Learning Latent Structural Relations with Message Passing Prior

Shaogang Ren, Hongliang Fei, Dingcheng Li, Ping Li
Cognitive Computing Lab
Baidu Research
10900 NE 8th St. Bellevue, WA 98004, USA
{renshaogang, feihongliang0, dingchengl, pingli98}@gmail.com

Abstract

Learning disentangled representations is an important topic in machine learning with a wide range of applications. Disentangled latent variables represent interpretable semantic information and reflect separate factors of variation in data. Although generative models can learn latent representations as well, most existing models ignore the structural information among latent variables. In this paper, we propose a novel approach to learn the disentangled latent structural representations from data using decomposable variational auto-encoders. We design a novel message passing prior for the latent representations to capture the interactions among different data components. Different from many previous methods that ignore data component or object interaction, our approach simultaneously learns component representation and encodes component relationships. We have applied our model to tasks of data segmentation and latent representation learning among different data components. Experiments on several benchmarks demonstrate the utility of the proposed method.

1. Introduction

Disentangled representation learning, which aims to learn factorized representations that disentangle the latent explanatory factors in data, is a fundamental but challenging problem in machine learning and artificial intelligence. Interpretable disentangled representations have demonstrated their power in unsupervised learning and semi-supervised learning [2, 12, 8, 3, 22].

Most existing methods for disentangled representation learning [18, 25, 4, 5, 41] are based on Variational Auto-Encoders (VAEs) [26] or Generative Adversarial Networks (GAN) [11, 31]. These works' commonality is that disentangled representations are extracted from a single entity or object in one data sample. However, in real-world scenarios, there are often multiple objects with complex interactions among them. Modelling object interactions has demonstrated its benefit in applications such as image segmentation [38] and video frame prediction [19]. In the literature of scene segmentation, there are a few attempts to leverage generative representation learning models at multiple objects level [13, 14, 36, 9]. Nevertheless, very few of them consider the structural interaction among multiple objects or sample portions.

The major challenge to learn representations from images with multiple objects lies in an unsupervised setting and complicated interaction patterns. Moreover, learning complicated object interactions in real-word requires a powerful and flexible prior for latent variables that can adaptively encode complicated structural relations. In this paper, we propose a bi-level variational auto-encoder based framework that can seamlessly integrate data segmentation, representation learning, and relation learning.

In our bi-level model, the latent representation vector for each object or component in a scene is divided into two sections, a local section and a global section. Firstly, the local section controls the individual properties that are independent of the other objects. The global section, shared by all the objects in a scene, encodes the object relationships as well as the global latent factors. The inference and interaction between different objects are handled with a flow-based model, in which a structural message passing prior of latent representation allows us to estimate correlation interaction between two components.

We have applied our models to different datasets and obtain significant improvement in scene segmentation, scene generation and object representation learning by modelling the interactions among different components. Compared to existing methods, our approach can capture more relations between objects. Furthermore, we provide the theoretical properties of our proposed bi-level VAE, such as relation identification and the Evidence Lower Bound (ELBO).

Overall, the contributions of our work are multi-folds: i) we develop a unified bi-level VAE framework with a latent structural message passing prior to seamlessly integrate data segmentation, representation learning, and relation learn-

ing, and ii) we provide a solid derivation of ELBO for the proposed bi-level VAE framework and comprehensive theoretical analysis for the latent structural prior as well as relation recovery and latent representation learning, and iii) We conduct extensive empirical evaluation of our approach on the tasks of latent representation learning and component segmentation. Experiments show that segmentation, generation, and disentangled representation of different components can be improved with the inference mechanism from our bi-level VAE with the novel prior.

2. Related Work

2.1. VAE Based Disentanglement

Variants of VAEs have achieved SOTA performance for unsupervised disentanglement learning. One can assume a specific prior $p(\mathbf{z})$ on the latent space and parameterize the conditional probability $p(\mathbf{x}|\mathbf{z})$ with a deep neural network. The distribution $p(\mathbf{z}|\mathbf{x})$ is approximated using a variational distribution $q(\mathbf{z}|\mathbf{x})$. The objective function for VAE is

$$\min_{\phi,\theta} \mathbb{E}_{p(\mathbf{x})}\big[-\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] + D_{\mathbf{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))\big]$$

which is also the negative Evidence Lower Bound (ELBO). It is also possible to introduce various properties of the final presentation by modifying the **KL** term. [18] proposed the β-VAE, introducing a hyper-parameter β for the **KL** regularizer of vanilla VAEs. When $\beta > 1$, β-VAE penalizes the mutual information between latent representation and data sample. There are several different approaches to learn disentangled data representation [25, 4]. Independent component analysis (ICA) has been extended to nonlinear cases to achieve disentanglement of variables [20, 21, 24].

Our work is different from vanilla VAE and its variants in that ours has a bi-level structure with a novel structural message passing prior to simultaneously realize data segmentation, representation learning, and relation learning.

2.2. Scene Segmentation

Recently, researchers integrated deep generative models with unsupervised scene segmentation methods [2, 12, 13, 8]. The most similar works to ours are [12], [2], and [8]. In [12], the authors proposed an approach to learn the representation of individual objects and scene segmentation simultaneously. By integrating iterative amortized inference [28] and VAE [26], the method is a fully unsupervised approach to learn visual concepts. They also showed how the complete system can be trained end-to-end by simply maximizing its Evidence Lower Bound (ELBO). MONet [2] employed a recurrent attention network to discriminate different objects instead of using complicated amortized inference. The scene is segmented by leveraging the weighted objective with attention masks. Besides the encoding of components, Genesis [8] improves performance by jointly learning the representations of both components and masks. The major difference between our work and [12, 2, 8] is that the interactions among objects in a scene are modeled with a latent message passing prior.

3. Latent Relational Learning with Message Passing Prior

We first introduce the proposed message passing prior, including forward message passing (encoding) and backward message passing (decoding) [29, 32, 33, 30]. Then we give details about the proposed bi-level VAE framework. Note that "components" represent objects in an image or different portions in a data sample and we will use the two terms interchangeably.

