} \left( \alpha / K, 1 \right);
\]

\[
z_{\text{ps}}^m,k,n \sim \text{Ber} \left( \sigma_k \right)
\]

\[
o_{m,k} = \exp \left( f_{\text{od}}(z_{\text{view}}^m, z_{\text{attr}}^m) / \lambda \right)
\]

\[
s_{m,k,n} = \left\{ \begin{array}{ll}
(1 - z_{\text{ps}}^k) z_{\text{m},k,n}, & k = 0 \\
(1 - s_{m,0,n}) z_{\text{ps}}^k z_{\text{m},k,n} o_{m,k}, & 1 \leq k \leq K
\end{array} \right.
\]

\[
a_{m,k,n} = \left\{ \begin{array}{ll}
(1 - z_{\text{ps}}^k) z_{\text{m},k,n} o_{m,k}, & k = 0 \\
(1 - s_{m,0,n}) z_{\text{ps}}^k z_{\text{m},k,n} o_{m,k}, & 1 \leq k \leq K
\end{array} \right.
\]

\[
x_{m,n} \sim \mathcal{N} \left( \sum_{k=0}^{K} s_{m,k,n} a_{m,k,n}, \sigma_k^2 I \right)
\]

In the above expressions, some of the ranges of indexes \( m \) (1 ≤ m ≤ M), \( n \) (1 ≤ n ≤ N), and \( k \) (0 ≤ k ≤ K) for \( z_{\text{attr}} \), and 1 ≤ k ≤ K for \( \rho_k, z_{\text{ps}}, z_{\text{shp}}, o \) are omitted for notational simplicity. \( \alpha, \lambda, \) and \( \sigma_k \) are tunable hyperparameters.

Let \( \Omega = \{ z_{\text{view}}, z_{\text{attr}}, \rho, z_{\text{ps}}, z_{\text{shp}} \} \) be the collection of all latent variables. The joint probability of \( \mathbf{x} \) and \( \Omega \) is

\[
p(\mathbf{x}, \Omega) = \prod_{k=1}^{K} p(z_{\text{attr}}^k) \prod_{k=1}^{K} p(\rho_k) p(z_{\text{ps}}^k | \rho_k)
\]

\[
\prod_{m=1}^{M} p(z_{\text{view}}^m) \prod_{k=1}^{K} p(z_{\text{m},k,n} | z_{\text{view}}^m, z_{\text{attr}}^k)
\]

\[
\prod_{m=1}^{M} \prod_{n=1}^{N} p(x_{m,n} | z_{\text{view}}^m, z_{\text{attr}}^k, a_{m,k,n})
\]

### Inference and Learning

The exact posterior distribution of latent variables \( p(\Omega | \mathbf{x}) \) is intractable to compute. Therefore, we adopt amortized variational inference, which approximates the complex posterior distribution with a tractable variational distribution \( q(\Omega | \mathbf{x}) \), and apply neural networks to transform the images \( \mathbf{x} \) into parameters of the variational distribution. The neural networks \( f_{\text{shp}}, f_{\text{ape}}, f_{\text{shp}}, f_{\text{ape}} \) in the generative model, as well as the inference networks, are jointly optimized with the goal of maximizing the evidence lower bound (ELBO). Details of the inference and learning are described below.

### Inference of Latent Variables

The variational distribution \( q(\Omega | \mathbf{x}) \) is factorized as

\[
q(\Omega | \mathbf{x}) = \prod_{m} q(z_{\text{attr}}^m | \mathbf{x}) \prod_{k} q(\rho_k | \mathbf{x}) q(z_{\text{ps}}^k | \mathbf{x})
\]

\[
\prod_{m} q(z_{\text{view}}^m | \mathbf{x}) \prod_{k} \prod_{n} q(z_{\text{shp}}^m,k,n | z_{\text{view}}^m, z_{\text{attr}}^k, x)
\]

#### Algorithm 1: Inference of latent variables

**Input:** Images of M viewpoints \( \mathbf{x}_{1:M} \)

**Output:** Parameters of \( q(\Omega | \mathbf{x}) \)

1. // Extract features and initialize intermediate variables
   1. \( y_{\text{feat}}^m \leftarrow g_{\text{feat}}(\mathbf{x}_m) \), \( \forall 1 \leq m \leq M \)
   2. \( y_{\text{view}}^m \sim \mathcal{N}(\mu_{\text{view}}^m, \text{diag}(\sigma_{\text{view}}^m)) \), \( \forall 1 \leq m \leq M \)
   3. \( y_{\text{attr}}^m \sim \mathcal{N}(\mu_{\text{attr}}^m, \text{diag}(\sigma_{\text{attr}}^m)) \), \( \forall 0 \leq k \leq K \)
   5. // Update intermediate variables \( y_{\text{view}}^m \) and \( y_{\text{attr}}^m \)
   6. for \( t \leftarrow 1 \) to \( T \) do \( \{ \forall 1 \leq m \leq M, 0 \leq k \leq K \} \) in the loop
   7. \( y_{\text{full}}^m \leftarrow [y_{\text{view}}^m, y_{\text{attr}}^m] \)
   8. \( a_{m,k} \leftarrow \text{softmax}(y_{\text{attr}}^m) \), \( \forall 0 \leq k \leq K \)
   9. \( \mu_{\text{view}}^m \leftarrow \sum_{m} \text{softmax}(\log a_{m,k}) y_{\text{full}}^m \)
   10. \( [y_{\text{view}}^m, y_{\text{attr}}^m] \leftarrow \logpdf(y_{\text{full}}^m) \cdot u_{1:M,0:K} \)
   11. \( y_{\text{view}}^m \leftarrow \text{mean}(y_{\text{view}}^m) \)
   12. \( y_{\text{attr}}^m \leftarrow \text{mean}(y_{\text{attr}}^m) \)

**The ranges of indexes in Eq. 2 are identical to the ones in Eq. 1, and are omitted for simplicity. The choices of terms on the right-hand side of Eq. 2 are**

\[
q(z_{\text{attr}}^k | \mathbf{x}) = \mathcal{N}(z_{\text{attr}}^k | \mu_{\text{attr}}^k, \text{diag}(\sigma_{\text{attr}}^k)^2)
\]

\[
q(\rho_k | \mathbf{x}) = \text{Beta}(\rho_k; \tau_{k,1}, \tau_{k,2})
\]

\[
q(z_{\text{ps}}^k | \mathbf{x}) = \text{Ber}(z_{\text{ps}}^k; \kappa_k)
\]

\[
q(z_{\text{view}}^m | \mathbf{x}) = \mathcal{N}(z_{\text{view}}^m | \mu_{\text{view}}^m, \text{diag}(\sigma_{\text{view}}^m)^2)
\]

\[
q(z_{\text{shp}}^m,k,n | z_{\text{view}}^m, z_{\text{attr}}^k, x) = p(z_{\text{shp}}^m,k,n | z_{\text{view}}^m, z_{\text{attr}}^k, x)
\]

In the variational distribution, \( q(z_{\text{attr}}^k | \mathbf{x}) \) and \( q(z_{\text{view}}^m | \mathbf{x}) \) are normal distributions with diagonal covariance matrices. \( z_{\text{ps}} \) is assumed to be independent of \( \rho_k \) given \( \mathbf{x} \), and \( q(\rho_k | \mathbf{x}) \) and \( q(z_{\text{ps}}^k | \mathbf{x}) \) are chosen to be a beta distribution and a Bernoulli distribution, respectively. The advantage of this formulation is that the Kullback-Leibler (KL) divergence between \( q(\rho_k | \mathbf{x})q(z_{\text{ps}}^k | \mathbf{x}) \) and \( p(\rho_k)q(z_{\text{ps}}^k | \mathbf{x}) \) has a closed-form solution. For simplicity, \( q(z_{\text{shp}}^m,k,n | z_{\text{view}}^m, z_{\text{attr}}^k, x) \) is assumed to be identical to \( p(z_{\text{shp}}^m,k,n | z_{\text{view}}^m, z_{\text{attr}}^k, x) \) in the generative model, so that no extra inference network is needed for \( z_{\text{shp}}^m,k,n \). The procedure to compute the parameters \( \mu_{\text{attr}}, \sigma_{\text{attr}}, \tau, \kappa, \mu_{\text{view}}, \sigma_{\text{view}} \), and \( \text{diag}(\sigma_{\text{view}}) \) of these distributions is presented in Algorithm 1 and the brief explanations are given below.

First, the feature maps \( y_{\text{feat}}^m \) of each image \( \mathbf{x}_m \) are extracted by a neural network \( g_{\text{feat}} \). Next, intermediate variables \( y_{\text{view}}^m \) and \( y_{\text{attr}}^m \) which fully characterize parameters of the viewpoint-dependent \( (\mu_{\text{view}}^m, \sigma_{\text{view}}^m) \) and viewpoint-independent \( (\mu_{\text{attr}}, \sigma_{\text{attr}}, \tau, \kappa) \) latent variables are not di-
points are integrated using neural networks \(\hat{\mu}_{\text{view}}, \hat{\sigma}_{\text{view}}, \hat{\mu}_{\text{attr}}, \) and \(\hat{\sigma}_{\text{attr}}\) and then iteratively updated, considering that there are infinitely many possible solutions (e.g., due to the change of global coordinate system) to disentangle the image representations into a viewpoint-dependent part and a viewpoint-independent part. In each step of the iterative updates, information of images observed from different viewpoints are integrated using neural networks \(g_{\text{key}}, g_{\text{qry}}, g_{\text{val}}\), and \(g_{\text{upd}}\), based on attentions between feature maps \(y_{\text{feat}}\) and intermediate variables \(y_{\text{view}}\) and \(y_{\text{attr}}\). To achieve permutation equivariance, which has been considered as an important property in object-centric learning (Emami et al. 2021), objects and background are not distinguished in the initialization and updates of \(y_{\text{attr}}\), and the index \(k^*\) that corresponds to background is determined after the iterative updates, by applying a neural network \(g_{\text{sel}}\) to transform \(y_{\text{attr}}\) into parameters \(\pi\) of a categorical distribution and sampling from the distribution. After rearranging \(y_{\text{attr}}\) based on \(k^*\), parameters of the variational distribution are computed by transforming \(y_{\text{attr}}, y_{\text{view}}, y_{\text{attr}}^{\text{sel}}, k\), and \(y_{\text{view}}\) with neural networks \(g_{\text{key}}, g_{\text{qry}}, g_{\text{val}}\), and \(g_{\text{view}}\), respectively. For further details, please refer to the Supplementary Material.

**Learning of Neural Networks**

The neural networks used in both the generative model and the amortized variational inference (including learnable parameters \(\mu_{\text{view}}, \sigma_{\text{view}}, \mu_{\text{attr}}, \) and \(\sigma_{\text{attr}}\)), are jointly optimized by minimizing the negative value of evidence lower bound (ELBO) that serves as the loss function \(\mathcal{L}\). The expression of \(\mathcal{L}\) is briefly given below, and a more detailed version is included in the Supplementary Material.

\[
\mathcal{L} = -\sum_m \sum_n \mathbb{E}_q(\Omega|x) \left[ \log p(x_{m,n}|z_{\text{view}}, z_{\text{attr}}, z_{\text{prs}}, z_{\text{shp}}) \right] \\
+ \sum_m D_{KL}(q(z_{\text{view}}^m|x)||p(z_{\text{view}}^m)) \\
+ \sum_k D_{KL}(q(z_{\text{attr}}^k|x)||p(z_{\text{attr}}^k)) \\
+ \sum_k D_{KL}(q(p_k|x)||p(p_k)) \\
+ \sum_{k} \mathbb{E}_q(p_k|x) \left[ D_{KL}(q(z_{\text{prs}}^k|x)||p(z_{\text{prs}}^k/p_k)) \right]
\]

In Eq. (3), the first term is negative log-likelihood, and the rest four terms are Kullback-Leibler (KL) divergences that are computed by \(D_{KL}(q||p) = \mathbb{E}_q[\log q - \log p]\). The loss function is optimized using the gradient-based method. All the KL divergences have closed-form solutions, and the gradients of these terms can be easily computed. The negative log-likelihood cannot be computed analytically, and the gradients of this term is approximated by sampling latent variables \(z_{\text{view}}, z_{\text{attr}}, z_{\text{prs}}, \) and \(z_{\text{shp}}\) from the variational distribution \(q(\Omega|x)\). To reduce the variances of gradients, the continuous variables \(z_{\text{view}}\) and \(z_{\text{attr}}\) are sampled using the reparameterization trick (Salimans and Knowles 2013, Kingma and Welling 2014), and the discrete variables \(z_{\text{prs}}\) and \(z_{\text{shp}}\) are approximated using a continuous relaxation (Maddison, Mnih, and Teh 2017, Jang, Gu, and Poole 2017). To learn the neural network \(g_{\text{key}}\) that computes parameters of the categorical distribution from which the background index \(k^*\) is sampled, NVIL (Mnih and Gregor 2014) is applied to obtain low-variance and unbiased estimates of gradients.

