Theory and Evaluation Metrics for Learning Disentangled Representations

Kien Do and Truyen Tran
Applied AI Institute, Deakin University, Geelong, Australia
{dkdo,truyen.tran}@deakin.edu.au

Abstract

We make two theoretical contributions to disentanglement learning by (a) defining precise semantics of disentangled representations, and (b) establishing robust metrics for evaluation. First, we characterize the concept “disentangled representations” used in supervised and unsupervised methods along three dimensions— informativeness, separability and interpretability—which can be expressed and quantified explicitly using information-theoretic constructs. This helps explain the behaviors of several well-known disentanglement learning models. We then propose robust metrics for measuring informativeness, separability and interpretability. Through a comprehensive suite of experiments, we show that our metrics correctly characterize the representations learned by different methods and are consistent with qualitative (visual) results. Thus, the metrics allow disentanglement learning methods to be compared on a fair ground. We also empirically uncovered new interesting properties of VAE-based methods and interpreted them with our formulation. These findings are promising and hopefully will encourage the design of more theoretically driven models for learning disentangled representations.

1 Introduction

Disentanglement learning holds the key for understanding the world from observations, transferring knowledge across different tasks and domains, generating novel designs, and learning compositional concepts [5, 17, 24, 32, 39]. Assuming the observation $x$ is generated from latent factors $z$ via $p(x|z)$, the goal of disentanglement learning is to correctly uncover a set of independent factors $\{z_i\}$ that give rise to the observation. While there has been a considerable progress in recent years, common assumptions about disentangled representations appear to be inadequate [26].

Unsupervised disentangling methods are highly desirable as they assume no knowledge about the ground truth factors. These methods typically impose constraints to encourage independence among latent variables. Examples of constraints include forcing the variational posterior $q(z|x)$ to be similar to a factorial $p(z)$ [7, 16], forcing the variational aggregated prior $q(z)$ to be similar to the prior $p(z)$ [28], adding total correlation loss [18], forcing the covariance matrix of $q(z)$ to be close to the identity matrix [23], and using a kernel-based measure of independence [27]. However, it remains unclear how the independence constraint affects other properties of representation. Indeed, more independence may lead to higher reconstruction error in some models [16, 18]. Worse still, the independent representations may mismatch human’s predefined concepts [26]. This suggests that supervised methods – which associate a representation (or a group of representations) $z$ with a particular ground truth factor $y_k$ – may be more adequate. However, most supervised methods have only been shown to perform well on toy datasets [14, 22, 29] in which data are generated from multiplicative combination of the ground truth factors. It is still unclear about their performance on real datasets.
We believe that there are at least two major reasons for the current unsatisfying state of disentanglement learning: i) the lack of a formal notion of disentangled representations to support the design of proper objective functions [42][26], and ii) the lack of robust evaluation metrics to enable a fair comparison between models, regardless of their architectures or design purposes. To that end, we contribute by formally characterizing disentangled representations along three dimensions, namely informativeness, separability and interpretability, drawing from concepts in information theory (Section 3). We then design robust quantitative metrics for these properties and argue that an ideal method for disentanglement learning should achieve high performance on these metrics (Section 4).

We run a series of experiments to demonstrate how to compare different models using our proposed metrics, showing that the quantitative results provided by these metrics are consistent with visual results (Section 5). In the process, we gain important insights about some well-known disentanglement learning methods namely FactorVAE [18] and AAE [28].

2 Preliminaries

VAE-based methods Variational Autoencoder (VAE) [20][24] is a class of latent variable models, assuming a generative process as \( p(x, z) = p(z)p_0(x|z) \), where \( x \) denotes the observed data whose samples are drawn from the empirical distribution \( p_D(x) \) and \( z \) denotes the latent variable. Standard VAEs are trained by minimizing the variational upper bound \( L_{\text{VAE}} \) of the expected negative log-likelihood over the data. However, this objective function does not encourage disentanglement in representation. A simple solution is \( \beta \)-VAE [16], which modifies the objective as follows:

\[
L^{\beta-\text{VAE}} = \mathbb{E}_{p_D(x)} \left[ \mathbb{E}_{q_\phi(z|x)} \left[ -\log p_0(x|z) \right] \right] + \beta \mathbb{E}_{p_D(x)} \left[ D_{KL} \left( q_\phi(z|x) \parallel p(z) \right) \right]
\]

where \( \beta \gg 1 \) and \( q_\phi(z|x) \) is a parameterized variational estimator of the true posterior distribution \( p(z|x) \). When \( \beta = 1 \), \( L^{\beta-\text{VAE}} \) reduces to \( L^{\text{VAE}} \). Another proposal is FactorVAE [18], which adds a constraint to the standard VAE loss to explicitly impose factorization of \( q_\phi(z) \):