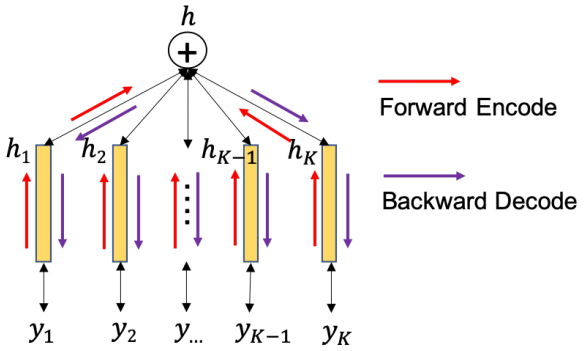

Figure 1. Diagram of message passing prior, including forward (encoding) and backward (decoding) with flow functions $f = \{f_1, f_2, ..., f_K\}$. Given observation $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K]$, we can infer the latent variable $\mathbf{h}$ with forward message passing (encoding), and obtain the reconstruction of $\mathbf{y}$, $\widehat{\mathbf{y}}$, with backward message passing (decoding). Here $\mathbf{h}_k = f_k(\mathbf{y}_k)$, $\mathbf{h} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{h}_k$, and reconstruction $\widehat{\mathbf{h}}_k = \mathbf{h}$, $\widehat{\mathbf{y}}_k = f_k^{-1}(\mathbf{h})$.

We introduce the proposed aggregation prior model in Figure 1. Let $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K]$ be the observed data, and $\mathbf{y}_k$ is from data component k, and $\mathbf{h}$ is the latent variable. We use $\mathcal{Y}$ to represent the distribution of $\mathbf{y}$. Relationship between $\mathbf{y}_k, k = 1, ..., K$ and $\mathbf{h}$ is modeled with invertible flow-based networks [6, 32, 29]. Flow function f_k specifies a parametric invertible transformation from the distribution of $\mathbf{y}_k$ to the latent variable $\mathbf{h}_k$, i.e., $f_k : \mathcal{R}^l \to \mathcal{R}^l$ is invertible. Here l is the dimension of $\mathbf{h}_k$ and $\mathbf{y}_k$. With $\mathbf{h}_k = f_k(\mathbf{y}_k)$, by change-of-variables we obtain

$$\log p(\mathbf{y}_k) = \log p(\mathbf{h}_k) + \log\left(\left|\det\left(\frac{\partial f_k(\mathbf{y}_k)}{\partial \mathbf{y}_k}\right)\right|\right).$$

As shown in Figure 1, the relation between $\mathbf{h}$ and $\mathbf{y}_k, k = 1, ..., K$ is given as encoding (with $f = [f_1, f_2, ..., f_K]$) and decoding (with $f = [f_1^{-1}, f_2^{-1}, ..., f_K^{-1}]$) procedures. $\mathbf{h}$ encodes $\mathbf{y}$ by aggregating outputs of all f_ks, i.e., $\mathbf{h} = f(\mathbf{y}) = \frac{1}{K}\sum_{k=1}^{K} f_k(\mathbf{y}_k)$. We hope that the aggregated latent variable $\mathbf{h}$ is a concise representation so that the model

can fully reconstruct all components of the data, i.e., ensures that $\widehat{\mathbf{h}}_k = \mathbf{h}_k = \mathbf{h}$, and $\widehat{\mathbf{y}}_k = \mathbf{y}_k = f_k^{-1}(\mathbf{h})$. Here $\widehat{\mathbf{h}}_k$ and $\widehat{\mathbf{y}}_k$ are reconstructions of $\mathbf{h}_k$ and $\mathbf{y}_k$, respectively.

3.1. Latent Variable Aggregation

We assume each entry of $\mathbf{h}_k, k = 1, ..., K$ follows Normal distribution, i.e., $\mathbf{h}_k \sim \mathrm{N}(\mu_k, \sigma^2)$. We set variance σ^2 as a fixed value across all ks. With $\mathbf{h} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{h}_k$, the prior distribution for each entry of $\mathbf{h}$ is a Normal distribution, N(μ, σ^2). Both $\mathbf{h}$ and $\mathbf{h}_k$ can be other distributions, e.g., Laplace distribution. Based on the encoder and decoder VAE scheme discussed previously, model parameters of the aggregation model can be learned by maximizing the evidence lower bound (ELBO),

$$\begin{aligned} &\log p_{f^{-1}}(\mathbf{y}) \geq \mathcal{L}(\mathbf{y}; f) \qquad (1)\\ &= \mathbb{E}_{q_f(\mathbf{h}|\mathbf{y})}\big[\log p_{f^{-1}}(\mathbf{y}|\mathbf{h})\big] - \mathbf{KL}\big(q_f(\mathbf{h}|\mathbf{y})||p(\mathbf{h})\big). \end{aligned}$$

Given a batch of training samples, the ELBO value is computed with the message passing procedures. We use $\mathbf{h} = f(\mathbf{y}) = \frac{1}{K}\sum_{k=1}^{K} f_k(\mathbf{y}_k)$ as the sample generated from $q_f(\mathbf{h}|\mathbf{y})$. Given a $\mathbf{h}$, we hope it can fully reconstruct the input data. Thus the reconstruction term $\log p_{f^{-1}}(\mathbf{y}|\mathbf{h})$ in the ELBO (1) is computed with

$$\begin{aligned} &\log p_{f^{-1}}(\mathbf{y}|\mathbf{h}) = \log p_{f^{-1}}(\mathbf{y}|\widehat{\mathbf{h}}_1 ... \widehat{\mathbf{h}}_K) + \log p(\widehat{\mathbf{h}}_1 ... \widehat{\mathbf{h}}_K|\mathbf{h})\\ &= -\sum_{k=1}^{K} \Big\{ \underbrace{\frac{1}{2\sigma_{\mathbf{y}}^2}||\mathbf{y}_k - f_k^{-1}(\mathbf{h})||^2 + \frac{1}{2\sigma^2}||\mathbf{h} - f_k(\mathbf{y}_k)||^2}_{\text{By } \widehat{\mathbf{h}}_k = \mathbf{h}} \Big\} + C \end{aligned} \qquad (2)$$

Here $C = -lK\ln(2\pi) - lK\ln(\sigma_{\mathbf{y}}^2)$. We use constant values for both $\sigma_{\mathbf{y}}^2$ and σ^2, hence the value of C. We use $\mathbf{h}$s from a batch of training samples to approximate the $\mathbf{KL}$ term in (1). It is easy to compute each flow function f_k's Jacobian matrix, and thus the log-density values. We use the proposed structure to estimate the relations among the components.

3.2. Graphical Interaction

Let $y_{k,i}$ be the ith entry of $\mathbf{y}_k$. We define a relation $e_{v,j}^{u,i}$ between $y_{u,i}$ and $y_{v,j}$ if there is a mapping or a function links them. A relation set is a connected graph $\mathbf{r} = \{\mathbf{e}, \mathbf{v}\}$ that consists of multiple relations, and here $\mathbf{v}$ represents the set of variables involved in $\mathbf{r}$, and $\mathbf{e}$ is the set of link functions between variables in $\mathbf{v}$. Let $\mathcal{R}$ be the set of all relation sets regarding a data set $\mathcal{Y}$. We have the following assumptions about $\mathcal{Y}$ and $\mathcal{R}$.