**Experiments**

In this section, we aim to verify that the proposed method:

- is able to learn from multiple viewpoints without any supervision, which cannot be solved by existing methods;
- competes with existing state-of-the-art methods that use viewpoint annotations in the learning;
- is comparable to the state-of-the-arts under an extreme condition that scenes are observed from one viewpoint.

**Evaluation Metrics:** Several metrics are used to evaluate the performance from four aspects. 1) *Adjusted Rand Index* (ARI) (Hubert and Arabie 1985) and *Adjusted Mutual Information* (AMI) (Nguyen, Epps, and Bailey 2010) assess the quality of segmentation, i.e., how accurately images are partitioned into different objects and background. Previous work usually evaluates ARI and AMI only at pixels belong to objects, and how accurately background is separated from objects is unclear. We evaluate ARI and AMI under two conditions. ARI-A and AMI-A are computed considering both objects and background, while ARI-O and AMI-O are computed considering only objects. 2) *Intersection over Union* (IoU) and *F1 score* (F1) assess the quality of amodal segmentation, i.e., how accurately complete shapes of objects are estimated. 3) *Object Counting Accuracy* (OCA) assesses the accuracy of the estimated number of objects. 4) *Object Ordering Accuracy* (OQA) as used in (Yuan, Li, and Xue 2019a) assesses the accuracy of the estimated pairwise ordering of objects. Formal definitions of these metrics are included in the Supplementary Material.

**Multi-Viewpoint Learning**

**Datasets:** The experiments are performed on four multi-viewpoint variants (referred to as CLEVR-M1 to CLEVR-M4) of the commonly used CLEVR dataset that differ in the ranges to sample viewpoints and in the attributes of objects. CLEVR-M3/CLEVR-M4 is harder than CLEVR-M1/CLEVR-M2 in that the poses of objects are more dissimilar in different images of the same visual scene because viewpoints are sampled from a larger range. CLEVR-M2/CLEVR-M4 is harder than CLEVR-M1/CLEVR-M3 in that there are fewer visual cues to distinguish objects from one another because all the objects in the same visual scene share the same colors, shapes, and materials. Further details are described in the Supplementary Material.

**Comparison Methods:** It is worth noting that the proposed method cannot be directly compared with existing methods in the novel problem setting considered in this paper. To verify that the proposed method can effectively achieve object constancy, a baseline method that does not maintain the identities of objects across viewpoints is compared with. This baseline method is derived from the proposed method by assigning each viewpoint a separate set of latent variables \(z_{\text{attr}}, \rho, \) and \(z_{\text{prs}}\) (all latent variables are viewpoint-dependent). To verify that the proposed method can effec-

---

1 Code is available at https://git.io/JDnne
Figure 2: Scene decomposition results of the proposed method in the multi-viewpoint learning setting. Objects are sorted based on the estimated $z_{prs}$. Models are tested with $K = 7$, and the last two objects with $z_{prs}^k = 0$ are not shown.

Figure 3: Results of interpolating and sampling viewpoints in latent space. The $i$th row of each sub-figure corresponds to the results evaluated on the CLEVR-M$_i$ dataset.

Quantitatively learn without supervision, we compare it with MulMON (Li, Eastwood, and Fisher 2020), which solves a simpler problem by using viewpoint annotations in both learning and testing. Another representative partially supervised method ROOTS (Chen, Deng, and Ahn 2020) is not compared with because the official code is not publicly available.

**Scene Decomposition:** Qualitative results of the proposed method evaluated on the CLEVR-M1 and CLEVR-M2 datasets are shown in Figure 2. The proposed method is able to achieve object constancy even if objects are fully occluded (object 1 in columns 1 and 4 of sub-figure (a)). In addition, under the circumstances that objects are less identifiable and the poses of objects vary significantly across different viewpoints (objects 1~4 in sub-figure (b)), the proposed method can also correctly identify the same objects across viewpoints. The proposed method tends to treat shadows as parts of objects instead of background, which is desirable because lighting effects are not explicitly modeled and the shadows will change accordingly as the objects move. More results can be found in the Supplementary Material.

Quantitative comparison of scene decomposition performance on all the datasets is presented in Table 1. The proposed method achieves high ARI-O, AMI-O, and OOA scores. As for ARI-A, AMI-A, IoU, and F1, the achieved performance is not so well. The major reason is that the proposed method tends to treat regions of shadows as objects, while they are considered as background in the ground truth annotations. MulMON also tends to incorrectly estimate shadows as objects, but slightly outperforms the proposed method in terms of ARI-A and AMI-A on all the datasets, possibly because MulMON does not explicitly model the complete shapes, the number, and the depth ordering of objects, but directly computes the perceived shapes using the softmax function, which makes it easier to learn the boundary regions of objects. For the similar reason, the IoU, F1, and OOA scores which require the estimations of complete shapes and depth ordering are not evaluated for MulMON. The OCA scores are computed based on the heuristically estimated number of objects (details in the Supplementary Material). The unsupervised proposed method achieves competitive or slightly better results compared to the partially supervised MulMON, which has validated the motivation of the proposed method.

**Generalizability:** Because visual scenes are modeled compositionally by the proposed method, the trained models are generalizable to novel scenes containing more numbers of objects than the ones used for training. Evaluations of generalizability are included in the Supplementary Material. Although the increased number of objects makes it more difficult to extract compositional scene representations, the proposed method performs reasonably well.

**Viewpoint Estimation:** The proposed method is able to estimate the viewpoints of images, under the condition that the viewpoint-independent attributes of objects and background are known. More specifically, given the approximate posteriors of object-centric representations, the proposed method is able to infer the corresponding viewpoint representations of different observations of the same visual scene. Please refer to the Supplementary Material for more details.

**Viewpoint Modification:** Multi-viewpoint images of the same visual scene can be generated by first inferring compositional scene representations and then modifying viewpoint
Table 1: Comparison of scene decomposition performance when learning from multiple viewpoints. All the methods are trained and tested with \( M = 4 \) and \( K = 7 \). The proposed fully unsupervised method achieves competitive or slightly better results compared with MulMON with viewpoint supervision.

| Dataset | Method | ARI-A | AMI-A | ARI-O | AMI-O | IoU | F1 | OCA | OOA |
|---------|--------|-------|-------|-------|-------|-----|----|-----|-----|
| CLEVR-M1 | Baseline | 0.512±0.03-4 | 0.361±0.03-3 | 0.269±0.01-2 | 0.418±0.01-2 | 0.171±0.03-3 | 0.279±0.04-3 | 0.004±0.05-3 | 0.628±0.03-2 |
| CLEVR-M2 | MulMON | \textbf{0.615±0.02-3} | \textbf{0.560±0.02-3} | 0.927±0.05-3 | 0.917±0.02-3 | N/A | N/A | 0.446±0.05-2 | N/A |
| Proposed | 0.507±0.02-3 | 0.486±0.02-3 | 0.948±0.03-3 | 0.934±0.02-3 | \textbf{0.422±0.03-3} | \textbf{0.603±0.03-3} | \textbf{0.730±0.05-2} | \textbf{0.970±0.01-2} |
| CLEVR-M3 | Baseline | 0.505±0.01-3 | 0.356±0.03-3 | 0.274±0.01-2 | 0.422±0.01-2 | 0.167±0.01-3 | 0.273±0.05-3 | 0.004±0.05-3 | 0.682±0.02-2 |
| Proposed | 0.507±0.03-3 | 0.479±0.02-3 | 0.941±0.03-3 | 0.933±0.02-3 | 0.428±0.03-3 | 0.587±0.07-3 | 0.686±0.05-2 | 0.939±0.02-2 |
| CLEVR-M4 | Baseline | 0.551±0.01-3 | 0.372±0.03-3 | 0.278±0.01-2 | 0.425±0.01-2 | 0.173±0.05-3 | 0.283±0.07-3 | 0.000±0.05-3 | 0.600±0.01-2 |
| Proposed | 0.591±0.01-3 | 0.552±0.01-3 | 0.938±0.02-3 | 0.923±0.02-3 | N/A | N/A | 0.424±0.05-2 | N/A |

Table 2: Comparison of scene decomposition performance when learning from a single viewpoint. All the methods are trained and tested with \( K = 6, 5 \), and \( K = 7 \) on the dSprites, Abstract, and CLEVR datasets, respectively. The proposed method is comparable with the state-of-the-arts under the extreme condition that visual scenes are observed from one viewpoint.

| Dataset | Method | ARI-A | AMI-A | ARI-O | AMI-O | IoU | F1 | OCA | OOA |
|---------|--------|-------|-------|-------|-------|-----|----|-----|-----|
| dSprites | Slot Attn | 0.142±0.03-3 | 0.286±0.03-3 | 0.935±0.01-3 | 0.917±0.01-3 | N/A | N/A | 0.000±0.01-0 | N/A |
| | GMIOO | 0.969±0.02-4 | 0.922±0.02-4 | 0.956±0.01-3 | 0.950±0.02-4 | 0.860±0.04-4 | 0.911±0.01-4 | \textbf{0.874±0.02-3} | 0.891±0.05-3 |
| | SPACE | 0.946±0.07-4 | 0.874±0.06-4 | 0.858±0.01-3 | 0.870±0.05-4 | 0.729±0.07-4 | 0.805±0.05-4 | 0.587±0.04-3 | 0.624±0.01-2 |
| | Proposed | 0.959±0.02-3 | 0.907±0.02-4 | 0.939±0.01-3 | 0.922±0.02-4 | 0.861±0.04-4 | 0.912±0.01-4 | 0.813±0.04-3 | \textbf{0.899±0.05-3} |
| Abstract | Slot Attn | 0.940±0.05-4 | 0.877±0.03-3 | 0.935±0.02-4 | 0.903±0.01-4 | N/A | N/A | 0.888±0.03-5 | N/A |
| | GMIOO | 0.832±0.02-4 | 0.751±0.03-4 | 0.947±0.02-3 | 0.927±0.01-3 | 0.750±0.08-4 | 0.849±0.04-4 | 0.955±0.02-3 | 0.940±0.03-3 |
| | SPACE | 0.888±0.06-4 | 0.797±0.06-4 | 0.816±0.01-3 | 0.817±0.02-3 | 0.722±0.07-4 | 0.798±0.08-4 | 0.685±0.02-3 | 0.795±0.05-3 |
| | Proposed | 0.887±0.03-4 | 0.812±0.07-4 | 0.947±0.01-3 | 0.933±0.02-4 | \textbf{0.801±0.03-4} | \textbf{0.883±0.02-4} | \textbf{0.940±0.03-3} | \textbf{0.962±0.02-3} |
| CLEVR | Slot Attn | 0.026±0.02-4 | 0.240±0.03-4 | 0.985±0.06-4 | 0.983±0.04-4 | N/A | N/A | 0.002±0.01-3 | N/A |
| | GMIOO | 0.716±0.05-4 | 0.665±0.04-4 | 0.943±0.01-3 | 0.955±0.03-4 | 0.603±0.02-3 | 0.725±0.02-3 | 0.683±0.02-3 | 0.906±0.04-3 |
| | SPACE | \textbf{0.860±0.03-4} | 0.796±0.03-4 | 0.976±0.03-4 | 0.973±0.01-4 | \textbf{0.776±0.03-4} | \textbf{0.863±0.03-4} | \textbf{0.711±0.02-3} | \textbf{0.930±0.07-4} |
| | Proposed | 0.649±0.01-4 | 0.614±0.02-4 | 0.982±0.09-4 | 0.978±0.05-4 | 0.591±0.06-4 | 0.736±0.07-4 | \textbf{0.875±0.08-3} | \textbf{0.952±0.05-3} |

**Single-Viewpoint Learning**

**Datasets:** Three datasets are constructed based on the dSprites (Matthey et al. 2017), Abstract Scene (Zitnick and Parikh 2013), and CLEVR (Johnson et al. 2017) datasets, in a way similar to the Multi-Objects Datasets (Kabra et al. 2019) but provides extra annotations (for evaluation only) of complete shapes of objects. These datasets are referred to as dSprites, Abstract, and CLEVR for simplicity. More details are provided in the Supplementary Material.