\[
L^{\text{FactorVAE}} = L^{\text{VAE}} + \gamma D_{KL} \left( q_\phi(z) \parallel \prod_i q_i(z_i) \right)
\]

where \( D_{KL} (q_\phi(z) \parallel \prod_i q_i(z_i)) \geq 0 \) is known as the total correlation (TC) of \( z \). Intuitively, \( \gamma \) can be large without affecting the mutual information between \( z \) and \( x \), making FactorVAE more robust than \( \beta \)-VAE in learning disentangled representations. Other variants include \( \beta \)-TCVAE [8] and DIP-VAE [23], which are technically equivalent to FactorVAE in the sense that they also force \( q_\phi(z) \) to be factorized.

InfoGAN Generative Adversarial Network (GAN) [13] is another class of generative models. GAN solves the minimax problem \( \min_G \max_D V^{\text{GAN}}(D, G) \) for a generator \( G \) and discriminator \( D \). InfoGAN [9] improves over GANs for learning disentangled representations. It assumes that the latent code vector \( z \) is a concatenation of two parts: a factorial part \( z_f \) and a noisy part \( z_n \), denoted as \( z = [z_f, z_n] \). InfoGAN learns to disentangle \( z_f \) by maximizing the mutual information between \( z_f \) and the observed data \( x \), using the following objective function:

\[
V^{\text{InfoGAN}}(D, G) = V^{\text{GAN}}(D, G) - \lambda I \left( z_f, G(z) \right)
\]

where \( I (\cdot, \cdot) \) denotes the mutual information; \( \lambda > 0 \); and \( V^{\text{Info}}(G, Q) \) is a lower bound of \( I \left( z_f, G(z) \right) \) [2]. Here, \( V^{\text{Info}}(G, Q) = \mathbb{E}_{z_f, z_n \sim p(z)} \mathbb{E}_{z \sim G([z_f, z_n])} \left[ \log Q(z_f|x) \right] \), where \( Q(z_f|x) \) is a variational estimator of the true conditional distribution \( p(z_f|x) \).

3 Rethinking Disentanglement

Inspired by [5][25], we adopt the notion of disentangled representation learning as “a process of decorrelating information in the data into separate informative representations, each of which corresponds to a concept defined by humans”. This suggests three important properties of a disentangled representation: informativeness, separability and interpretability, which we quantify as follows:
Informativeness. We formulate the informativeness of a particular representation $z_i$ w.r.t. the data $x$ as the mutual information between $z_i$ and $x$:

$$I(x, z_i) = \int_x \int_z p_D(x) q(z_i|x) \log \frac{q(z_i|x)}{q(z_i)} \, dz \, dx$$

(4)

where $q(z_i) = \int_x p_D(x) q(z_i|x) \, dx$. In order to represent the data faithfully, a representation $z_i$ should be informative of $x$, meaning $I(x, z_i)$ should be large. Because $I(x, z_i) = H(z_i) - H(z_i|x)$, a large value of $I(x, z_i)$ means that $H(z_i|x) \approx 0$ given that $H(z_i$) can be chosen to be relatively fixed. In other words, if $z_i$ is informative w.r.t. $x$, $q(z_i|x)$ usually has small variance. It is important to note that $I(x, z_i)$ in Eq. (4) is defined on the variational encoder $q(z_i|x)$, and does not require a decoder. This implies we do not need to minimize the reconstruction error over $x$ (e.g., in VAEs) to increase the informativeness of a particular $z_i$.

Separability and Independence. Two representations $z_i, z_j$ are separable w.r.t. the data $x$ if they do not share common information about $x$, which can be formulated as follows:

$$I(x, z_i, z_j) = 0$$

(5)

where $I(x, z_i, z_j)$ denotes the multivariate mutual information [31] between $x, z_i$ and $z_j$. $I(x, z_i, z_j)$ can be decomposed into standard bivariate mutual information terms as follows:

$$I(x, z_i, z_j) = I(x, z_i) + I(x, z_j) - I(x, (z_i, z_j)) = I(z_i, z_j) - I(z_i, z_j|x)$$

$I(x, z_i, z_j)$ can be either positive or negative. It is positive if $z_i$ and $z_j$ contain redundant information about $x$. The meaning of a negative $I(x, z_i, z_j)$ remains elusive [4]. Achieving separability with respect to $x$ does not guarantee that $z_i$ and $z_j$ are separable in general. $z_i$ and $z_j$ are fully separable or statistically independent if and only if:

$$I(z_i, z_j) = 0$$

(6)