Assumption 1: $\mathcal{Y}$ is continuously distributed. Data value of $\mathcal{Y}$ is bounded, i.e., $y_{u,i} \in [-M, M], \forall 1 \leq u \leq K$, $1 \leq i \leq l$, and M is a constant value.

Assumption 2: Relation functions are continuous, monotone, and invertible. Their inverse functions are also continuous[1].

[1]The relation e and its inverse lie in a Hölder ball $\mathcal{W}^{\beta,\infty}([-1,1]^d)$ with smoothness $\beta \in \mathbb{N}_+$, i.e. $e, e^{-1} \in \mathcal{W}^{\beta,\infty}([-1,1]^d)$.

As relation set r is a connected graph, there is always a path connecting any two variables. Let's use $g_{u,i}^*$ to represent the prediction function from other variables to $y_{u,i}$ in a relation set $\mathbf{r}$, and $\widehat{g}_{u,i}$ is the estimation with the proposed message passing model. With n as the number of training samples from $\mathcal{Y}$, we have the following theorem regarding the estimation.

Theorem 1. *Let the assumptions 1-2 hold, and $|\mathcal{R}| \leq dim(\mathbf{h})$. Let $\widehat{g}_{u,i}$ be the estimator that consists of deep coupling layers with width $W \asymp n^{\frac{d}{2(\beta+d)}}\log^2 n$, and depth $D \asymp \log n$. For large enough n, with probability at least $1 - \exp(-n^{\frac{d}{\beta+d}}\log^8 n)$,*

a) $\|\widehat{g}_{u,i} - g_{u,i}^*\|_{L_2(Y)}^2 \leq C \cdot \big\{ -n^{\frac{\beta}{\beta+d}}\log^8 n + \frac{\log\log n}{n} \big\}$ *and*

b) $\mathbb{E}_n\big[(\widehat{g}_{u,i} - g_{u,i}^*)^2\big] \leq C \cdot \big\{ -n^{\frac{\beta}{\beta+d}}\log^8 n + \frac{\log\log n}{n} \big\}$.

Here $C > 0$ is a constant independent of n.

Theorem 1 says that the interactions among different components can be approximately recovered under conditions. Under the assumption that variables from different relation sets are independent with other, the regularization of the $\mathbf{KL}$ term in (1) will guide the model to learn latent variables that control different relation sets. Minimizing the $\mathbf{KL}$ term is to force each entry of the root latent variable $\mathbf{h}$ in (1) to become more independent with each other, and it is because that different entries of the prior distribution $p(\mathbf{h})$ we employed are independent with each other.

3.3. Identifiability of Latent Representation with Unsupervised Component Segmentation

With the invertible flow-based model, we can fit the proposed model to the nonlinear ICA framework [24, 15, 21]. For component k, suppose the distribution regarding $\mathbf{h}_k$ is a factorial member of the exponential family with m sufficient statistics, conditioned on $\mathbf{u}_k$. Here $\mathbf{u}_k$ is additional observed variable. The general form of the distribution can be written as

$$p_{\mathbf{h}_k}(\mathbf{h}_k|\mathbf{u}_k) = \mathbf{\Pi}_{i=1}^{l} \frac{Q_i(h_{k,i})}{Z_i(\mathbf{u}_k)} \exp\Big[\sum_{j=1}^{m} T_{i,j}(h_{k,i})\lambda_{i,j}(\mathbf{u}_k)\Big]. \qquad (3)$$

Here Q_i is the base measure, Z_i is the normalizing constant, $T_{i,j}$ are the component of the sufficient statistic and $\lambda_{i,j}$ the corresponding parameters, depending on $\mathbf{u}_k$. The variable $\mathbf{y}_k$ is the output of an arbitrarily complex, inevitable, and deterministic transformation from the latent space to the data space, i.e., $\mathbf{y}_k = f_k^{-1}(\mathbf{h}_k)$. Let $\mathbf{T} = [\mathbf{T}_1, ..., \mathbf{T}_l]$, $\lambda = [\lambda_1, ..., \lambda_l]$, and $\Theta = \{\theta := (\mathbf{T}, \lambda, f_k^{-1})\}$. With parameter $\theta = (\mathbf{T}, \lambda, f_k^{-1})$,

$$p_\theta(\mathbf{y}_k, \mathbf{h}_k|\mathbf{u}_k) = p_{f_k^{-1}}(\mathbf{y}_k|\mathbf{h}_k) p_{\mathbf{T},\lambda}(\mathbf{h}_k|\mathbf{u}_k). \qquad (4)$$

Let $\widehat{\Theta}$ be the set of parameters obtained with some learning algorithm, i.e., $\widehat{\Theta} = \{\widehat{\theta} := (\widehat{\mathbf{T}}, \widehat{\lambda}, g_k)\}$. We use g_k to represent the learned approximation of f_k^{-1}, and $\mathbf{y}_k = g_k(\mathbf{h}_k)$. Following [24, 15], we define identifiable equivalence relations on Θ. We do not have explicit additional observable variable $\mathbf{u}_k$ for component k. But we have $K-1$ signals from other components relate to it. With the statements in Theorem 1, suppose we can fully recover the relations involving component k, and can obtain sufficient label support from other components, then the model is identifiable.

We use $\mathbf{y}_{-k}$ to represent components other than component k, and $\mathbf{u}_k(\mathbf{y}_{-k})$ is the additional variable recovered from the relations with other components. In the limit of infinite data and good convergence, the estimating model will give the same conditional likelihood to all data points as the true generating model: $p_{\mathbf{T},\lambda,f_k^{-1}}\big(\mathbf{y}_k|\mathbf{u}_k(\mathbf{y}_{-k})\big) = p_{\widehat{\mathbf{T}},\widehat{\lambda},g_k}\big(\mathbf{y}_k|\mathbf{u}_k(\mathbf{y}_{-k})\big)$. We define the domain of f_k^{-1} as $\mathcal{H} = \mathcal{H}_1 \times ... \times \mathcal{H}_l$. We have the follow theorem regarding the identifiability of the model.

Theorem 2. *Assume we observe data distributed according to the generative model given by* (3) *and* (4)*, we further have the following assumptions,*

(a) The sufficient statistics $T_{ij}(h)$ are differentiable almost everywhere and their derivatives $\frac{dT_{i,j}}{dh}$ are nonzero almost surely for all $h \in \mathcal{H}_i$ and all $1 \leq i \leq l$ and $1 \leq j \leq m$.

(b) The relations involving component k can be approximately fully recovered and can be represented with $\mathbf{u}_k(\mathbf{y}_{-k})$.