**Comparison Methods:** The proposed method is compared with three state-of-the-art compositional scene representation methods. Slot Attention (Locatello et al. 2020) is chosen because the proposed method adopts a similar attention mechanism in the inference. GMIOO (Yuan, Li, and Xue 2019a) and SPACE (Lin et al. 2020) are chosen because they are two representative methods that also explicitly model the varying number of objects, and can distinguish background from objects and determine the depth ordering of objects. 

**Experimental Results:** Comparison of scene decomposition performance under the extreme condition that each visual scene is only observed from one viewpoint is shown in Table 2. The proposed method is competitive with the state-of-the-arts and achieves the best or the second-best scores in almost all the cases. The Supplementary Material includes discussions and further quantitative and qualitative results.

**Conclusions**

In this paper, we have considered a novel problem of learning compositional scene representations from multiple unspecified viewpoints in a fully unsupervised way, and proposed a deep generative model called OCLOC to solve this problem. On several specifically designed synthesized datasets, the proposed fully unsupervised method achieves competitive or slightly better results compared with a state-of-the-art method with viewpoint supervision, which has validated the effectiveness of the proposed method.
Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No.62176060), STCSM project (No.20511100400), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103), Shanghai Research and Innovation Functional Program (No.17DZ2260900), and the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

References

Burgess, C. P.; Matthey, L.; Watters, N.; Kabra, R.; Higgins, I.; Botvinick, M.; and Lerchner, A. 2019. MONet: Unsupervised scene decomposition and representation. ArXiv, 1901.11390.

Chen, C.; Deng, F.; and Ahn, S. 2020. ROOTS: Object-centric representation and rendering of 3D scenes. ArXiv, 2006.06130.

Crawford, E.; and Pineau, J. 2019. Spatially invariant unsupervised object detection with convolutional neural networks. In AAAI, 3412–3420.

Crawford, E.; and Pineau, J. 2020. Exploiting spatial invariance for scalable unsupervised object tracking. In AAAI, 3684–3692.

Emami, P.; He, P.; Ranka, S.; and Rangarajan, A. 2021. Efficient iterative amortized inference for learning symmetric and disentangled multi-object representations. In ICML, 2970–2981.

Engelcke, M.; Kosior, A. R.; Jones, O. P.; and Posner, I. 2020. GENESIS: Generative scene inference and sampling with object-centric latent representations. In ICLR.

Eslami, S.; Heess, N.; Weber, T.; Tassa, Y.; Szepesvari, D.; Kavukcuoglu, K.; and Hinton, G. E. 2016. Attend, infer, repeat: Fast scene understanding with generative models. In NeurIPS, 3225–3233.

Eslami, S.; Rezende, D. J.; Besse, F.; Viola, F.; Morcos, A. S.; Gamelo, M.; Ruderman, A.; Rusu, A. A.; Danihelka, I.; Gregor, K.; Reichert, D. P.; Buesing, L.; Weber, T.; Vinyals, O.; Rosenbaum, D.; Lillicrap, P. T.; King, H.; Hillier, C.; Botvinick, M.; Wierstra, D.; Kavukcuoglu, K.; and Hassabis, D. 2018. Neural scene representation and rendering. Science, 360: 1204–1210.

Fodor, J.; and Pylyshyn, Z. 1988. Connectionism and cognitive architecture: A critical analysis. Cognition, 28: 3–71.

Greff, K.; Kaufman, R. L.; Kabra, R.; Watters, N.; Burgess, C. P.; Zoran, D.; Matthey, L.; Botvinick, M.; and Lerchner, A. 2019. Multi-object representation learning with iterative variational inference. In ICML, 2424–2433.

Greff, K.; van Steenkiste, S.; and Schmidhuber, J. 2017. Neural Expectation Maximization. In NeurIPS, 6694–6704.

He, Z.; Li, J.; Liu, D.; He, H.; and Barber, D. 2019. Tracking by animation: Unsupervised learning of multi-object attentive trackers. In CVPR, 1318–1327.

Huang, J.; and Murphy, K. 2016. Efficient inference in occlusion-aware generative models of images. In ICLR Workshop.

Hubert, L.; and Arabie, P. 1985. Comparing partitions. Journal of Classification, 2: 193–218.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical reparameterization with Gumbel-softmax. In ICLR.

Jiang, J.; and Ahn, S.-J. 2020. Generative neurosymbolic machines. In NeurIPS, 12572–12582.

Jiang, J.; Janghorbani, S.; de Melo, G.; and Ahn, S. 2020. SCALOR: Generative world models with scalable object representations. In ICLR.

Johnson, J.; Hariharan, B.; van der Maaten, L.; Fei-Fei, L.; Zitnick, C. L.; and Girshick, R. B. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In CVPR, 1988–1997.

Johnson, S. 2010. How infants learn about the visual world. Cognitive Science, 34 7: 1158–1184.

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

Kingma, D. P.; and Welling, M. 2014. Auto-encoding variational Bayes. In ICLR.

Kosior, A. R.; Kim, H.; Posner, I.; and Teh, Y. 2018. Sequential attend, infer, repeat: Generative modelling of moving objects. In NeurIPS, 8615–8625.

Lake, B.; Ullman, T. D.; Tenenbaum, J.; and Gershman, S. 2017. Building machines that learn and think like people. The Behavioral and Brain Sciences, 40: e253.

Li, N.; Eastwood, C.; and Fisher, R. B. 2020. Learning object-centric representations of multi-object scenes from multiple views. In NeurIPS, 5656–5666.

Lin, Z.; Wu, Y.-F.; Peri, S.; Sun, W.; Singh, G.; Deng, F.; Jiang, J.; and Ahn, S. 2020. SPACE: Unsupervised object-oriented scene representation via spatial attention and decomposition. In ICLR.

Locatello, F.; Weissborn, D.; Unterthiner, T.; Mahendran, A.; Heigold, G.; Uszkoreit, J.; Dosovitskiy, A.; and Kipf, T. 2020. Object-centric learning with slot attention. In NeurIPS, 11515–11528.

Maddison, C. J.; Mnih, A.; and Teh, Y. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In ICLR.

Marino, J.; Yue, Y.; and Mandt, S. 2018. Iterative amortized inference. In ICML, 3403–3412.

Marr, D. 1982. Vision: A computational investigation into the human representation and processing of visual information. Henry Holt and Co., Inc.

Matthey, L.; Higgins, I.; Hassabis, D.; and Lerchner, A. 2017. dSprites: Disentanglement testing Sprites dataset. https://github.com/deepmind/dsprites-dataset/.

Mnih, A.; and Gregor, K. 2014. Neural variational inference and learning in belief networks. In ICML, 1791–1799.

Nguyen, X.; Epps, J.; and Bailey, J. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research, 11: 2837–2854.
Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An imperative style, high-performance deep learning library. In NeurIPS, 8026–8037.

Qi, L.; Jiang, L.; Liu, S.; Shen, X.; and Jia, J. 2019. Amodal instance segmentation with KINS dataset. In CVPR, 3009–3018.

Salimans, T.; and Knowles, D. A. 2013. Fixed-form variational posterior approximation through stochastic linear regression. Bayesian Analysis, 8(4): 837–882.

Shepard, R.; and Metzler, J. 1971. Mental rotation of three-dimensional objects. Science, 171: 701–703.

Stanic, A.; and Schmidhuber, J. 2019. R-SQAIR: Relational sequential attend, infer, repeat. In NeurIPS Workshop.

Turnbull, O.; Carey, D.; and McCarthy, R. 1997. The neuropsychology of object constancy. Journal of the International Neuropsychological Society, 3 3: 288–98.

van Steenkiste, S.; Chang, M.; Greff, K.; and Schmidhuber, J. 2018. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In ICLR.

Veerapaneni, R.; Co-Reyes, J. D.; Chang, M.; Janner, M.; Finn, C.; Wu, J.; Tenenbaum, J.; and Levine, S. 2020. Entity abstraction in visual model-based reinforcement learning. In CoRL, 1439–1456.

Weis, M. A.; Chitta, K.; Sharma, Y.; Brendel, W.; Bethge, M.; Geiger, A.; and Ecker, A. S. 2021. Benchmarking unsupervised object representations for video sequences. Journal of Machine Learning Research, 22(183): 1–61.

Yuan, J.; Li, B.; and Xue, X. 2019a. Generative modeling of infinite occluded objects for compositional scene representation. In ICML, 7222–7231.

Yuan, J.; Li, B.; and Xue, X. 2019b. Spatial Mixture Models with Learnable Deep Priors for Perceptual Grouping. In AAAI, 9135–9142.

Yuan, J.; Li, B.; and Xue, X. 2021. Knowledge-Guided Object Discovery with Acquired Deep Impressions. In AAAI, 10798–10806.

Zablotskaia, P.; Dominici, E. A.; Sigal, L.; and Lehrmann, A. M. 2021. PROVIDE: A probabilistic framework for unsupervised video decomposition. In UAI.

Zitnick, C. L.; and Parikh, D. 2013. Bringing semantics into focus using visual abstraction. In CVPR, 3009–3016.
Details of Algorithm 1
The feature maps $y_{feat} \in \mathbb{R}^{N \times D_y}$ summarize the information of nearby region at each pixel in the $m$th image $x_m \in \mathbb{R}^{N \times C}$, and are extracted by transforming $x_m$ with a neural network $g_{feat}$. Considering that there are infinitely many possible solutions (e.g., due to the change of global coordinate system) to disentangle the image representations into a viewpoint-dependent part and a viewpoint-independent part, parameters of the variational distribution are first randomly initialized and then iteratively updated. To simplify the updates of parameters of the variational distribution, we use intermediate variables $y_{view}^{m} \in \mathbb{R}^{M \times D_{view}}$ and $y_{attr}^{m} \in \mathbb{R}^{(K+1) \times D_{view}}$ to represent parameters of the viewpoint-dependent part ($\mu_{view}$ and $\sigma_{view}$) and the viewpoint-independent part ($\mu_{attr}$, $\sigma_{attr}$, $\tau$, and $\kappa$), respectively. These intermediate variables are sampled independently from normal distributions with learnable parameters $\mu_{view}$, $\sigma_{view}$, $\mu_{attr}$, and $\sigma_{attr}$. To achieve permutation equivariance, which has been considered as an important property in object-centric learning [Emami et al.2021], objects and background are not distinguished in the initialization and updates of $y_{attr}$, and the index that corresponds to background is determined after the iterative updates.