Let us consider how FactorVAE and InfoGAN implement this independence requirement. FactorVAE enforces the independence in Eq. (5) for every pair of $z_i, z_j$ via the TC term (see Eq. (1)). In InfoGAN, the existence of such condition is not clear. However, if we look closely into the term $V^{Info}(G, Q)$ in Eq. (3) it is actually the reconstruction error over $z_i, z_j$ sampled from a factorial prior $p(z_i)p(z_j)$. By minimizing this term, we will force $q(z_i)$ close to $p(z_i)$ and $q(z_j)$ close to $p(z_j)$, making $q(z_i)$ and $q(z_j)$ independent. Note that $q(z_i, z_j)$ are derived from the assumption $p(x, z_i, z_j) = p_G(x)q(z_i, z_j|x)$ where $p_G(x)$ is the implicit generative distribution of $x$ by transforming $p(z)$ via the generator $G$. The original GAN objective in InfoGAN assumes that $p_G(x)$ matches the empirical data distribution $p_D(x)$. However, when this assumption does not hold, $q(z_i, z_j) = \mathbb{E}_{p_G(x)}[q(z_i, z_j|x)]$ is not really grounded on real data making it hard to interpret the independence.

Note that there is a trade-off between informativeness, independence and the number of latent variables which we discuss in Appdx. A5.

Interpretability. Obtaining informative and independent representations does not guarantee interpretability by human [26]. We argue that in order to achieve interpretability, we should provide models with a set of predefined concepts $y$. In this case, a representation $z_i$ is interpretable with respect to $y_k$ if it only contains information about $y_k$ (given that $z_i$ is separable from all other $z_{\neq i}$ and all $y_k$ are distinct). Full interpretability can be formulated as follows:

$$I(z_i, y_k) = H(z_i) = H(y_k)$$

(7)

Eq. (7) is equivalent to the condition that $z_i$ is an invertible function of $y_k$. If we want $z_i$ to generalize beyond the observed $y_k$ (i.e., $H(z_i) > H(y_k)$), we can change the condition in Eq. (7) into:

$$I(z_i, y_k) = H(y_k) \text{ or } H(y_k|z_i) = 0$$

(8)

which suggests that the model should accurately predict $y_k$ given $z_i$. If $z_i$ satisfies the condition in Eq. (8) it is said to be partially interpretable w.r.t $y_k$.

In real data, underlying factors of variation are usually correlated. For example, men usually have beard and short hair. Therefore, it is very difficult to match independent latent variables to different ground truth factors at the same time. We believe that in order to achieve good interpretability, we should isolate the factors and learn one at a time.
3.1 An information-theoretic definition of disentangled representations

Given a dataset \( D = \{x_i\}_{i=1}^N \), where each data point \( x \) is associated with a set of \( K \) labeled factors of variation \( y = \{y_1, \ldots, y_K\} \). Assume that there exists a mapping of \( x \) to \( m \) groups of hidden representation \( z = \{z_1, z_2, \ldots, z_m\} \) which follows the distribution \( q(z|x) \). Denoting \( q(z_i|x) = \sum_{z_{\neq i}} q(z|x) \) and \( q(z_i) = E_{p_{x}(x)}[q(z_i|x)] \). We define disentangled representations for unsupervised cases as follows:

**Definition 1** (Unsupervised). A representation or a group of representations \( z_i \) is said to be "fully disentangled" w.r.t a ground truth factor \( y_k \) if \( z_i \) is marginally independent of all other representations \( z_{\neq i} \), and \( z_i \) is fully interpretable w.r.t \( y_k \). Mathematically, this can be written as:

\[
I(z_i, z_{\neq i}) = 0 \quad \text{and} \quad I(z_i, y_k) = H(z_i, y_k)
\]

where

\[
q(z) = \sum_x p_D(x)q(z|x)
\]

\[
I(z_i, z_{\neq i}) = \sum_z q(z) \log \frac{q(z)}{q(z_i)q(z_{\neq i})}
\]

\[
I(z_i, y_k) = \sum_y \sum_z q(z_i, y_k) \log \frac{q(z_i, y_k)}{q(z_i)q(y_k)}
\]

\[
H(z_i, y_k) = -\sum_y \sum_z q(z_i, y_k) \log q(z_i, y_k)
\]

The definition of disentangled representations for supervised cases is similar as above except that now we model \( q(z|x, y) \) instead of \( q(z|x) \) and \( q(z) = \sum_{x,y} p_D(x, y)q(z|x, y) \).