(c) There exist $lm+1$ distinct conditions $\mathbf{u}_k^{(0)}$, ..., $\mathbf{u}_k^{(lm)}$ from $\mathbf{y}_{-k}$ such that the matrix

$$\mathbf{L} = [\lambda(\mathbf{u}_k^{(1)}) - \lambda(\mathbf{u}_k^{(0)}), ..., \lambda(\mathbf{u}_k^{(lm)}) - \lambda(\mathbf{u}_k^{(0)})]$$

of size $lm \times lm$ is invertible. Then the model parameters $(\mathbf{T}, \lambda, f_k^{-1})$ are $\sim_{\mathbf{A}}$ identifiable.

The proof of Theorem 2 and analysis can be found in the supplemental file. Real-world datasets are usually more complicated with non-stationary component locations. We try to develop a bi-level latent model that is more flexible by integrating the proposed aggregation prior model, attention mechanism, and component segmentation as discussed in the following sections.

4. Bi-level Latent Structure for Component Segmentation

We aim to develop a generative model that can identify the hierarchical representation and relations of components in datasets. In this section, we first introduce a decomposed latent representation scheme, and then show that the proposed message passing aggregation prior can be seamlessly integrated with some existing models.

4.1. Global Latent Variable for Component Interaction

Generative models learn a generator that maps the latent space $\mathbb{Z}$ to a manifold $\mathbb{X}$ embedded in the sample input space. Assume there are K conditional independent components for the samples of a dataset. Let $\mathbf{x} = [\mathbf{x}_1, ..., \mathbf{x}_K]$ be the output variable of the generator, and $\mathbf{z} = [\mathbf{z}_1, ..., \mathbf{z}_K]$ is the latent variable of the generator. $\mathbf{x}_k$ is the kth component, and $\mathbf{z}_k$ is the corresponding latent variable that contains all the latent information of component k. Each $\mathbf{z}_k$ has two sections, $\mathbf{z}_k^c$ and $\mathbf{z}_k^g$, i.e., $\mathbf{z}_k = [\mathbf{z}_k^c, \mathbf{z}_k^g]$. $\mathbf{z}_k^c$ controls the properties of component k that are independent with other components, and $\mathbf{z}_k^g$ controls the properties relating to other components. We use $\mathbf{z}^0$ to denote the latent vector encodes the global properties information across all components regarding each data sample $\mathbf{x}$. We first assume the components are conditional independent with each other given the latent variable, i.e., $\mathbf{x}_i \perp \mathbf{x}_k|\mathbf{z}$, if $i \neq k$.

We also have the following independent assumption about the components and latent variables, $\mathbf{x}_i \perp \mathbf{z}_k|\mathbf{z}^0$, if $i \neq k$, and $\mathbf{z}_i \perp \mathbf{z}_k|\mathbf{z}^0$, if $i \neq k$. It is easy to show that the distribution of the generated samples are following

$$p(\mathbf{x}_1...\mathbf{x}_K|\mathbf{z}) = p(\mathbf{x}_1...\mathbf{x}_K|\mathbf{z}^0\mathbf{z}) = p(\mathbf{x}_1...\mathbf{x}_K|\mathbf{z}^0\mathbf{z}_1...\mathbf{z}_K)$$
$$= \prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{z}^0\mathbf{z}_1...\mathbf{z}_K) = \prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{z}^0\mathbf{z}_k) = \prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{z}^0\mathbf{z}_k^c).$$

In the last step, $\mathbf{z}_k^g$s are deterministic given $\mathbf{z}^0$, thus they can be omitted.

We employ a hierarchy structure for the latent variables. As shown in Figure 2, $\mathbf{z}_1$, ..., $\mathbf{z}_K$ are the first layer latent representation, and $\mathbf{z}^0$ is the second layer. As mentioned previously, $\mathbf{z}^0$ encodes the global properties of the generated samples, and the correlations or interactions between different components. $\mathbf{z}_k^g$ is the global information decoded from $\mathbf{z}^0$ regarding component k. We can use the human face as an illustration example. Here different components

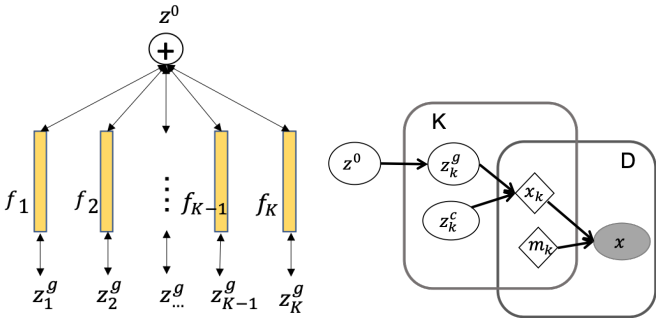

Figure 2. Hierarchy structure for latent variables. Left: $\mathbf{z}_k^g$s link to the global latent variable $\mathbf{z}^0$ with the message passing prior. Right: Global latent variable $\mathbf{z}^0$ is shared by K components. $\mathbf{m}_k$ is the mask of component $\mathbf{x}_k$. D is the dimension number of the input data samples.

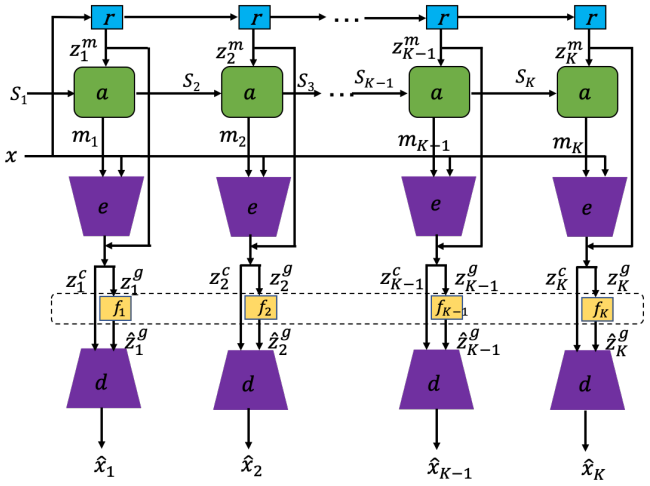

Figure 3. Network structure for MPP^G. a is the attention network, e is the encoder, d is the decoder, and f_k is the flow inference network for component k, and r is the recurrent network for latent variable $\mathbf{z}^m$. $\widehat{\mathbf{z}}_k^g$ and $\widehat{\mathbf{x}}_k$ are the reconstructions of $\mathbf{z}_k^g$ and $\mathbf{x}_k$, respectively. $\mathbf{z}_k^m$ is the latent variable of mask k. The input scope for kth component is defined by $\mathbf{s}_k = \mathbf{s}_{k-1} \circ (1 - \mathbf{m}_{k-1})$.

represent different parts of the face, such as eyes, hair, facial skin, mouth, etc. The common latent factor $\mathbf{z}^0$ includes factors such as age or emotion that controls the overall appearance of the face. We aim to develop a framework that can encode each component's individual features as well as the global latent factors for the whole scene. The structure of the proposed prior provides sufficient capacity to capture the relationships among different components. It can capture the structural configurations even in scenarios that some component or objects are absent. The detailed structure relationships among components are represented with the correlations of the input entries of different flow branches.