In each step of the iterative updates, intermediate variables $y_{view}^{m} \in \mathbb{R}^{M \times D_{view}}$ and $y_{attr}^{m} \in \mathbb{R}^{(K+1) \times D_{view}}$ are first broadcasted and concatenated to form $y_{full}^{m} \in \mathbb{R}^{M \times (K+1)D_{view}}$. Next, the attention maps $a_{m,k,n} \in [0,1]^{(K+1) \times N}$ ($1 \leq m \leq M$) are computed separately for each viewpoint, by normalizing the similarities between the keys $y_{view}^{m}$ and the queries $y_{view}^{m}$ across different objects and background (i.e., $(\forall m,n) \sum_{k=0}^{K} a_{m,k,n} = 1$) with temperature $\sqrt{D_{keys}}$. Both $g_{view}$ and $g_{attr}$ are neural networks, and the similarities are measured by first broadcasting keys and queries to $\mathbb{R}^{(K+1) \times N \times D_{view}}$, and then performing dot product in the last dimension. After that, $u \in \mathbb{R}^{M \times (K+1) \times D_{view}}$ which contains information to update $y_{view}$ and $y_{attr}$ is computed as the weighted average of the values $g_{val}(y_{view}^{m}) \in \mathbb{R}^{N \times D_{view}}$ across $N$ pixels, with the attention maps $a_{m,k,n} \in [0,1]^{(K+1) \times N}$ as weights. $g_{val}$ is a neural network, and the weighted average is computed by first broadcasting values and the normalized weights $softmax_N(\log a_{m,k,n})$ to $\mathbb{R}^{(K+1) \times N \times D_{view}}$, and then performing dot product in the second dimension. Finally, a neural network $g_{upd}$ is applied to transform $y_{full}$ and $u$ to $y_{view} \in \mathbb{R}^{M \times (K+1) \times D_{view}}$ and $v_{attr} \in \mathbb{R}^{M \times (K+1) \times D_{view}}$, and intermediate variables $y_{view}^{1:M,0,K}$ and $v_{attr}^{1:M,0,K}$ are updated as the averages of $y_{view}^{1:M,0,K}$ and $v_{attr}^{1:M,0,K}$ across the second and the first dimensions, respectively.

After the iterative updates, each $y_{attr}^{k}$ ($0 \leq k \leq K$) is transformed to a scalar by a neural network $g_{attr}$, which is expected to output high value for the $y_{attr}^{k}$ that corresponds to background and low values for the rest ones corresponding to objects. After normalizing the $K+1$ outputs to form valid parameters $\pi_{k,K}$ of a categorical distribution and sampling the index $k^*$ of background from the distribution, $y_{view}^{k^*}$ is rearranged so that $y_{view}^{0}$ and $y_{view}^{k^*}$ correspond to background and objects, respectively. The final outputs of the inference, i.e., parameters of the variational distribution, are computed by transforming $y_{view}^{0}$, $y_{view}^{1:M,0,K}$, and $y_{view}^{1:M}$ with neural networks $g_{bck}$, $g_{obj}$, and $g_{view}$, respectively.

Details of Loss Function
The loss function $\mathcal{L}$ can be decomposed as
$$\mathcal{L} = \frac{MNC}{2} \log 2\pi \sigma_x^2 + \mathcal{L}_{all} + \mathcal{L}_{view} + \mathcal{L}_{attr} + \mathcal{L}_{\rho} + \mathcal{L}_{\psi}$$

The first term on the right-hand side of the above equation is a constant. The rest terms are computed by
$$\mathcal{L}_{all} = \frac{1}{2\sigma_x^2} \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{k=0}^{K} \left[ (x_{m,n} - \sum_{i=0}^{K} s_{m,k,n} a_{m,k,n})^2 \right]$$
$$\mathcal{L}_{view} = \frac{1}{2} \sum_{m=1}^{M} \left( \sum_{i=1}^{K} (\mu_{view,i}^2 + \sigma_{view,i}^2 - \log \sigma_{view,i}^2 - 1) \right)$$
$$\mathcal{L}_{attr} = \frac{1}{2} \sum_{k=0}^{K} \left( \sum_{i=1}^{K} (\mu_{attr,i}^2 + \sigma_{attr,i}^2 - \log \sigma_{attr,i}^2 - 1) \right)$$
$$\mathcal{L}_{\rho} = \sum_{k=1}^{K} \left( \log \Gamma(\tau_{k,1}) + \log \Gamma(\tau_{k,2}) - \log \frac{\kappa}{K} \right) + \sum_{k=1}^{K} \left( \tau_{k,1} - \kappa \log(1 - \kappa) \right) + \sum_{k=1}^{K} \left( \tau_{k,2} - \kappa \log \kappa \right)$$
$$\mathcal{L}_{\psi} = \sum_{k=1}^{K} \left( \psi(\tau_{k,1} + \tau_{k,2}) + \kappa \log(\kappa) \right)$$

The $\Gamma$ and $\phi$ in the computations of both $\mathcal{L}_{\rho}$ and $\mathcal{L}_{\psi}$ are gamma and digamma functions, respectively.

Details of Evaluation Metrics
Formal definitions of evaluation metrics are given below. These metrics are only used to assess the performance of the models after training. The best model parameters (i.e., parameters of neural networks) are chosen based on the value of loss function on the validation set. All the models are trained once and tested for five runs. The reported scores included both means and standard deviations.

Adjusted Rand Index (ARI)
$\hat{K}_i$ denotes the ground truth number of objects in the $i$th visual scene of the test set, and $\hat{p}^i \in \{0,1\}^{M \times N \times (K+1)}$ is the ground truth pixel-wise partition of objects and background in the $M$ images of this visual scene. $K$ denotes the maximum number of objects that may appear in the visual scene, and $\hat{r}^i \in \{0,1\}^{M \times N \times (K+1)}$ is the estimated partition. ARI is compute using the following expressions.
$$\text{ARI} = \frac{1}{K} \sum_{i=1}^{l} \left( \frac{b_i^{all} - b_i^{row} \cdot b_i^{col}}{b_i^{row} \cdot b_i^{col}} - c_i \right)$$
The meanings of \( \tilde{K}_i, \hat{r}^i, K, \) and \( r^i \) are identical to the ones in the descriptions of ARI. AMI is computed by

\[
AMI = \frac{1}{I} \sum_{i=1}^{I} \frac{\text{MI}(\hat{l}_i, l^i) - \mathbb{E}[\text{MI}(\hat{l}_i, l^i)]}{(H(\hat{l}_i) + H(l^i))/2 - \mathbb{E}[\text{MI}(\hat{l}_i, l^i)]}
\]

where

\[
\hat{l}_i \in \{0, 1, \ldots, \hat{K}_i+1\}^{|[S]}
\]

\[
\hat{l}_j = \text{arg max}_k \hat{r}^i_{j,k,n} \quad \text{for} \quad (m_j, n_j) \in S_j
\]

\[
l^i \in \{0, 1, \ldots, K+1\}^{|[S]}
\]

\[
l^i_j = \text{arg max}_k r^i_{j,k,n} \quad \text{for} \quad (m_j, n_j) \in S_j
\]

In the above expressions, MI denotes mutual information and \( H \) denotes entropy. When computing AMI-A/AMI-O, the choice of \( S \) is the same as ARI-A/ARI-O.

**Intersection over Union (IOU)**

\( \hat{s}^i \in [0, 1]^{M \times N \times \hat{K}_i} \) and \( s^i \in [0, 1]^{M \times N \times K} \) denote the ground truth and estimated shapes of objects in the \( M \) images of the \( i \)th visual scene of the test set. IOU can be used to evaluate the performance of amodal instance segmentation \cite{Qi et al. 2019}. Compared to ARI and AMI, it provides extra information about the estimation of occluded regions of objects because complete shapes instead of perceived shapes of objects are used to compute this metric. Because both the number and the indexes of the estimated objects may be different from the ground truth, \( \hat{s}^i \) and \( s^i \) cannot be compared directly. Let \( \Xi \) be the set of all the \( \hat{K}_i \) possible permutations of the indexes \( \{1, 2, \ldots, K\} \). \( \xi^i \in \Xi \) is a permutation chosen based on the ground truth \( \hat{s}^i \) and estimated \( r^i \) partitions of objects and background using the following expression.

\[
\xi^i = \max_{\xi \in \Xi} \sum_{k=1}^{\hat{K}_i} \sum_{m=1}^{M} \sum_{n=1}^{N} \hat{r}^i_{m,n,k} \cdot r^i_{m,n,\xi^i_k}
\]

IOU is computed by

\[
\text{IOU} = \frac{1}{I} \sum_{i=1}^{I} \frac{K_i}{K_i + \text{min}(\hat{s}^i_{m,n,k}, s^i_{m,n,k})}
\]

where

\[
\text{d}_{\text{inter}} = \sum_{m=1}^{M} \sum_{n=1}^{N} \text{min}(\hat{s}^i_{m,n,k}, s^i_{m,n,k})
\]

\[
\text{d}_{\text{union}} = \sum_{m=1}^{M} \sum_{n=1}^{N} \text{max}(\hat{s}^i_{m,n,k}, s^i_{m,n,k})
\]

Although the set \( \Xi \) contain \( \hat{K}_i! \) elements, the permutation \( \xi^i \) can still be computed efficiently by formulating the computation as a linear sum assignment problem.

**F1 Score (F1)**

F1 score can also be used to assess the performance of amodal segmentation like IOU, and is computed in a similar way. The meanings of \( \hat{s}^i, s^i, \xi, \) and \( \Xi \) as well as the computations of \( d_{\text{inter}} \) and \( d_{\text{union}} \) are identical to the ones in the descriptions of IOU. F1 is computed by

\[
F1 = \frac{1}{I} \sum_{i=1}^{I} \frac{K_i}{K_i \cdot \text{d}_{\text{inter}} + \text{d}_{\text{union}}}
\]

**Object Counting Accuracy (OCA)**

\( \hat{K}_i \) and \( K_i \) denote the ground truth number and the estimated number of objects in the \( i \)th visual scene of the test set. Let \( \delta \) denote the Kronecker delta function. OCA is computed by

\[
\text{OCA} = \frac{1}{I} \sum_{i=1}^{I} \delta_{\hat{K}_i,K_i}
\]

**Object Ordering Accuracy (OOA)**

Let \( \hat{t}^i_{m,k_1,k_2} \in \{0, 1\} \) and \( t^i_{m,k_1,k_2} \in \{0, 1\} \) denote the ground truth and estimated pairwise orderings of the \( k_1 \)th and \( k_2 \)th objects in the \( m \)th viewpoint of the \( i \)th image. The correspondences between the ground truth and estimated indexes of objects are determined based on the permutation of indexes \( \xi^i \) as described in the computation of IOU. Because the relative ordering of objects is hard to estimate if these objects do not overlap, OOA is computed by

\[
\text{OOA} = \frac{1}{I} \sum_{i=1}^{I} \frac{\hat{K}_i - 1}{\hat{K}_i} \sum_{k_1=1}^{K} \sum_{k_2=k_1+1}^{K} w_{\hat{m}k_1k_2} \sum_{k_1=1}^{K} \sum_{k_2=k_1+1}^{K} w_{mk_1k_2}
\]

where \( w_{\hat{m}k_1k_2} \) is the weight computed based on the ground truth complete shapes of objects \( \hat{s}^i \)

\[
w_{\hat{m}k_1k_2} = \sum_{m=1}^{N} \hat{s}^i_{m,n,k_1} \cdot s^i_{m,n,k_2}
\]

\( w_{mk_1k_2} \) measures the overlapped area of the ground truth shapes of the \( k_1 \)th and \( k_2 \)th objects. The more the two objects overlap, the easier it is to determine the relative ordering of these objects, and thus the more important it is for the model to estimate the relative ordering correctly.
Details of Datasets

Configurations of the datasets used in the multi-viewpoint and single-viewpoint learning settings are presented in Table 3 and Table 4 respectively. Explanations of the configurations are described in the captions of the tables. The CLEVR-M and CLEVR datasets are generated based on the official code provided by (Johnson et al. 2017). Images in the multi-viewpoint CLEVR-M dataset are generated with size $108 \times 80$ and cropped to size $64 \times 64$ at locations $10$ (up), $74$ (down), $22$ (left), and $86$ (right). Code is modified to skip the check of object visibility because the observations of objects vary as viewpoints change. Images in the single-viewpoint CLEVR dataset are generated with size $214 \times 160$ and cropped to size $128 \times 128$ at locations $19$ (up), $147$ (down), $43$ (left), and $171$ (right). Code is modified to ensure that at least $128$ pixels of each object is visible after cropping (instead of before cropping). In images of the dSprites datasets, the colors of backgrounds are sampled uniformly from grayscale RGB colors, and the colors of the objects provided by (Matthey et al. 2017) are sampled uniformly from RGB colors with the constraint that the $l_2$ distance between the colors of object and background is at least 0.5 (the range of each channel is $[0, 1]$). As for the Abstract dataset, the background and $10$ objects in the Abstract Scene dataset (Zitnick and Parikh 2013) are selected to synthesize images. The colors of objects and background are both randomly permuted in HSV space (the range of each channel is $[0, 1]$). The H channel of all the pixels of the same object or background is multiplied with a different random number sampled from $\mathcal{U}(-0.1, 0.1)$. And the S/V channel of all the pixels of the same object or background is multiplied with the same random number sampled from $\mathcal{U}(0.9, 1)$. If not explicitly mentioned, models are tested on the Test 1 splits and the corresponding experimental results are reported.