Recently, there have been several works \([12,15,36]\) that attempted to define disentangled representations. Higgins et. al. \([15]\) proposed a definition based on group theory \([10]\) which is (informally) stated as follows: “A representation \( z \) is disentangled w.r.t a particular subgroup \( y_k \) from a symmetry group \( y = \{y_1, \ldots, y_K\} \) if \( z \) can be decomposed into different subspaces \( \{z_i\}_{i=1}^H \) in which the subspace \( z_i \) should be independent of all other representation subspaces \( z_{\neq i} \), and \( z_i \) should only be affected by the action of a single subgroup \( y_k \) and not by other subgroups \( y_{\neq k} \)." Their definition shares similar observation as ours. However, it is less convenient for designing models and metrics than our information-theoretic definition.

Eastwood et. al. \([12]\) did not provide any explicit definition of disentangled representation but characterizing it along three dimensions namely “disentanglement”, “compactness”, and “informativeness” (between \( z \) any \( y_k \)). A high “disentanglement” score \((\approx 1)\) for \( z_i \) indicates that it captures at most one factor, let’s say \( y_k \). A high “compactness” score \((\approx 1)\) for \( y_k \) indicates that it is captured by at most one latent \( z_j \) and \( j \) is likely to be \( i \). A high “informativeness” score \([3]\) for \( y_k \) indicates that all information of \( y_k \) is captured by the representations \( z \). Intuitively, when all the three notions achieve optimal values, there should be only a single representation \( z_i \) that captures all information of the factor \( y_k \) but no information from other factors \( y_{\neq k} \). However, even in that case, \( z_i \) is still not fully interpretable w.r.t \( y_k \) since \( z_i \) may contain some information in \( x \) that does not appear in \( y \). This makes their notions only applicable to toy datasets on which we know that the data \( x \) are only generated from predefined ground truth factors \( y \). Our definition can handle the situation where we only know some but not all factors of variation in the data. The notions in \([36]\) follow those in \([12]\), hence, suffer from the same disadvantage.

3.2 Representations learned by FactorVAE

We empirically observed that FactorVAE learns the same set of disentangled representations across different runs with varying numbers of latent variables (see Appdx. A8). This behavior is akin to that of deterministic PCA which uncovers a fixed set of linearly independent factors (or principal
components). Standard VAE is theoretically similar to probabilistic PCA (pPCA) \(^{41}\) as both assume the same generative process \(p(x, z) = p_\theta(x|z)p(z)\). Unlike deterministic PCA, pPCA learns a rotation-invariant family of factors instead of an identifiable set of factors. However, in a particular pPCA model, the relative orthogonality among factors is still preserved. This means that the factors learned by different pPCA models are statistically equivalent. We hypothesize that by enforcing independence among latent variables, FactorVAE can also learn statistically equivalent factors (or \(q(z_i|x)\)) which correspond to visually similar results. We provide a proof sketch for the hypothesis in Appdx. A6. We note that Rolinek et. al. \(^{37}\) also discovered the same phenomenon in \(\beta\)-VAE.

4 Robust Evaluation Metrics

We argue that a robust metric for disentanglement should meet the following criteria: i) it supports both supervised/unsupervised models; ii) it can be applied for real datasets; iii) it is computationally straightforward, i.e. not requiring any training procedure; iv) it provides consistent results across different methods and different latent representations; and v) it agrees with qualitative (visual) results. Here we propose information-theoretic metrics to measure informativeness, independence and interpretability which meet all of these robustness criteria.

4.1 Metrics for informativeness

We measure the informativeness of a particular representation \(z_i\) w.r.t. \(x\) by computing \(I(x, z_i)\) in Eq. 3. The main challenges are estimating \(q(z_i)\) and computing the integral over \(z_i\). We deal with these problems by quantizing \(z_i\). To ensure \(I(x, z_i)\) to be consistent and comparable among different \(z_i\) as well as different models, we apply the same quantization range for different \(z_i\). In practice, we choose the range \([-4, 4]\) since most of the latent values fall within this range. We divide the range into a set \(\mathcal{S}\) of equal-size bins and estimate \(I(x, z_i)\) as follows:

\[
I(x, z_i) = \frac{1}{N} \sum_{n=1}^{N} \left[ \sum_{s_i \in \mathcal{S}} Q(s_i|x^{(n)}) \left( \log Q(s_i|x^{(n)}) - \log Q(s_i) \right) \right]
\]

(10)

where \(Q(s_i)\) and \(Q(s_i|x^{(n)})\) are the probability mass function and the conditional probability mass function of a particular bin \(s_i\). Because \(Q(s_i) = \frac{1}{N} \sum_{n=1}^{N} Q(s_i|x^{(n)})\) we only have to compute \(Q(s_i|x^{(n)})\), which by definition, is:

\[
Q(s_i|x^{(n)}) = \int_{a}^{b} q(z_i|x^{(n)})\,dz_i
\]

(11)

where \(a, b\) are two ends of the bin \(s_i\).