4.2. Bi-level Latent Model Structure

The proposed message passing prior is to encode and decode each component and capture the global latent factor as well. To derive a simple model, we use one single VAE framework for encoding and decoding of all components. The sequence of masks for each component can be generated with the approaches in MONet [2] or Genesis [8]. The method in Genesis leverages the latent represent of masks to improve performance.

We use MPP^M to represent the model that follows the MONet attention structure but enhanced with the proposed message passing prior. Similarly, MPP^G is the model that employs latent representation for masks (Genesis) and also uses message passing prior to integrate components. Figure 3 presents the integrated model structure of Genesis with the proposed message passing prior. In component k, with image $\mathbf{x}$ and scope $\mathbf{s}_k$ as the input, the attention network a yields the mask $\mathbf{m}_k$ to indicate whether each pixel of $\mathbf{x}$ belonging to component k or not. Here $\mathbf{s}_k$ is the attention leftover from components 1 to $k-1$, i.e., $\mathbf{s}_k = \overline{\cup_{i=1}^{k-1} \mathbf{m}_i}$, and $\mathbf{s}_1 = \mathbf{1}$. Figure 3 shows that the scope for component k is calculated by $\mathbf{s}_k = \mathbf{s}_{k-1} \circ (\mathbf{1} - \mathbf{m}_{k-1})$, and we have $\sum_{k=1}^K \mathbf{m}_k = \mathbf{1}$. $\circ$ denotes element-wise multiplication.

The encoder e encodes the image and the mask $(\mathbf{x}, \mathbf{m}_k)$ into the latent variables $\mathbf{z}_k = \mathbf{z}_k^c \mathbf{z}_k^g$. We use the message passing prior proposed in the previous section as the second layer auto-encoder to encode all $\mathbf{z}_k^g$s into $\mathbf{z}^0$ and then decode back as $\widehat{\mathbf{z}}_k^g$s. Then we feed each $(\mathbf{z}_k^c, \widehat{\mathbf{z}}_k^g)$ to the decoder d to generate the image reconstruction $\widehat{\mathbf{x}}_k$. The model performs image segmentation by leveraging a mixture model that takes masks as the distribution weights of different components.The message passing prior can curb the model's degree of freedom and can capture the interaction between different segments or components as well. Notations for the bi-level model are given by a table in the supplement.

4.3. ELBO of Bi-level Latent Model

The proposed prior and the latent decomposition scheme can be applied to many generative models for segmentation [12, 2, 8]. Let $\mathbf{z}_k^m$ be the latent representation of mask k. Genesis [8] has the following assumption about latent variables: $p(\mathbf{z}_{1:K}^m) = p(\mathbf{z}_1^m) \prod_{k=2}^K p(\mathbf{z}_k^m | \mathbf{z}_{1:k-1}^m)$ and $p(\mathbf{z}_{1:K}^c | \mathbf{z}_{1:K}^m) = \prod_{k=1}^K p(\mathbf{z}_k^c | \mathbf{z}_k^m)$.

They have the sequential dependent assumption about latent represents of masks, and the components' latent representation also relates to masks'. Message passing prior can incorporate different assumptions on the latent representation. We provide a general ELBO for bi-level latent model:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z}^c, \mathbf{z}^g, \mathbf{z}^m | \mathbf{x})} \big[\log p_\theta(\mathbf{x}) | \mathbf{z}^c, \mathbf{z}^g, \mathbf{z}^m \big] \\ - \mathbf{KL}(q(\mathbf{z}^c, \mathbf{z}^m | \mathbf{x}) \| p(\mathbf{z}^c, \mathbf{z}^m)) + \mathbf{H}(\mathbf{z}^g | \mathbf{x}) \\ + \mathbb{E}_{q(\mathbf{z}^g | \mathbf{x})} \big[\log p(\mathbf{z}^g | \mathbf{z}^0) \big] - \mathbf{KL}(q(\mathbf{z}^0 | \mathbf{z}^g) \| p(\mathbf{z}^0)).\end{aligned} \tag{5}$$

We use MPP^G as an example to show how to generalize (5) to a specific latent model. MPP^G has the latent variable $\mathbf{z}_k^m$ for mask k, and $\mathbf{z}_k^m$ dependent on $\mathbf{z}_{k-1}^m$, $1 < k \leq K$. Meanwhile, component k's latent variables $\mathbf{z}_k^c$ and $\mathbf{z}_k^g$ dependent on the mask's latent variable,

$$p_r(\mathbf{z}^m) = p_r(\mathbf{z}_1^m) \prod_{k=2}^K p_r(\mathbf{z}_k^m | \mathbf{z}_{1:k-1}^m), \tag{6}$$

$$p(\mathbf{z}^c | \mathbf{z}^m) = \prod_{k=1}^K p(\mathbf{z}_k^c | \mathbf{z}_k^m), \; p(\mathbf{z}^g | \mathbf{z}^m, \mathbf{z}^0) = \prod_{k=1}^K p(\mathbf{z}_k^g | \mathbf{z}_k^m, \mathbf{z}^0),$$

$$p(\mathbf{z}^c, \mathbf{z}^m) = p_r(\mathbf{z}^m) p(\mathbf{z}^c | \mathbf{z}^m).$$

Here $p_r(\cdot)$ means a distribution is parameterized by a neural network or a function r. The global latent variable $\mathbf{z}^g$ not only depends on $\mathbf{z}^m$ but also $\mathbf{z}^0$. The data distribution is a mixture model of K different components, $p(\mathbf{x}|\mathbf{z}^c, \mathbf{z}^g, \mathbf{z}^m) = \sum_{k=1}^K \mathbf{m}_k(\mathbf{z}_k^m) p_d(\mathbf{x}_k|\mathbf{z}_k^c, \mathbf{z}_k^g)$. Here $\mathbf{m}_k(\mathbf{z}_k^m)$ is the attention network a in Figure 3, and $p_d(\mathbf{x}_k|\mathbf{z}_k^c, \mathbf{z}_k^g)$ is parameterized with the decoder d in Figure 3. The approximate posteriors reads

$$q(\mathbf{z}^c, \mathbf{z}^g, \mathbf{z}^m|\mathbf{x}) = q_r(\mathbf{z}^m|\mathbf{x}) q_e(\mathbf{z}^c|\mathbf{z}^m, \mathbf{x}) q_e(\mathbf{z}^g|\mathbf{z}^m, \mathbf{x}) \tag{7}$$