**Choices of Hyperparameters**

**Multi-Viewpoint Learning**

**Proposed Method** In the generative model, the standard deviation $\sigma_{l}$ of the likelihood function is chosen to be 0.2. The maximum number $K$ of objects that may appear in the
visual scene is set to 7 during training. The respective dimensions of latent variables \(z_{\text{view}}\), \(z_{\text{attr}}\), and \(z_{\text{att}}\) with \(1 \leq k \leq K\) are 4, 8, and 64. \(\alpha\) is 4.5 and \(\lambda\) is 0.5. In the inference, the dimensions of \(D_{\text{view}}\) and \(D_{\text{att}}\) of intermediate variables \(y_{\text{view}}\) and \(y_{\text{att}}\) are 8 and 128 respectively. \(D_{\text{key}}\) is 64, \(D_{\text{val}}\) is 136, and \(T\) is 3. In the learning, the batch size is chosen to be 32. The initial learning rate is \(4 \times 10^{-4}\), and is decayed exponentially with a factor 0.5 every 50,000 steps. In the first 10,000 training steps, the learning rate is multiplied by a factor that is increased linearly from 0 to 1. We have found that the optimization of neural networks with randomly initialized weights tends to get stuck into undesired local optima. To solve this problem, a better initialization of weights is obtained by using only one viewpoint per visual scene to train neural networks in the first 10,000 steps. On CLEVR-M1 and CLEVR-M2, the proposed method is trained from scratch for 100,000 steps, even though the relatively large range of azimuth (i.e., \([0, \pi]\)) makes the fully-unsupervised learning difficult. On CLEVR-M3/CLEVR-M4 in which the azimuth is sampled from the full range \([0, 2\pi]\), we have found it beneficial to adopt a curriculum learning strategy that first pretrains the model on the simpler CLEVR-M1/CLEVR-M2 for 100,000 steps and then continues to train on CLEVR-M3/CLEVR-M4 for 100,000 steps. The choices of neural networks in both generative model and variational inference are described below. Instead of adopting a superior but more time-consuming method such as grid search, we manually choose the hyperparameters of neural networks based on experience.

- \(f_{\text{shp}}\) and \(f_{\text{pc}}\) in the generative model are implemented as one convolutional neural network (CNN). The outputs of the CNN are split in the channel dimension into 1 and 3 for \(f_{\text{shp}}\) and \(f_{\text{pc}}\), respectively.
  - Fully Connected, 4096 ReLU
  - Fully Connected, 4096 ReLU
  - Fully Connected, \(8 \times 8 \times 128\) ReLU
  - 2x nearest-neighbor upsample; 5 × 5 Conv, 128 ReLU
  - 5 × 5 Conv, 64 ReLU
  - 2x nearest-neighbor upsample; 5 × 5 Conv, 64 ReLU
  - 5 × 5 Conv, 32 ReLU
  - 2x nearest-neighbor upsample; 5 × 5 Conv, 32 ReLU
  - 3 × 3 Conv, 1 + 3 Linear

- \(f_{\text{bck}}\) in the generative model is a CNN.
  - Fully Connected, 512 ReLU
  - Fully Connected, 512 ReLU
  - Fully Connected, \(4 \times 4 \times 16\) ReLU
  - 4x nearest-neighbor upsample; 5 × 5 Conv, 16 ReLU
  - 5 × 5 Conv, 16 ReLU
  - 4x nearest-neighbor upsample; 5 × 5 Conv, 16 ReLU
  - 3 × 3 Conv, 3 Linear

- \(f_{\text{obj}}\) in the generative model is a multi-layer perceptron (MLP).
  - Fully Connected, 512 ReLU
  - Fully Connected, 512 ReLU

- \(g_{\text{shp}}\) in the variational inference is a CNN augmented with positional embedding.
  - 5 × 5 Conv, 64 ReLU
  - 5 × 5 Conv, 64 ReLU
  - 5 × 5 Conv, 64 ReLU
  - 5 × 5 Conv, 64 ReLU; Positional Embedding
  - Layer Norm; Fully Connected, 64 ReLU
  - Fully Connected, 64 Linear

- \(g_{\text{bck}}\) in the variational inference is a linear layer with layer Normalization.
  - Layer Norm; Fully Connected, 64 Linear

- \(g_{\text{obj}}\) in the variational inference is a linear layer with layer normalization.
  - Layer Norm; Fully Connected, 64 Linear

- \(g_{\text{d}}\) in the variational inference is a gated recurrent unit (GRU) followed by a residual MLP with layer normalization, which independently updates \(y_{\text{full}}\) for each \(m\) and \(k\). Information of different viewpoints are integrated in the two average operations following \(g_{\text{opt}}\). It is possible to apply a more complex and powerful neural network such as a graph neural network (GNN) to integrate information of different viewpoints earlier, and we leave the investigation in future work.
  - GRU, 136 Tanh
  - Layer Norm; Fully Connected, 128 ReLU
  - Fully Connected, \(8 + 128\) Linear

- \(g_{\text{d}}\) in the variational inference is a MLP.
  - Fully Connected, 512 ReLU
  - Fully Connected, 512 ReLU
  - Fully Connected, \(8 + 8\) Linear

- \(g_{\text{obj}}\) in the variational inference is an MLP.
  - Fully Connected, 512 ReLU
  - Fully Connected, 512 ReLU
  - Fully Connected, \(64 + 64 + 2 + 1\) Linear

- \(g_{\text{view}}\) in the variational inference is an MLP.
  - Fully Connected, 512 ReLU
  - Fully Connected, 512 ReLU
  - Fully Connected, \(4 + 4\) Linear

- The neural network used by NVIL is a CNN.
  - \(3 \times 3\) Conv, 16 ReLU
  - \(3 \times 3\) Conv, stride 2, 16 ReLU
  - \(3 \times 3\) Conv, 32 ReLU
  - \(3 \times 3\) Conv, stride 2, 32 ReLU
  - \(3 \times 3\) Conv, 64 ReLU
  - \(3 \times 3\) Conv, stride 2, 64 ReLU
  - Fully Connected, 256 ReLU
  - Fully Connected, 1 Linear
Baseline Method  The baseline method which is derived from the proposed method uses the same set of hyperparameters, and differs from the proposed method in two aspects. In the generative model, the viewpoint-independent latent variables \( z^{\text{attr}}_k, \rho_k, \) and \( z^{\text{prs}}_k \) are replaced with viewpoint-dependent versions \( z^{\text{attr}}_{m,k}, \rho_{m,k}, \) and \( z^{\text{prs}}_{m,k} \). In the variational inference, the lines 3, 4, 7, 11, 12, 18, 19, 20 in Algorithm 1 are replaced with the following expressions.

\[
\begin{align*}
\text{line 3}: & \quad y^{\text{view}}_{m,k} \sim \mathcal{N}(\mu^{\text{view}}, \text{diag}(\sigma^{\text{view}})) \\
\text{line 4}: & \quad y^{\text{attr}}_{m,k} \sim \mathcal{N}(\mu^{\text{attr}}, \text{diag}(\sigma^{\text{attr}})) \\
\text{line 7}: & \quad y^{\text{full}}_{m,k} \leftarrow [y^{\text{view}}_{m,k}, y^{\text{attr}}_{m,k}] \\
\text{line 11}: & \quad y^{\text{view}}_{m,k} \leftarrow v^{\text{view}}_{m,k} \\
\text{line 12}: & \quad y^{\text{attr}}_{m,k} \leftarrow v^{\text{attr}}_{m,k} \\
\text{line 18}: & \quad \mu^{\text{attr}}_{m,0}, \sigma^{\text{attr}}_{m,0} \leftarrow g_{\text{fck}}(y^{\text{attr}}_{m,0}) \\
\text{line 19}: & \quad \mu^{\text{attr}}_{m,k}, \sigma^{\text{attr}}_{m,k}, \tau_{m,k}, \kappa_{m,k} \leftarrow g_{\text{obj}}(y^{\text{attr}}_{m,k}) \\
\text{line 20}: & \quad \mu^{\text{view}}_{m,k}, \sigma^{\text{view}}_{m,k} \leftarrow g_{\text{view}}(y^{\text{view}}_{m,k})
\end{align*}
\]

MulMON  MulMON is trained with the default hyperparameters described in the “scripts/train_clevr_parallel.sh” file of the official code repository\footnote{https://github.com/NanboLi/MulMON} except: 1) the number of training steps is 600,000; 2) the number of viewpoints for inference is sampled from \( n \sim \mathcal{U}(1, 3) \) and the number of viewpoints for query is \( 4 - n \); 3) the number of slots \( K+1 \) is 8 during training.

Single-Viewpoint Learning

Proposed Method  In the generative model, the standard deviation \( \sigma_{\text{obj}} \) of the likelihood function is 0.2. The maximum number \( K \) of objects that may appear in the visual scene is 6, 5, and 7 on the dSprites, Abstract, and CLEVR datasets, respectively. The respective dimensionalities of \( z^{\text{view}}_m, z^{\text{obj}}_m, \) and \( z^{\text{attr}}_k \) with \( 1 \leq k \leq K \), as well as the hyperparameter \( \alpha \) are \( 1/11, 4/11, 32/32/64 \), and \( 3,5/3, 0/4,5/4 \) on the dSprites/Abstract/CLEVR dataset. \( \lambda \) is chosen to be 0.5. In the inference, the dimensionalities \( D_{\text{view}}, D_{\text{attr}}, D_{\text{key}}, \) and \( D_{\text{val}} \) are 1, 64, 64, and 65, respectively. \( T \) is chosen to be 3. In the learning, the batch size is 64 and the number of training steps is 500,000. The initial learning rate is \( 4 \times 10^{-4} \), and is decayed exponentially with a factor 0.5 every 100,000 steps. In the first 10,000 training steps, the learning rate is multiplied by a factor that is increased linearly from 0 to 1. Hyperparameters of neural networks are described below.