There are two ways to compute \(Q(s_i|x^{(n)})\). In the first way, we simply consider the unnormalized \(Q'(s_i|x^{(n)})\) as the area of a rectangle whose width is \(b - a\) and height is \(q(\tilde{z}_i|x^{(n)})\) with \(\tilde{z}_i\) at the center value of the bin \(s_i\). Then, we normalize \(Q' (s_i|x^{(n)})\) over all bins to get \(Q(s_i|x^{(n)})\). In the second way, if \(q(z_i|x^{(n)})\) is approximately a Gaussian distribution, we can estimate the above integral with a closed-form function (see Appdx. A12 for detail). After computing \(I(x, z_i)\), we can divide it by \(H(z_i) = -\sum_{s_i \in \mathcal{S}} Q(s_i) \log Q(s_i)\) to normalize it to the range \([0, 1]\). However this normalization will change the interpretation of the metric and may lead to a situation where latent variable \(z_i\) is less informative than variable \(z_j\) (i.e., \(I(x, z_i) < I(x, z_j)\)) but still has a higher rank than \(z_j\) because \(H(z_i) < H(z_j)\). A better way is to divide it by \(\log |\mathcal{S}|\) where \(|\mathcal{S}|\) denotes the number of bins. An important note for implementation is that sometimes, the standard deviation of \(q(z_i|x^{(n)})\) is close to 0 (or \(z_i\) is deterministic given \(x^{(n)}\)), causing \(q(\tilde{z}_i|x^{(n)})\) to be close to 0 for all bins. In this case, we set \(Q(s_i|x^{(n)}) = 1\) if \(s_i\) is the bin that contains the mean of \(q(z_i|x^{(n)})\) and 0 otherwise.

\(\text{\textsuperscript{3}}\)We must take into account the whole quantized distribution \(Q(s_i|x^{(n)})\). Simply counting the quantized mean \(\mathbb{E}_{q(z_i|x^{(n)})}[z_i]\) for all \(x^{(n)}\) is totally wrong.
4.2 Metrics for independence

We can compute the independence between two latent variables \( z_i, z_j \) based on \( I(z_i, z_j) \). However, a serious problem of \( I(z_i, z_j) \) is that it generates the following order among pairs of representations:

\[
I(z_{f,i}, z_{f,j}) > I(z_{f,i}, z_{n,j}) > I(z_{n,i}, z_{n,j}) \geq 0
\]

where \( z_{f,i}, z_{f,j} \) are informative representations and \( z_{n,i}, z_{n,j} \) are uninformative (or noisy) representations. This means if we simply want \( z_i, z_j \) to be independent, the best scenario is that \textit{both are noisy and independent} (e.g. \( q(z_i|x) \approx q(z_j|x) \approx \mathcal{N}(0,1) \)). Therefore, we propose a new metric for independence named \textsc{MISJED} (which stands for Mutual Information Sums Joint Entropy Difference), defined as follows:

\[
\textsc{MISJED}(z_i, z_j) = \bar{I}(z_i, z_j) = H(z_i) + H(z_j) - H(\bar{z}_i, \bar{z}_j) \\
= H(z_i) + H(z_j) - H(z_i, z_j) + H(z_i, z_j) - H(\bar{z}_i, \bar{z}_j) \\
= I(z_i, z_j) + H(z_i, z_j) - H(\bar{z}_i, \bar{z}_j)
\]

where \( \bar{z}_i \) and \( \bar{z}_j \) are the means of \( q(z_i|x) \) and \( q(z_j|x) \), respectively. Since \( q(\bar{z}_i), q(\bar{z}_j) \) have less variance than \( q(z_i), q(z_j) \) respectively, \( H(z_i, z_j) - H(\bar{z}_i, \bar{z}_j) \geq 0 \), making \( \bar{I}(z_i, z_j) \geq 0 \).

To achieve a small value of \( \bar{I}(z_i, z_j) \), i.e., a high degree of independence, we must have representations \( z_i, z_j \) to be both \textit{independent} and \textit{informative} (or, in an extreme case, are deterministic given \( x \)). Using the \textsc{MISJED} metric, we can ensure the following order: \( 0 \leq \bar{I}(z_{f,i}, z_{f,j}) < \bar{I}(z_{f,i}, z_{n,j}) < \bar{I}(z_{n,i}, z_{n,j}) \).

Because \( \bar{I}(z_i, z_j) \leq H(z_i) + H(z_j) \leq 2 \log |S| \), we can divide \( \bar{I}(z_i, z_j) \) by \( 2 \log |S| \) to normalize it to \([0, 1]\).