$$\begin{aligned}
q_r(\mathbf{z}^m|\mathbf{x}) &= \mathbf{\Pi}_{k=1}^K q_r(\mathbf{z}_k^m|\mathbf{z}_{1:k-1}^m, \mathbf{x}), \\
q_e(\mathbf{z}^c|\mathbf{z}^m, \mathbf{x}) &= \mathbf{\Pi}_{k=1}^K q_e(\mathbf{z}_k^c|\mathbf{z}_k^m, \mathbf{x}), \\
q_e(\mathbf{z}^g|\mathbf{z}^m, \mathbf{x}) &= \mathbf{\Pi}_{k=1}^K q_e(\mathbf{z}_k^g|\mathbf{z}_k^m, \mathbf{x}), \\
q(\mathbf{z}^c, \mathbf{z}^m|\mathbf{x}) &= q_r(\mathbf{z}^m|\mathbf{x}) q_e(\mathbf{z}^c|\mathbf{z}^m, \mathbf{x}).
\end{aligned}$$

The posterior $q_r(\mathbf{z}^m|\mathbf{x})$ is modeled with a RNN (blue blocks in Figure 3 with label r). The posteriors of $\mathbf{z}^c$ and $\mathbf{z}^g$, $q_e(\mathbf{z}^c|\mathbf{z}^m, \mathbf{x})$ and $q_e(\mathbf{z}^g|\mathbf{z}^m, \mathbf{x})$, are parameterized with the encoder e network. As shown in the figure, both of them also dependents on $\mathbf{z}^m$. For the bi-level auto-encoder, $(\mathbf{x}, \mathbf{m}_k)$ is the first layer's input, and $(\mathbf{z}_k^c, \mathbf{z}_k^g)$ is the first layer's latent variable. Meanwhile, $\mathbf{z}_k^g$ is also the second layer's input, and $\mathbf{z}^0$ is the second layer's latent variable. $\widehat{\mathbf{x}}_k$ and $\widehat{\mathbf{z}}_k^g$ are the reconstructions regarding the first level and second level inputs, respectively.

As shown in the graphical representation of MPP^G (Figure 3), with $\mathbf{z}^0$ MPP^G can aggregate the information from all components simultaneously. The second level auto-encoder is parameterized with the proposed message passing prior model $f = \{f_1, f_2, ..., f_K\}$, i.e., $p_{f^{-1}}(\mathbf{z}^g|\mathbf{z}^0) = \mathbf{\Pi}_{k=1}^K p_{f_k^{-1}}(\mathbf{z}_k^g|\mathbf{z}^0)$, and the posterior of $\mathbf{z}^0$, $q_f(\mathbf{z}^0|\mathbf{z}^g)$, is the encoding process of the model f. The ELBO of MPP^G is

$$\begin{aligned}
&\log p(\mathbf{x}) \geq \mathcal{L}_{\text{ELBO}}(\mathbf{x}; a, d, e, f, r) \qquad (8) \\
=&\mathbb{E}_{q_{e,r}(\mathbf{z}^c, \mathbf{z}^g, \mathbf{z}^m|\mathbf{x})} \big[\log p_d(\mathbf{x}|\mathbf{z}^c, \mathbf{z}^g, \mathbf{z}^m) \big] \\
&- \mathbf{KL}(q_{e,r}(\mathbf{z}^c, \mathbf{z}^m|\mathbf{x}) \| p(\mathbf{z}^c, \mathbf{z}^m)) + \mathbf{H}(\mathbf{z}^g|\mathbf{z}^m, \mathbf{x}) \\
&+ \mathbb{E}_{q_f(\mathbf{z}^0|\mathbf{z}^g)}[\log p_{f^{-1}}(\mathbf{z}^g|\mathbf{z}^0)] - \mathbf{KL}(q_f(\mathbf{z}^0|\mathbf{z}^g) \| p(\mathbf{z}^0)).
\end{aligned}$$

The difference between (8) and the generalized ELBO (5) is the entropy term, $\mathbf{H}(\mathbf{z}^g|\mathbf{z}^m, \mathbf{x})$. It is due to the assumptions of Genesis. The terms in the ELBO (8) can be computed based on the discussion of Equations (6-7). The last two terms in (8) correspond to the ELBO defined in (1) of the message passing prior.

5. Experiments

We compare the proposed models (MPP^M and MPP^G) with baselines, MONet [2] and Genesis [8], using both synthetic and real-world datasets. The synthetic data is simulated under a multi-object setting, from which we demonstrate that the proposed prior can help learning the correlations between objects. We further validate our model on several real-world benchmarks.

5.1. Performance Metric

We primarily focus on the study of disentanglement and segmentation and compare our model to existing methods.

Disentanglement. Disentanglement evaluation metrics have been proposed by [18, 25, 7, 4]. For the experiments in this manuscript, we utilize the protocol proposed in [7], which is a regression-based approach to divide the latent space data into training, evaluation, and testing. The disentanglement score is obtained based on the performance of the learned regression model. The metric [7] is one of the common methods to measure disentanglement learning, and they are computed based on available ground-truth latent structure to evaluate representation according to disentanglement, completeness, and informativeness.

Segmentation. Following [12], we employ the adjusted rand index (ARI) to evaluate the segmentation. The ground truth mask and predicted mask are converted to binary values, and the similarity of a pair of masks is based on the number of the same entry values. An ARI score can be computed with the pair-wise similarity matrix.

Image Generation. FID [17] score is widely used to evaluate generative models, e.g., VAEs. In this paper, we utilize it to measure the quality of image synthesis.

In the result Tables, $\uparrow$ means a larger value gives a better result, and $\downarrow$ indicates a smaller value is with a better result.

5.2. Simulated Multi-Object Dataset

Now we investigate the proposed model with multiple-object images. The images are generated with three types of objects, green squares, red circles, and blue diamonds. The dataset has 50,000 samples for training and 2,000 samples for testing. The sample images are shown in the first row of both Figure 4-a) and Figure 4-b). We use the LASSO regressor with $\alpha = 0.2$ for the disentanglement score for this experiment. We try to incorporate object relations into the dataset to evaluate the performance of different models.

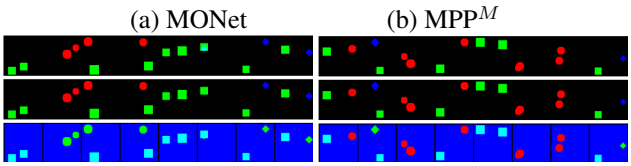

Figure 4. The original images, the reconstructed images, and mask images from MONet (left) and our method MPP^M (right) on the simulated 2-object dataset. There are 8 sample images for each method. MONet cannot distinguish between diamonds and circles. It is clear that the proposed method MPP^M can robustly distinguish and segment different types of object.