- \( f_{\text{fck}} \) and \( f_{\text{fpc}} \) on the dSprites and Abstract datasets.
  - \( 64 \times 64 \) spatial broadcast; Positional Embedding
  - \( 5 \times 5 \) Conv, 32 ReLU
  - \( 5 \times 5 \) Conv, 32 ReLU
  - \( 5 \times 5 \) Conv, 32 ReLU
  - \( 3 \times 3 \) Conv, 1 + 3 Linear
- \( f_{\text{fck}} \) and \( f_{\text{fpc}} \) on the CLEVR dataset.
  - \( 8 \times 8 \) spatial broadcast; Positional Embedding
  - \( 5 \times 5 \) Trans Conv, stride 2, 64 ReLU
  - \( 5 \times 5 \) Trans Conv, stride 2, 64 ReLU
  - \( 5 \times 5 \) Trans Conv, stride 2, 64 ReLU
  - \( 5 \times 5 \) Conv, 64 ReLU
  - \( 3 \times 3 \) Conv, 1 + 3 Linear
- \( f_{\text{fck}} \) on the dSprites and Abstract datasets.
  - Fully Connected, 512 ReLU
  - Fully Connected, 512 ReLU
  - Fully Connected, 3 Linear
- \( g_{\text{fck}} \) on the dSprites and Abstract datasets.
  - \( 64 \times 64 \) spatial broadcast; Positional Embedding
  - \( 5 \times 5 \) Conv, 8 ReLU
  - \( 5 \times 5 \) Conv, 8 ReLU
  - \( 3 \times 3 \) Conv, 3 Linear
- \( f_{\text{ord}} \) on all the datasets.
  - Fully Connected, 512 ReLU
  - Fully Connected, 512 ReLU
  - Fully Connected, 1 Linear
- \( g_{\text{ord}} \) on the dSprites and Abstract datasets.
  - \( 5 \times 5 \) Conv, 32 ReLU
  - \( 5 \times 5 \) Conv, 32 ReLU
  - \( 5 \times 5 \) Conv, 32 ReLU
  - \( 5 \times 5 \) Conv, 32 ReLU
  - \( 5 \times 5 \) Conv, 32 ReLU
  - Layer Norm; Fully Connected, 32 ReLU
  - Fully Connected, 32 Linear
- \( g_{\text{fck}} \) on the CLEVR dataset.
  - \( 5 \times 5 \) Conv, 64 ReLU
  - \( 5 \times 5 \) Conv, 64 ReLU
  - \( 5 \times 5 \) Conv, 64 ReLU
  - \( 5 \times 5 \) Conv, 64 ReLU
  - \( 5 \times 5 \) Conv, 64 ReLU
  - Layer Norm; Fully Connected, 64 ReLU
  - Fully Connected, 64 Linear
- \( g_{\text{key}} \) on all the datasets.
  - Layer Norm; Fully Connected, 64 Linear
- \( g_{\text{qry}} \) on all the datasets.
  - Layer Norm; Fully Connected, 64 Linear
- \( g_{\text{val}} \) on all the datasets.
  - Layer Norm; Fully Connected, 65 Linear
- \( g_{\text{fck}} \) on all the datasets.
  - GRU, 65 Tanh
  - Layer Norm; Fully Connected, 128 ReLU
  - Fully Connected, 1 + 64 Linear
- \( g_{\text{ord}} \) on the dSprites and Abstract datasets.
  - Fully Connected, 512 ReLU
  - Fully Connected, 512 ReLU
Slot Attention  Slot Attention is trained with the default hyperparameters described in the official code repository with the exception: 1) the number of slots $K + 1$ during training is 7, 6, and 8 on the dSprites, Abstract, and CLEVR datasets; 2) the hyperparameters of neural networks on the dSprites and Abstract datasets are chosen to be same as the default hyperparameters used for the Multi-dSprites dataset by Slot Attention, i.e., Tables 4 and 6 in the supplementary material of [Locatello et al. 2020].

GMIOO  GMIOO is trained with the default hyperparameters described in the “experiments/config.yaml” file of the official code repository with the exception: 1) the number of training steps is 200,000 and the batch size is 64; 2) the upper bound $K$ of the number of objects during inference and the hyperparameter $\alpha$ are 6/5/7 and 3.5/3.0/4.5 on the dSprites/Abstract/CLEVR dataset; 3) the dimensionality of the background latent variable is 4 on the Abstract and dSprites datasets; 4) the mean parameter of the prior distribution of bounding box scale is $-0.5$ on the dSprites and CLEVR datasets; 5) the standard deviation parameter of the prior distribution of bounding box scale is 0.5 on all the datasets; 6) the glimpse size is 32 on the dSprites and Abstract datasets and is 64 on the CLEVR dataset; 7) some neural networks use different hyperparameters, which are described below.

- The decoder of appearance on the Abstract dataset.
  - Fully Connected, 256 ReLU
  - Fully Connected, 8 $\times$ 8 $\times$ 32 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 32 ReLU
  - 3 $\times$ 3 Conv, 16 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 16 ReLU
  - 3 $\times$ 3 Conv, 3 Linear

- The decoder of appearance on the CLEVR dataset.
  - Fully Connected, 256 ReLU
  - Fully Connected, 8 $\times$ 8 $\times$ 32 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 32 ReLU
  - 3 $\times$ 3 Conv, 16 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 16 ReLU
  - 3 $\times$ 3 Conv, 8 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 8 ReLU
  - 3 $\times$ 3 Conv, 3 Linear

- The decoder of appearance on the Abstract dataset.
  - Fully Connected, 256 ReLU
  - Fully Connected, 512 ReLU
  - Fully Connected, 512 ReLU
  - Fully Connected, 32 + 32 + 2 + 1 Linear

- The decoder of background on the CLEVR dataset.
  - Fully Connected, 2 $\times$ 2 $\times$ 8 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 8 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 8 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 8 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 8 ReLU
  - 3 $\times$ 3 Conv, 1 Linear

- The decoder of shape on the dSprites and Abstract datasets.
  - Fully Connected, 256 ReLU
  - Fully Connected, 8 $\times$ 8 $\times$ 32 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 32 ReLU
  - 3 $\times$ 3 Conv, 16 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 16 ReLU
  - 3 $\times$ 3 Conv, 1 Linear

- The convolutional parts of the neural networks used to initialize and update latent variables of background on the CLEVR dataset.
  - 3 $\times$ 3 Conv, 4 ReLU
  - 3 $\times$ 3 Conv, stride 2, 4 ReLU
  - 3 $\times$ 3 Conv, 8 ReLU
  - 3 $\times$ 3 Conv, 16 ReLU
  - 3 $\times$ 3 Conv, stride 2, 8 ReLU
  - 3 $\times$ 3 Conv, 16 ReLU
  - 3 $\times$ 3 Conv, stride 2, 16 ReLU
  - 3 $\times$ 3 Conv, 8 ReLU
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 8 ReLU
  - 3 $\times$ 3 Conv, 3 Linear
  - 2x bilinear upsample; 3 $\times$ 3 Conv, 3 Linear
• The convolutional parts of the neural networks used to initialize and update latent variables of objects on the CLEVR dataset.
  – 3 × 3 Conv, 8 ReLU
  – 3 × 3 Conv, stride 2, 8 ReLU
  – 3 × 3 Conv, 16 ReLU
  – 3 × 3 Conv, stride 2, 16 ReLU
  – 3 × 3 Conv, 32 ReLU
  – 3 × 3 Conv, stride 2, 32 ReLU

SPACE SPACE is trained with the default hyperparameters described in the “src/configs/3d_room_small.yaml” file of the official code repository, except: 1) the number of training steps is 200,000; 2) the number of background components is 1 and “CompDecoder” is used as the decoder of background on all the datasets; 3) the dimensionality of the background latent variable is 4 on the Abstract dataset; 4) the mean parameter of the prior distribution of bounding box scale is decreased from 0 to −0.5 on the CLEVR and dSprites datasets, and from 0.5 to 0 on the Abstract dataset; 5) the standard deviation parameter of the prior distribution of bounding box scale is 0.5 on all the datasets; 6) the standard deviations of foreground and background on the Abstract dataset are 0.2 and 0.1, respectively; 7) the glimpse size is 64 on the CLEVR dataset; 8) some neural networks use different hyperparameters, which are described below.

• The convolutional part of the glimpse encoder on the CLEVR dataset.
  – 3 × 3 Conv, 16 CELU; Group Norm
  – 4 × 4 Conv, stride 2, 32 CELU; Group Norm
  – 3 × 3 Conv, 32 CELU; Group Norm
  – 4 × 4 Conv, stride 2, 64 CELU; Group Norm
  – 4 × 4 Conv, stride 2, 128 CELU; Group Norm
  – 8 × 8 Conv, 256 CELU; Group Norm

• The convolutional part of the glimpse decoder on the CLEVR dataset.
  – 1 × 1 Conv, 256 CELU; Group Norm
  – 1 × 1 Conv, 128 × 4 × 4 CELU
  – 4x pixel shuffle; Group Norm
  – 3 × 3 Conv, 128 CELU; Group Norm
  – 1 × 1 Conv, 128 × 2 × 2 CELU
  – 2x pixel shuffle; Group Norm
  – 3 × 3 Conv, 128 CELU; Group Norm
  – 1 × 1 Conv, 64 × 2 × 2 CELU
  – 2x pixel shuffle; Group Norm
  – 3 × 3 Conv, 64 CELU; Group Norm
  – 1 × 1 Conv, 32 × 2 × 2 CELU
  – 2x pixel shuffle; Group Norm
  – 3 × 3 Conv, 32 CELU; Group Norm
  – 1 × 1 Conv, 16 × 2 × 2 CELU
  – 2x pixel shuffle; Group Norm
  – 3 × 3 Conv, 16 CELU; Group Norm

Extra Experimental Results

Computing Infrastructure

Experiments are conducted on a server with Intel Xeon CPU E5-2678 v3 CPUs, NVIDIA GeForce RTX 2080 Ti GPUs, 256Gi memory, and Ubuntu 20.04 operating system. The code is developed based on the PyTorch framework [Paszke et al. 2019] with version 1.8. In all the experiments, the random seeds are sampled randomly using the built-in python function “random.randint(0, 0xfffffff)”. On the CLEVR-M, Abstract, and dSprites datasets, the proposed method can be trained and tested with one NVIDIA GeForce RTX 2080 Ti GPU. On the CLEVR dataset, at least two NVIDIA GeForce RTX 2080 Ti GPUs are needed.

Multi-Viewpoint Learning

Scene Decomposition The qualitative comparison of the baseline method, MulMON, and the proposed method on the CLEVR-M1 to CLEVR-M4 datasets is shown in Figures 4, 5, 6, and 7 respectively. All the methods tend to estimate shadows as parts of objects. The baseline method is not able to accurately identify the same object across viewpoints, while MulMON and the proposed method are able to achieve object constancy relatively well. The baseline method and the proposed method explicitly model the varying number of objects and distinguish background from objects, while MulMON does not have such abilities. However, it is still possible to estimate the number of objects based on the scene decomposition results of MulMON, in a reasonable though heuristic way. More specifically, let \( r \in \{0, 1\}^{M \times N \times (K+1)} \) be the estimated pixel-wise partition of \( K+1 \) slots in \( M \) viewpoints. Whether the visual entity represented by the \( k \)th slot is included in the visual scene can be computed by \( \max_m \max_n r_{m, n, k} \), and the estimated number of objects \( \hat{K} \) is

\[
\hat{K} = \sum_{k=0}^{K} \left( \max_m \max_n r_{m, n, k} \right) - 1.
\]

The proposed method significantly outperforms the baseline that randomly guesses the identities of objects, in all the evaluation metrics except ARI-A. The possible reason is that the baseline is derived from the proposed method and outputs similar background region estimations, which dominates the computation of ARI-A under the circumstance that shadows are incorrectly estimated as objects. Compared with MulMON, the ARI-A and AMI-A scores of the proposed method are slightly lower, and the rest scores are competitive or slightly better.

Generalizability The quantitative results of generalizing the trained models to different number of objects and different number of viewpoints are presented in Tables 5, 6, 7, 8, 9, 10 and 11. Generally speaking, both MulMON and the proposed method perform reasonably well when visual scenes contain more objects and the number of viewpoints varies compared with the ones used for training. MulMON generalizes better than the proposed method when the number of objects is increased (Tables 6, 8, 9, and 11), possibly because MulMON adopts a more powerful but time-consuming architecture.
Viewpoint Estimation Both the viewpoint representations and viewpoint-independent object-centric representations are inferred simultaneously during training, and the model is able to maintain the consistency between these two parts (e.g., using the same global coordinate system to represent both parts). When testing the models, it is possible to only estimate the viewpoints of images, under the condition that the viewpoint-independent attributes of objects and background are known. More specifically, given the intermediate variables $y_{0:k}^{\text{attr}}$ that fully characterize the approximate posteriors of object-centric representations, the proposed method is able to infer the corresponding viewpoint representations of different observations of the same visual scene, while achieving the consistency between the inferred viewpoint representations and the given object-centric representations. This kind of inference can be derived from Algorithm 1 by initializing $y_{0:k}^{\text{attr}}$ with the given representation instead of sampling from a normal distribution in line 4, and not executing the update operation in line 12. The viewpoints of $M_2 = 4$ images are estimated given the intermediate variables $y_{0:k}^{\text{attr}}$ that are estimated on $M_1 = 1, 2, 4$ images. The viewpoints of the $M_1 + M_2$ images are all different. Experimental results on the Test 1 and Test 2 splits are shown in Tables 12 and 13 respectively. Generally speaking, the estimated viewpoint representations are consistent with the given object-centric representations, and the scene decomposition performance increases as more viewpoints are used to extract object-centric representations.