4.3 Metrics for interpretability

Recently, several metrics have been proposed to quantitatively evaluate the interpretability of representations by examining the relationship between the representations and manually labeled factors of variation. The most popular ones are Z-diff score [16][18], SAP [23] and MIG [8]. Detailed analysis of these metrics is provided in Appdx. A9. Among them, only MIG is based on mutual information and, to some extent, matches with the formulation of "interpretability" in Section 3.

However, MIG has only been used for toy datasets like dSprites [30]. The main drawback comes from its probabilistic assumption \( p(z_i, y_k, x^{(n)}) = q(z_i|x^{(n)})p(x^{(n)}|y_k)p(y_k) \) (see Fig. 1). Note that \( p(x^{(n)}|y_k) \) is a distribution over the high dimensional data space, and is very hard to robustly estimate but the authors simplified it to be \( p(n|y_k) \) if \( x^{(n)} \in \mathcal{D}_{y_k} (\mathcal{D}_{y_k} \) is the support set for a particular value \( y_k \)) and 0 otherwise. This equation only holds for toy datasets where we know exactly how \( x \) is generated from \( y \). In addition, since \( p(n|y_k) \) depends on the value of \( y_k \), it will be problematic if \( y_k \) is continuous.

\textbf{RMIG} Addressing the drawbacks of MIG, we propose RMIG (which stands for Robust MIG), formulated as follows:

\[
\text{RMIG}(y_k) = I^*(z_{i^*}, y_k) - I^*(z_{j^*}, y_k)
\]

(12)

where \( I^*(z_{i^*}, y_k) \) and \( I^*(z_{j^*}, y_k) \) are the highest and the second highest mutual information values computed between every \( z_i \) and \( y_k \); \( z_{i^*} \) and \( z_{j^*} \) are the corresponding latent variables. Like MIG, we can normalize \( \text{RMIG}(y_k) \) to \([0, 1]\) by dividing it by \( H(y_k) \) but it will favor imbalanced factors (small \( H(y_k) \)). Details of computation are given in Appdx. A10.
We evaluated the performance of FactorVAE [18], β-VAE [16] and AAE [28] using our proposed metrics on the CelebA [25], MNIST and dSprites [30] datasets. Details about the datasets and model settings are provided in Appdx. A1 and Appdx. A2, respectively. For space limit, we only report here results on CelebA, leaving the rest in the supplementary materials.

5 Experiments

We evaluated the performance of FactorVAE [18], β-VAE [16] and AAE [28] using our proposed metrics on the CelebA [25], MNIST and dSprites [30] datasets. Details about the datasets and model settings are provided in Appdx. A1 and Appdx. A2, respectively. For space limit, we only report here results on CelebA, leaving the rest in the supplementary materials.

Informativeness We sorted the representations of different models according to their informativeness scores in the descending order and plot the results in Fig. 2. There are distinct patterns for different methods. AAE captures equally large amounts of information from the data while FactorVAE and β-VAE capture smaller and varying amounts. This is because FactorVAE and β-VAE penalize the informativeness of representations while AAE does not. Recall that \( I(z_i, x) = H(z_i) - H(z_i | x) \). For AAE, \( H(z_i | x) = 0 \) and \( H(z_i) \) is equal to the entropy of \( \mathcal{N}(0, 1) \). For FactorVAE and β-VAE, \( H(z_i | x) > 0 \) and \( H(z_i) \) is usually smaller than the entropy of \( \mathcal{N}(0, 1) \) due to a narrow \( q(z_i) \).

In Fig. 2 we see a sudden drop of the scores to 0 for some FactorVAE’s and β-VAE’s representations. These representations \( z_i \) are totally random and contain no information about the data (i.e., \( q(z_i | x) \approx \mathcal{N}(0, 1) \)). We call them “noisy” representations and provide discussions in Appdx. A5.

---

Note that \( H(z_i) \) does not depend on whether \( q(z_i) \) is zero-centered or not.
Figure 3: Visualization of the top informative representations. Scores are unnormalized.

|                      | $z_1, z_2$ | $z_1, z_3$ | $z_1, z_{-1}$ | $z_1, z_{-2}$ | $z_{-1}, z_{-2}$ | $z_{-1}, z_{-3}$ |
|----------------------|------------|------------|---------------|---------------|-----------------|-----------------|
| FactorVAE            | 0.008      | 0.009      | 2.476         | 2.443         | 4.858           | 4.892           |
| $\beta$-VAE          | 0.113      | 0.131      | 3.413         | 3.401         | 6.661           | 6.739           |
| AAE                  | 0.022      | 0.023      | 0.022         | 0.021         | 0.021           | 0.020           |

Table 1: Unnormalized MISJED scores (#bins = 50, 10% data). $z_1, z_2, z_3$ and $z_{-1}, z_{-2}, z_{-3}$ denote the top 3 and the bottom 3 latent variables sorted by the informativeness scores in descending order. Boldness indicates best results.