In the first set of experiments, we generate images that contain two objects. The predefined *generating logic* is: only object pairs {circle, circle}, {circle, square}, {square, square}, and {square, diamond} appear in the same image, and circles and diamonds can not appear in the same image. The bottom row of Figure 4 compares the segmentation from MONet and the proposed method. Different components learned by the algorithms are labeled with different colors. We can see that our method clearly categorizes three types of shapes into three different colored components as indicated in the bottom-right plots of Figure 4. Whereas, MONet puts circles and diamonds in the same component (Figure 4 bottom-left). The result indicates that our model can distinguish circles and diamonds as well as the designed logic relationship, but MONet cannot do it.

We further investigate the model with more complicated object relations that involve 3-object in an image. In this set of generated images, each image has two or three objects. Similarly, the designed *generating logic* is: circles and squares, squares, and diamonds can appear in one image, and circles and diamonds are not allowed to appear in the same image. We also notice that structured latent space with total correlation (TC) [39] penalization can also improve the disentanglement score of MONet.

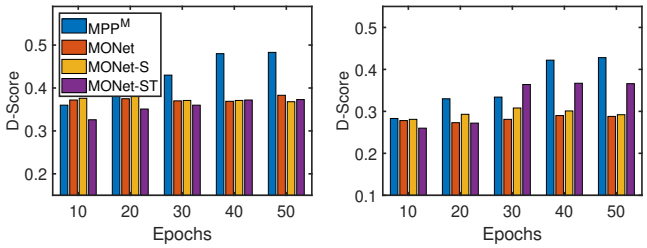

Figure 5. Disentanglement scores (D-Score ↑, $\alpha = 0.2$) over epochs for different methods on 2-object (left) and 3-object (right) datasets.

Figure 5 gives the disentanglement scores for different models along with different epoch numbers. In Figure 5, "MONet-S" means MONet with the structured latent variables introduced in the bi-level latent structure section 4.1. "MONet-ST" means MONet with structured latent variables in addition to the total correlation (TC). The left plot in Figure 5 show the disentanglement scores (D-Scores) from different methods on the simulated 2-object dataset, and the right plot gives the D-Socres on the 3-object dataset. In the 3-object dataset, each image contains two or three objects. Similar to the 2-object dataset, circles and squares, squares, and diamonds can appear in one image. Circles and diamonds are not allowed to appear in the same image. These rules are the latent component relations of the datasets.

As we can see from the left plot of Figure 5, the proposed aggregation prior with message passing can effectively capture the latent factor structures and improve the disentanglement score on the 2-object dataset. Figure 5 (right) shows the disentanglement scores for different models on the simulated 3-object dataset. We see that with the help of the proposed prior, the proposed method can effectively disentangle structured latent factors on a more complicated dataset.

5.3. Multi-dSprites Dataset

We further evaluate the proposed prior using Multi-dSprites dataset [23]. Each image consists of multiple oval, heart, or square-shaped sprites (with some occlusions) set against a uniformly colored background. Each scene image has 1 to 4 sprites. We use all the available features for disentanglement testing, which include positions (x and y), shape, color (RGB values), orientation, and scale, visibility (a binary feature indicating which objects are not null).

Methods	Disentangle ↑
MONet	0.623
MPP^{M} (Ours)	**0.636**

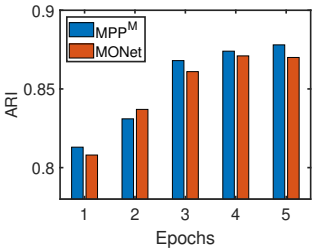

Figure 6. Disentanglement and segmentation scores of both methods on Multi-dSprites dataset. Left: disentanglement scores (↑, $\alpha = 0.2$). Right: segmentation scores (ARI ↑) at different epochs.

The disentanglement score is computed with LASSO as the regressor and $\alpha = 0.2$. The disentanglement performance is given in the left of Figure 6 after 20 epochs with a learning rate 10^{-4}. We observe that the proposed method can achieve superior disentanglement scores. The right of Figure 6 gives the values of segmentation score (ARI) over epochs. With adjustment information between components, because of the message passing scheme, the proposed model can consistently improve segmentation along with more epochs. Our method improves the segmentation along with the updating steps, and it produces more reasonable object segmentation.

5.4. Tetrominoes Dataset

Each image in the Tetrominoes dataset [23] contains three tetrominoes, sampled from 17 unique shapes or orientations. We use four components for all the models, i.e., MONet, Genesis, MPP^{M}, and MPP^{G}. We randomly select 1,000 images to evaluate disengagement and use the rest 999,000 images to train the models. Firstly, the three rows of images in Figure 7 are the original images, reconstructed images, and masks generated from the attention network of MPP^{M}, respectively. Clearly, the proposed method can well segment the objects. Secondly, Table 1 gives the disentanglement and FID scores for the four methods ($\alpha = 0.001$ for disentanglement score). We can that with the help of

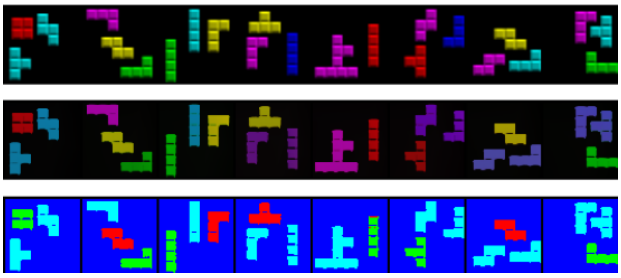

Figure 7. The original images, reconstructed images, and the masks generated from the proposed method for Tetrominoes dataset.

Methods	Disentangle ↑	FID ↓
MONet	0.286	230.3
MPP^{M} (Ours)	0.311	234.6
Genesis	0.302	144.1
MPP^{G} (Ours)	**0.362**	**128.9**

Table 1. Disentanglement ($\alpha = 0.001$) and FID scores of all models on Tetrominoes dataset.

the proposed prior, MPP^{M} improves MONet on the disentanglement task. MPP^{G} can significantly improve both the disentanglement and FID scores compared with Genesis.

5.5. CelebA Dataset

There are 202,599 images in the CelebA dataset. We use randomly selected 192,599 images for training and the rest 10,000 for testing. There are around 40 attributes, including gender, hair color, with glasses or not, etc. We use all the attributes to assess the disentanglement of all models.

Methods	Disentangle ↑	FID ↓
MONet	0.500	349.9
MPP^{M} (Ours)	0.510	180.8
Genesis	0.513	209.6
MPP^{G} (Ours)	**0.545**	**90.0**

Table 2. Disentanglement ($\alpha = 0.01$) and FID scores of all models on the CelebA dataset.