Single-Viewpoint Learning

The qualitative comparison of Slot Attention, GMIOO, SPACE, and the proposed method on the dSprites, Abstract, and CLEVER datasets is shown in Figures 5, 9, and 10 respectively. Slot Attention does not distinguish background from objects when modeling visual scenes. Therefore, there is no nature way to determine which slot corresponds to background. The trained model is able to use one random slot to represent background on the Abstract dataset, but fails to do so on the dSprites and CLEVR datasets. GMIOO and the proposed method work well even when objects are heavily occluded on the Abstract dataset. However, GMIOO tends to group nearby small objects as a single object on the CLEVR dataset. SPACE performs well on the CLEVR dataset, but tends to group multiple nearby or occluded objects as one object on the dSprites and Abstract datasets. On the dSprites dataset, GMIOO in general achieves the best performance, possibly because the inferred latent representations are iteratively refined based on both the observed and reconstructed images, which is beneficial when objects are rich in diversity. The performance of the proposed method on this dataset is only slightly lower than GMIOO. On the Abstract dataset, the proposed method in general performs best. The ARI-A and AMI-A scores achieved by Slot Attention is significantly higher than the rest methods, mainly because objects and background are not separately modeled, which leads to better estimations in the boundary regions of objects. GMIOO, SPACE and the proposed method tend to consider the background pixels in the boundary regions as object pixels, possibly because these pixels can be better reconstructed using the object decoders that have higher model capacities than the background decoders. On the CLEVR dataset, SPACE in general achieves the best results. The possible reason is that SPACE adopts a two-stage framework to infer latent variables, and the attention mechanism with the spatial invariant property in the first stage acts as a strong inductive bias to guide the discovery of objects and to correctly estimate shadows of objects as background. The proposed method can better distinguish different objects from one another than SPACE, but achieves lower ARI-A, AMI-A, IoU, and F1 scores mainly because of the incorrect estimation of shadows. The performance of generalizing the trained models to images containing more numbers of objects is shown in Table 14. The proposed method achieves reasonable performance. GMIOO and SPACE in general generalize best, possibly because these methods adopt two-stage frameworks that first estimate bounding boxes of objects and then infer latent variables based on the cropped images in the bounding boxes. More specifically, the estimations of bounding boxes are more robust to the increase of number of objects and latent variables are easier to estimate given bounding boxes, which make GMIOO and SPACE more advantageous on images containing more objects than the ones used for training.

consuming inference method, i.e., iterative amortized inference, which iteratively refines parameters of the variational distribution based on the gradients of loss function, while the proposed method does not exploit the information of gradient during inference. The baseline method performs best when the number of viewpoints is 1 when testing (Tables 5 and 6). The possible reason is that the baseline method treats images observed from multiple viewpoints of the same visual scene as images observed from a single viewpoint of different visual scenes, and thus the model does not need to learn how to maintain identities of objects across viewpoints and can better focus on acquiring information of the visual scene from a single viewpoint. When visual scenes are observed from more than one viewpoint, the baseline method does not perform well, while MulMON and the proposed method are both effective.

Slot Attention does not distinguish background from objects when modeling visual scenes. Therefore, there is no nature way to determine which slot corresponds to background. The trained model is able to use one random slot to represent background on the Abstract dataset, but fails to do so on the dSprites and CLEVR datasets. GMIOO and the proposed method work well even when objects are heavily occluded on the Abstract dataset. However, GMIOO tends to group nearby small objects as a single object on the CLEVR dataset. SPACE performs well on the CLEVR dataset, but tends to group multiple nearby or occluded objects as one object on the dSprites and Abstract datasets. On the dSprites dataset, GMIOO in general achieves the best performance, possibly because the inferred latent representations are iteratively refined based on both the observed and reconstructed images, which is beneficial when objects are rich in diversity. The performance of the proposed method on this dataset is only slightly lower than GMIOO. On the Abstract dataset, the proposed method in general performs best. The ARI-A and AMI-A scores achieved by Slot Attention is significantly higher than the rest methods, mainly because objects and background are not separately modeled, which leads to better estimations in the boundary regions of objects. GMIOO, SPACE and the proposed method tend to consider the background pixels in the boundary regions as object pixels, possibly because these pixels can be better reconstructed using the object decoders that have higher model capacities than the background decoders. On the CLEVR dataset, SPACE in general achieves the best results. The possible reason is that SPACE adopts a two-stage framework to infer latent variables, and the attention mechanism with the spatial invariant property in the first stage acts as a strong inductive bias to guide the discovery of objects and to correctly estimate shadows of objects as background. The proposed method can better distinguish different objects from one another than SPACE, but achieves lower ARI-A, AMI-A, IoU, and F1 scores mainly because of the incorrect estimation of shadows. The performance of generalizing the trained models to images containing more numbers of objects is shown in Table 14. The proposed method achieves reasonable performance. GMIOO and SPACE in general generalize best, possibly because these methods adopt two-stage frameworks that first estimate bounding boxes of objects and then infer latent variables based on the cropped images in the bounding boxes. More specifically, the estimations of bounding boxes are more robust to the increase of number of objects and latent variables are easier to estimate given bounding boxes, which make GMIOO and SPACE more advantageous on images containing more objects than the ones used for training.
| Dataset | Method | ARI-A | AMI-A | ARI-O | AMI-O | IoU | F1  | OCA | OOA |
|---------|--------|-------|-------|-------|-------|-----|-----|-----|-----|
| CLEV-R-M1 Baseline | 0.510±3e-3 | 0.502±2e-3 | 0.957±4e-3 | 0.954±3e-3 | 0.437±3e-3 | 0.585±3e-3 | 0.688±1e-2 | 0.853±5e-2 |
| MulMON Proposed | 0.605±6e-3 | 0.555±3e-3 | 0.919±4e-3 | 0.913±2e-3 | N/A | N/A | 0.410±4e-2 | N/A |
| CLEV-R-M2 Baseline | 0.508±2e-3 | 0.495±2e-3 | 0.959±7e-3 | 0.954±6e-3 | 0.434±2e-3 | 0.583±3e-3 | 0.666±3e-2 | 0.872±3e-2 |
| MulMON Proposed | 0.596±2e-3 | 0.544±1e-3 | 0.931±4e-3 | 0.919±4e-3 | N/A | N/A | 0.462±6e-2 | N/A |
| CLEV-R-M3 Baseline | 0.545±2e-3 | 0.522±2e-3 | 0.956±2e-3 | 0.951±2e-3 | 0.468±3e-3 | 0.615±4e-3 | 0.708±4e-2 | 0.887±4e-2 |
| MulMON Proposed | 0.581±9e-3 | 0.538±4e-3 | 0.890±7e-3 | 0.885±6e-3 | N/A | N/A | 0.382±5e-2 | N/A |
| CLEV-R-M4 Baseline | 0.532±2e-3 | 0.511±2e-3 | 0.959±6e-3 | 0.961±4e-3 | 0.460±4e-3 | 0.612±5e-3 | 0.760±4e-2 | 0.859±5e-2 |
| MulMON Proposed | 0.643±3e-3 | 0.579±3e-3 | 0.917±6e-3 | 0.911±4e-3 | N/A | N/A | 0.426±2e-2 | N/A |

Table 5: Comparison of scene decomposition performance when learning from multiple viewpoints. All the methods are trained with $M = 4$ and $K = 7$, and tested on the Test 1 splits with $M = 1$ and $K = 7$.

| Dataset | Method | ARI-A | AMI-A | ARI-O | AMI-O | IoU | F1  | OCA | OOA |
|---------|--------|-------|-------|-------|-------|-----|-----|-----|-----|
| CLEV-R-M1 Baseline | 0.358±1e-3 | 0.488±2e-3 | 0.912±5e-3 | 0.920±3e-3 | 0.356±5e-3 | 0.493±6e-3 | 0.300±3e-2 | 0.877±8e-3 |
| MulMON Proposed | 0.551±5e-3 | 0.568±3e-3 | 0.888±4e-3 | 0.887±2e-3 | N/A | N/A | 0.412±3e-2 | N/A |
| CLEV-R-M2 Baseline | 0.367±3e-3 | 0.502±2e-3 | 0.914±1e-3 | 0.921±2e-3 | 0.374±4e-3 | 0.513±5e-3 | 0.260±2e-2 | 0.855±2e-2 |
| MulMON Proposed | 0.516±5e-3 | 0.555±3e-3 | 0.877±5e-3 | 0.881±4e-3 | N/A | N/A | 0.286±2e-2 | N/A |
| CLEV-R-M3 Baseline | 0.325±3e-3 | 0.476±1e-3 | 0.913±3e-3 | 0.919±3e-3 | 0.355±2e-3 | 0.492±3e-3 | 0.280±6e-2 | 0.856±3e-2 |
| MulMON Proposed | 0.513±3e-3 | 0.533±4e-3 | 0.820±1e-2 | 0.840±7e-3 | N/A | N/A | 0.200±3e-2 | N/A |
| CLEV-R-M4 Baseline | 0.358±2e-3 | 0.495±1e-3 | 0.913±4e-3 | 0.923±3e-3 | 0.374±3e-3 | 0.513±4e-3 | 0.294±4e-2 | 0.877±2e-2 |
| MulMON Proposed | 0.534±5e-3 | 0.556±3e-3 | 0.856±4e-3 | 0.867±3e-3 | N/A | N/A | 0.242±3e-2 | N/A |

Table 6: Comparison of scene decomposition performance when learning from multiple viewpoints. All the methods are trained with $M = 4$ and $K = 7$, and tested on the Test 2 splits with $M = 1$ and $K = 11$.

| Dataset | Method | ARI-A | AMI-A | ARI-O | AMI-O | IoU | F1  | OCA | OOA |
|---------|--------|-------|-------|-------|-------|-----|-----|-----|-----|
| CLEV-R-M1 Baseline | 0.510±2e-3 | 0.431±2e-3 | 0.540±1e-2 | 0.669±7e-3 | 0.248±3e-3 | 0.376±4e-3 | 0.108±3e-2 | 0.721±7e-2 |
| MulMON Proposed | 0.609±2e-3 | 0.558±2e-3 | 0.930±7e-3 | 0.920±4e-3 | N/A | N/A | 0.428±3e-2 | N/A |
| CLEV-R-M2 Baseline | 0.502±5e-4 | 0.422±3e-4 | 0.529±7e-3 | 0.665±2e-3 | 0.240±4e-3 | 0.365±5e-3 | 0.106±2e-2 | 0.666±4e-2 |
| MulMON Proposed | 0.596±1e-3 | 0.548±7e-4 | 0.944±3e-3 | 0.931±2e-3 | N/A | N/A | 0.512±2e-2 | N/A |
| CLEV-R-M3 Baseline | 0.535±1e-3 | 0.445±2e-3 | 0.538±1e-2 | 0.668±7e-3 | 0.258±3e-3 | 0.389±3e-3 | 0.110±2e-2 | 0.665±3e-2 |
| MulMON Proposed | 0.594±4e-3 | 0.552±2e-3 | 0.924±7e-3 | 0.914±3e-3 | N/A | N/A | 0.378±5e-2 | N/A |
| CLEV-R-M4 Baseline | 0.524±2e-3 | 0.354±3e-2 | 0.534±3e-2 | 0.673±2e-2 | 0.249±6e-3 | 0.379±6e-3 | 0.090±3e-2 | 0.680±5e-2 |
| MulMON Proposed | 0.645±8e-4 | 0.582±7e-4 | 0.936±2e-3 | 0.926±2e-3 | N/A | N/A | 0.462±6e-2 | N/A |