We visualize the top 10 most informative representations for these models in Fig. 3. AAE’s representations are more detailed than FactorVAE’s and $\beta$-VAE’s, suggesting the effect of high informativeness. However, AAE’s representations mainly capture information within the support of $p_D(x)$. This explains why we still see a face when interpolating AAE’s representations. By contrast, FactorVAE’s and $\beta$-VAE’s representations usually contain information outside the support of $p_D(x)$. Thus, when we interpolate these representations, we may see something not resembling a face.

**Independence** Table 1 reports MISJED scores (Section 4.2) for the top most informative representations. FactorVAE achieves the lowest MISJED scores, AAE comes next and $\beta$-VAE is the worst. We argue that this is because FactorVAE learns independent and nearly deterministic representations, $\beta$-VAE learns strongly independent yet highly stochastic representations, and AAE, on the other extreme side, learns strongly deterministic yet not very independent representations. From Table 1 and Fig. 4, it is clear that MISJED produces correct orders among pairs of representations according to their informativeness.

**Interpretability** We report the RMIG scores and JEMMI scores for several ground truth factors on the CelebA dataset in Tables 2 and 3 respectively. In general, FactorVAE learns representations that agree better with the ground truth factors than $\beta$-VAE and AAE do. This is consistent with the
Table 2: Normalized RMIG scores (#bins=100, 100% data) for some factors. Higher is better.

|        | RMIG (normalized) |        |        |        |        |
|--------|-------------------|--------|--------|--------|--------|
|        | Bangs H=0.4256    | Black Hair H=0.5500 | Eyeglasses H=0.2395 | Goatee H=0.2365 | Male H=0.6801 | Smiling H=0.6923 |
| FactorVAE | **0.1742** | **0.0430** | **0.0409** | **0.0343** | 0.0060 | **0.0962** |
| β-VAE   | 0.0176 | 0.0223 | 0.0045 | 0.0325 | 0.0094 | 0.0184 |
| AAE     | 0.0035 | 0.0276 | 0.0018 | 0.0069 | 0.0060 | 0.0099 |

Table 3: Normalized JEMMI scores (#bins=100, 100% data) for some factors. Lower is better.

|        | JEMMI (normalized) |        |        |        |        |
|--------|-------------------|--------|--------|--------|--------|
|        | Bangs H=0.4256    | Black Hair H=0.5500 | Eyeglasses H=0.2395 | Goatee H=0.2365 | Male H=0.6801 | Smiling H=0.6923 |
| FactorVAE | **0.6118** | **0.6334** | **0.6041** | **0.6616** | **0.6875** | **0.6150** |
| β-VAE   | 0.8632 | 0.8620 | 0.8602 | 0.8600 | 0.8690 | 0.8699 |
| AAE     | 0.8463 | 0.8613 | 0.8423 | 0.8496 | 0.8644 | 0.8575 |

Sensitivity of the number of bins All metrics we propose in this paper require computing the mutual information (MI). To handle continuous cases, we use quantization. It is important to note that quantization is just a trick for computing MI, not the inherent problem of our metrics. With quantization, we need to specify the number of bins (#bins) in advance. Fig. 6 (left, middle) shows the effect of #bins on RMIG scores and JEMMI scores for different models.

We can see that when #bins is small, RMIG scores are low. This is because the quantized distributions \( Q(z_{i*}) \) and \( Q(z_{j\circ}) \) look similar, causing \( I^{*}(z_{i*}, y_k) \) and \( I^{\circ}(z_{j\circ}, y_k) \) to be similar as well. When #bins is large, the quantized distribution \( Q(z_{i*}) \) and \( Q(z_{j\circ}) \) look more different, leading to higher RMIG scores. RMIG scores are stable when #bins > 200.

Unlike RMIG scores, JEMMI scores keep increasing when we increase #bins. Note that JEMMI only differs from RMIG in the appearance of \( H(z_{i*}, y_k) \). Finer quantizations of \( z_{i*} \) introduce more information about \( z_{i*} \), hence, always lead to higher \( H(z_{i*}, y_k) \) (see Fig. 6 (right)). Larger JEMMI scores also reflect the fact that finer quantizations of \( z_{i*} \) make \( z_{i*} \) look more continuous, thus, less interpretable w.r.t. the discrete factor \( y_k \).

Despite the fact that #bins affects the RMIG and JEMMI scores of a single model, the relative order among different models remains the same. It suggests that once we fixed the #bins, we can use RMIG and JEMMI scores to compare different models.