Table 2 gives the disentanglement score ($\alpha = 0.01$) using data sample attributes as the ground truth label. FID values are also listed for all four methods. Segmentation scores are not provided in the table because of the absence of ground truth segmentation labels in this dataset. From Table 2, we can see that the proposed message passing prior can significantly improve both the disentangled data representation learning and image generation quality.

5.6. ShapeStacks Dataset

In ShapeStacks dataset, images contain simulated block towers that consist of two to six blocks. The blocks have different shapes, sizes, and colors. Each image comes with annotations such as tower stability, the number of blocks, properties of the blocks, location of tower mass center, light presets, camera viewpoints, etc. More details can be found at [1]. Table 3 presents the FID scores for MONet, MPP^{M}, Genesis, and MPP^{G}. Results in the table show that, the proposed message passing prior can consistently improve the performance of existing methods on image generation.

Methods	MONet	MPP^{M} (Ours)	Genesis	MPP^{G} (Ours)
ShapeStacks	328.4	306.3	235.4	**196.7**

Table 3. Fréchet Inception Distances (FID ↓) for different methods on ShapeStacks.

5.7. Sensitivity Analysis on α

The disentanglement metric [7] used in this paper is sensitive to the hyper-parameter α. To study how sensitive our model is, we give the disentanglement scores of the models at different α values on Tetrominoes Dataset in Table 4.

Methods	MONet	MPP^{M} (Ours)	Genesis	MPP^{G} (Ours)
$\alpha = 0.001$	0.286	0.311	0.302	**0.362**
$\alpha = 0.010$	0.410	0.412	0.402	**0.460**
$\alpha = 0.050$	0.567	0.534	0.530	**0.584**

Table 4. Disentanglement scores (↑) at different α values.

We use the same experimental setup as in the Tetrominoes section. Table 4 shows the same pattern that a larger α values leads to larger disentanglement scores. However, MPP^{G} always provides the best disentanglement results at all α values. Moreover, MPP^{M} also performs better than MONet and Genesis at $\alpha = 0.001$ and $\alpha = 0.01$. The results in Table 4 indicate that the message passing prior improves the representation and feature learning from the data. More results on above datasets are provided in the supplemental file. These experimental results show that the proposed model indeed can improve the data representation learning and hence the generation and segmentation of different datasets.

6. Conclusions

We propose a novel bi-level framework to learn disentangled structured latent factors. The flow-based structure prior of latent presentation enables the model to learn interactions among components via a message-passing scheme. The framework can capture the inner interactions between data components in the experiments and improves disentanglement, segmentation as well as data generation. Besides the applications in this paper, there are potentially more scenarios where the proposed method is applicable. One future work following the proposed method is physical interaction extraction, which is an important common sense or prior knowledge for humans to make actionable decisions.

References

[1] Shapestacks. https://shapestacks.robots.ox.ac.uk/.

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

[3] Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. A multi-task approach for disentangling syntax and semantics in sentence representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2453–2464, Minneapolis, MN, 2019.

[4] Tian Qi Chen, Xuechen Li, Roger B. Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2615–2625, Montréal, Canada, 2018.

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2172–2180, Barcelona, Spain, 2016.

[6] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[7] Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.

[8] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: generative scene inference and sampling with object-centric latent representations. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020.

[9] S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3225–3233, Barcelona, Spain, 2016.

[10] Max H Farrell, Tengyuan Liang, and Sanjog Misra. Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213, 2021.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, Montreal, Canada, 2014.

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

[13] Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hotloo Hao, Harri Valpola, and Jürgen Schmidhuber. Tagger: Deep unsupervised perceptual grouping. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4484–4492, Barcelona, Spain, 2016.

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

[15] Hermanni Hälvä and Aapo Hyvärinen. Hidden markov nonlinear ICA: unsupervised learning from nonstationary time series. In *Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 939–948, virtual online, 2020.

[16] Andrew J. Hanson. *Graphics gems iv. chapter Geometry for N-dimensional Graphics*. Academic Press Professional, Inc, 1994.

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6626–6637, Long Beach, CA, 2017.

[18] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[19] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Fei-Fei Li, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 515–524, Montréal, Canada, 2018.

[20] Aapo Hyvärinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3765–3773, Barcelona, Spain, 2016.

[21] Aapo Hyvärinen and Hiroshi Morioka. Nonlinear ICA of temporally dependent stationary sources. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 460–469, Fort Lauderdale, FL, 2017.

[22] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1875–1885, New Orleans, LA, 2018.

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

[24] Ilyes Khemakhem, Diederik P. Kingma, Ricardo Pio Monti, and Aapo Hyvärinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *Proceedings of the*

23rd International Conference on Artificial Intelligence and Statistics (AISTATS), pages 2207–2217, Online [Palermo, Sicily, Italy], 2020.

[25] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 2654–2663, Stockholmsmässan, Stockholm, Sweden, 2018.

[26] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014.

[27] Steven Krantz and Harold Parks. Analytical tools: The area formula, the coarea formula, and poincaré inequalities. *Geometric Integration Theory*, pages 1–33, 2008.

[28] Joseph Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 3400–3409, Stockholmsmässan, Stockholm, Sweden, 2018.

[29] Shaogang Ren, Belhal Karimi, Dingcheng Li, and Ping Li. Variational flow graphical model. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1493–1503, Washington, DC, 2022.

[30] Shaogang Ren, Dingcheng Li, and Ping Li. Causal effect prediction with flow-based inference. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, Orlando, FL, 2022.

[31] Shaogang Ren, Dingcheng Li, Zhixin Zhou, and Ping Li. Estimate the implicit likelihoods of gans with application to anomaly detection. In *Proceedings of the Web Conference (WWW)*, pages 2287–2297, Taipei, 2020.

[32] Shaogang Ren and Ping Li. Flow-based perturbation for cause-effect inference. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*, Atlanta, GA, 2022.

[33] Shaogang Ren, Haiyan Yin, Mingming Sun, and Ping Li. Causal discovery with flow-based conditional density estimation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 1300–1305, Auckland, New Zealand, 2021.

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Part III*, pages 234–241, Munich, Germany, 2015.

[35] Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. In *Advances in Neural Information Processing Systems (NeurIPS)*, virtual, 2020.

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

[37] Theodore Voronov. *Geometric integration theory on supermanifolds*, volume 1. CRC Press, 1991.

[38] Guangrun Wang, Ping Luo, Liang Lin, and Xiaogang Wang. Learning object interactions and descriptions for semantic image segmentation. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5235–5243, Honolulu, HI, 2017.

[39] Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.

[40] Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

[41] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. DNA-GAN: learning disentangled representations from multi-attribute images. In *Proceedings of the 6th International Conference on Learning Representations (ICLR Workshop)*, Vancouver, Canada, 2018.