Table 7: Comparison of scene decomposition performance when learning from multiple viewpoints. All the methods are trained with $M = 4$ and $K = 7$, and tested on the Test 1 splits with $M = 2$ and $K = 7$. 
Table 8: Comparison of scene decomposition performance when learning from multiple viewpoints. All the methods are trained with $M = 4$ and $K = 7$, and tested on the Test 2 splits with $M = 2$ and $K = 11$.

| Dataset | Method | ARI-A | AMI-A | ARI-O | AMI-O | IoU | F1 | OCA | OOA |
|---------|--------|-------|-------|-------|-------|-----|----|-----|-----|
| CLEVR-M1 | Baseline | 0.342±1e-3 | 0.403±2e-3 | 0.529±5e-3 | 0.692±3e-3 | 0.201±3e-3 | 0.316±4e-3 | 0.060±6e-3 | 0.683±1e-2 |
|         | MulMON Prop. | 0.577±3e-2 | 0.579±1e-3 | 0.910±2e-3 | 0.907±1e-3 | N/A | N/A | 0.396±4e-2 | N/A |
| CLEVR-M2 | Baseline | 0.334±3e-3 | 0.400±1e-3 | 0.525±6e-3 | 0.668±3e-3 | 0.207±2e-3 | 0.323±2e-3 | 0.064±3e-2 | 0.673±3e-2 |
|         | MulMON Prop. | 0.566±1e-2 | 0.891±5e-3 | 0.896±1e-3 | N/A | N/A | 0.420±2e-2 | N/A |
| CLEVR-M3 | Baseline | 0.331±3e-3 | 0.394±2e-3 | 0.531±9e-3 | 0.691±4e-3 | 0.203±2e-3 | 0.318±3e-3 | 0.070±1e-2 | 0.680±3e-2 |
|         | MulMON Prop. | 0.566±1e-2 | 0.891±3e-3 | 0.890±1e-3 | N/A | N/A | 0.330±5e-2 | N/A |
| CLEVR-M4 | Baseline | 0.340±1e-3 | 0.409±2e-3 | 0.537±5e-3 | 0.700±4e-3 | 0.215±1e-3 | 0.334±2e-3 | 0.060±2e-2 | 0.720±1e-2 |
|         | MulMON Prop. | 0.551±2e-2 | 0.579±2e-3 | 0.901±4e-3 | 0.903±2e-3 | N/A | N/A | 0.426±5e-2 | N/A |

Table 9: Comparison of scene decomposition performance when learning from multiple viewpoints. All the methods are trained with $M = 4$ and $K = 7$, and tested on the Test 2 splits with $M = 4$ and $K = 11$.

| Dataset | Method | ARI-A | AMI-A | ARI-O | AMI-O | IoU | F1 | OCA | OOA |
|---------|--------|-------|-------|-------|-------|-----|----|-----|-----|
| CLEVR-M1 | Baseline | 0.333±2e-3 | 0.318±5e-3 | 0.289±4e-3 | 0.478±2e-3 | 0.136±9e-3 | 0.228±1e-3 | 0.002±4e-3 | 0.611±1e-2 |
|         | MulMON Prop. | 0.557±1e-3 | 0.579±1e-3 | 0.916±4e-3 | 0.910±2e-3 | N/A | N/A | 0.426±6e-2 | N/A |
| CLEVR-M2 | Baseline | 0.332±1e-3 | 0.324±1e-3 | 0.294±4e-3 | 0.482±2e-3 | 0.140±1e-3 | 0.235±2e-3 | 0.002±4e-3 | 0.602±2e-2 |
|         | MulMON Prop. | 0.570±8e-4 | 0.593±1e-3 | 0.903±1e-3 | N/A | N/A | 0.546±5e-2 | N/A |
| CLEVR-M3 | Baseline | 0.300±2e-3 | 0.307±2e-3 | 0.291±7e-3 | 0.478±4e-3 | 0.134±7e-4 | 0.225±9e-4 | 0.002±4e-3 | 0.625±2e-2 |
|         | MulMON Prop. | 0.566±1e-3 | 0.569±1e-3 | 0.899±5e-3 | 0.897±3e-3 | N/A | N/A | 0.426±6e-2 | N/A |
| CLEVR-M4 | Baseline | 0.325±1e-3 | 0.320±2e-3 | 0.294±5e-3 | 0.484±4e-3 | 0.143±8e-4 | 0.239±1e-3 | 0.000±0e-0 | 0.632±1e-2 |
|         | MulMON Prop. | 0.552±9e-4 | 0.581±6e-4 | 0.902±1e-3 | 0.902±5e-4 | N/A | N/A | 0.462±4e-2 | N/A |

Table 10: Comparison of scene decomposition performance when learning from multiple viewpoints. All the methods are trained with $M = 4$ and $K = 7$, and tested on the Test 1 splits with $M = 8$ and $K = 7$. 
| Dataset       | Method      | ARI-A | AMI-A | ARI-O | AMI-O | IoU | F1  | OCA  | OOA  |
|---------------|-------------|-------|-------|-------|-------|-----|-----|------|------|
| CLEVR-M1      | Baseline    | 0.329±1e-3 | 0.245±2e-3 | 0.151±1e-3 | 0.294±4e-3 | 0.103±2e-3 | 0.180±2e-3 | 0.000±0e-0 | 0.570±9e-3 |
|               | MulMON      | 0.552±8e-4 | 0.575±1e-3 | 0.913±1e-3 | 0.906±1e-3 | N/A | N/A | 0.442±4e-2 | N/A |
|               | Proposed    | 0.366±8e-4 | 0.463±2e-3 | 0.855±6e-3 | 0.859±3e-3 | 0.367±2e-3 | 0.513±3e-3 | 0.190±4e-2 | 0.923±4e-3 |
| CLEVR-M2      | Baseline    | 0.327±6e-4 | 0.248±1e-3 | 0.155±4e-3 | 0.297±3e-3 | 0.105±1e-3 | 0.184±2e-3 | 0.000±0e-0 | 0.559±1e-2 |
|               | MulMON      | 0.534±6e-4 | 0.570±8e-4 | 0.900±2e-3 | 0.895±1e-3 | N/A | N/A | 0.538±6e-2 | N/A |
|               | Proposed    | 0.353±6e-4 | 0.450±6e-4 | 0.810±3e-3 | 0.828±2e-3 | 0.354±2e-3 | 0.498±2e-3 | 0.170±2e-2 | 0.889±1e-2 |
| CLEVR-M3      | Baseline    | 0.301±7e-4 | 0.236±1e-3 | 0.156±4e-3 | 0.295±4e-3 | 0.101±7e-4 | 0.177±1e-3 | 0.000±0e-0 | 0.591±2e-2 |
|               | MulMON      | 0.531±2e-3 | 0.566±1e-4 | 0.906±2e-3 | 0.899±8e-4 | N/A | N/A | 0.394±7e-2 | N/A |
|               | Proposed    | 0.357±4e-3 | 0.452±6e-4 | 0.838±5e-3 | 0.847±3e-3 | 0.360±3e-3 | 0.504±4e-3 | 0.178±3e-2 | 0.912±1e-2 |
| CLEVR-M4      | Baseline    | 0.322±7e-4 | 0.245±2e-3 | 0.156±3e-3 | 0.299±5e-3 | 0.106±2e-3 | 0.186±2e-3 | 0.000±0e-0 | 0.592±2e-2 |
|               | MulMON      | 0.556±5e-4 | 0.579±5e-4 | 0.892±1e-3 | 0.893±8e-4 | N/A | N/A | 0.508±3e-2 | N/A |
|               | Proposed    | 0.278±1e-3 | 0.409±2e-3 | 0.783±6e-3 | 0.813±4e-3 | 0.310±3e-3 | 0.451±4e-3 | 0.134±3e-2 | 0.873±1e-2 |

Table 11: Comparison of scene decomposition performance when learning from multiple viewpoints. All the models are trained with \( M = 4 \) and \( K = 7 \), and tested on the Test 2 splits with \( M = 8 \) and \( K = 11 \).
| Dataset | Method       | ARI-A       | AMI-A       | ARI-O       | AMI-O       | IoU         | F1          | OCA         | OOA         |
|---------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|         |              |             |             |             |             |             |             |             |             |
| dSprites| Slot Attn    | 0.143±1e-3  | 0.348±8e-4  | 0.862±1e-3  | 0.869±7e-4  | N/A         | N/A         | 0.000±0e-0  | N/A         |
|         | GMIOO        | 0.958±4e-4  | 0.902±7e-4  | 0.920±1e-3  | 0.927±9e-4  | 0.780±2e-3  | 0.843±2e-3  | 0.558±8e-3  | 0.871±3e-3  |
|         | SPACE        | 0.914±3e-4  | 0.818±1e-4  | 0.779±8e-4  | 0.828±3e-4  | 0.562±6e-4  | 0.653±8e-4  | 0.204±8e-3  | 0.654±9e-3  |
|         | Proposed     | 0.779±5e-4  | 0.692±4e-4  | 0.797±2e-3  | 0.824±1e-3  | 0.615±1e-3  | 0.740±1e-3  | 0.330±1e-2  | 0.677±6e-3  |
| Abstract| Slot Attn    | 0.909±1e-3  | 0.843±1e-3  | 0.904±1e-3  | 0.881±8e-4  | N/A         | N/A         | 0.632±1e-2  | N/A         |
|         | GMIOO        | 0.916±3e-4  | 0.758±4e-4  | 0.905±1e-3  | 0.902±7e-4  | 0.736±1e-3  | 0.829±1e-3  | 0.796±4e-3  | 0.936±1e-3  |
|         | SPACE        | 0.849±2e-4  | 0.758±3e-4  | 0.761±8e-4  | 0.799±7e-4  | 0.614±7e-4  | 0.693±8e-4  | 0.267±2e-3  | 0.789±4e-3  |
|         | Proposed     | 0.844±3e-4  | 0.776±2e-4  | 0.901±1e-3  | 0.894±8e-4  | 0.770±9e-4  | 0.861±1e-3  | 0.664±1e-2  | 0.926±2e-3  |
| CLEVR   | Slot Attn    | 0.078±8e-5  | 0.378±2e-4  | 0.939±8e-4  | 0.945±5e-4  | N/A         | N/A         | 0.011±2e-3  | N/A         |
|         | GMIOO        | 0.666±3e-4  | 0.670±4e-4  | 0.884±1e-3  | 0.923±8e-4  | 0.515±1e-3  | 0.624±1e-3  | 0.160±5e-3  | 0.855±2e-3  |
|         | SPACE        | 0.823±3e-4  | 0.781±2e-4  | 0.932±5e-4  | 0.939±4e-4  | 0.684±4e-4  | 0.773±5e-4  | 0.417±3e-3  | 0.895±4e-3  |
|         | Proposed     | 0.478±8e-4  | 0.560±7e-4  | 0.916±1e-3  | 0.925±1e-3  | 0.473±2e-3  | 0.621±2e-3  | 0.309±1e-2  | 0.932±3e-3  |

Table 14: Comparison of scene decomposition performance when learning from a single viewpoint. All the methods are trained with \( K = 6 \), \( K = 5 \), and \( K = 7 \), and tested on the Test 2 splits with \( K = 9 \), \( K = 7 \), and \( K = 11 \) for the dSprites, Abstract, and CLEVR datasets, respectively.

Figure 4: Qualitative comparison on the CLEVR-M1 dataset.
Figure 5: Qualitative comparison on the CLEVR-M2 dataset.
Figure 6: Qualitative comparison on the CLEVR-M3 dataset.
Figure 7: Qualitative comparison on the CLEVR-M4 dataset.
Figure 8: Qualitative comparison on the dSprites dataset.
Figure 9: Qualitative comparison on the Abstract dataset.
Figure 10: Qualitative comparison on the CLEVR dataset.