6 Discussion

We have proposed information-theoretic characterizations of disentangled representations, and designed robust metrics for evaluation, along three dimensions: informativeness, separability and interpretability. We examined three well-known representation learning models namely FactorVAE, β-VAE and AAE on CelebA, MNIST and dSprites datasets. Under our metrics, FactorVAE is the best among the three, with reasonably good informativeness and very good MISJED scores. In addition, FactorVAE also learns consistent representations. However, all the examined models still perform poorly under our metric for interpretability, meaning that they have not met desirable requirements for disentanglement learning. Our work also shows that unsupervised disentanglement may not be possible, and that labels of ground truth factors should be provided during learning. Thus, we plan to investigate methods which support semi-supervised or few-shot learning in the future.

References

[1] Error function. https://en.wikipedia.org/wiki/Error_function, May 2019.
Figure 5: Top 10 representations that are most correlated with some ground truth factors. For each representation, we show its mutual information with the ground truth factor.

Figure 6: Dependences of RMIG (normalized), JEMMI (normalized) and $H(z,y)$ on the number of bins. We examined different FactorVAE and $\beta$-VAE models on the dSprite dataset.

[2] David Barber Felix Agakov. The im algorithm: A variational approach to information maximization. Advances in Neural Information Processing Systems, 16:201, 2004.

[3] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. arXiv preprint arXiv:1612.00410, 2016.

[4] Anthony J Bell. The co-information lattice. In Proceedings of the Fifth International Workshop on Independent Component Analysis and Blind Signal Separation: ICA, volume 2003, 2003.

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1798–1828, 2013.

[6] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. arXiv preprint arXiv:1509.00519, 2015.

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

[8] Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. arXiv preprint arXiv:1802.04942, 2018.
[9] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[10] Taco Cohen and Max Welling. Learning the irreducible representations of commutative lie groups. In *International Conference on Machine Learning*, pages 1755–1763, 2014.

[11] Guillaume Desjardins, Aaron Courville, and Yoshua Bengio. Disentangling factors of variation via generative entangling. *arXiv preprint arXiv:1210.5474*, 2012.

[12] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. 2018.

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

[14] Ananya Harsh Jha, Saket Anand, Maneesh Singh, and VSR Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 805–820, 2018.

[15] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

[16] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[17] Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P Burgess, Matko Bosnjak, Murray Shanahan, Matthew Botvinick, Demis Hassabis, and Alexander Lerchner. Scan: Learning hierarchical compositional visual concepts. *arXiv preprint arXiv:1707.03389*, 2017.

[18] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *ICML*, 2018.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[21] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[22] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.

[23] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

[24] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[26] Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *ICML*, 2019.

[27] Romain Lopez, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Information constraints on auto-encoding variational bayes. In *Advances in Neural Information Processing Systems*, pages 6114–6125, 2018.

[28] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.
[29] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In Advances in Neural Information Processing Systems, pages 5040–5048, 2016.

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

[31] William McGill. Multivariate information transmission. Transactions of the IRE Professional Group on Information Theory, 4(4):93–111, 1954.

[32] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Elements of causal inference: foundations and learning algorithms. MIT press, 2017.

[33] Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In International Conference on Machine Learning, pages 1431–1439, 2014.

[34] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082, 2014.

[35] Karl Ridgeway. A survey of inductive biases for factorial representation learning. arXiv preprint arXiv:1612.05299, 2016.

[36] Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. In Advances in Neural Information Processing Systems, pages 185–194, 2018.

[37] Michal Rolinek, Dominik Zietlow, and Georg Martius. Variational autoencoders pursue pca directions (by accident). arXiv preprint arXiv:1812.06775, 2018.

[38] Adrià Ruiz, Oriol Martinez, Xavier Binefa, and Jakob Verbeek. Learning disentangled representations with reference-based variational autoencoders. arXiv preprint arXiv:1901.08534, 2019.

[39] Jürgen Schmidhuber. Learning factorial codes by predictability minimization. Neural Computation, 4(6):863–879, 1992.

[40] N Siddharth, Brooks Paige, Jan-Willem van de Meent, Alban Desmaison, Noah D Goodman, Pushmeet Kohli, Frank Wood, and Philip HS Torr. Learning disentangled representations with semi-supervised deep generative models. NIPS, 2017.

[41] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 61(3):611–622, 1999.

[42] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. arXiv preprint arXiv:1812.05069, 2018.

[43] Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaowu Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In Advances in Neural Information Processing Systems, pages 217–225, 2014.
A Appendix

A.1 Datasets

We used CelebA, MNIST and dSprites datasets. For CelebA, we resized the original images to 64 × 64. Dataset statistics are provided in Table 4.

| Dataset  | #Train  | #Test  | Image size |
|----------|---------|--------|------------|
| CelebA   | 162,770 | 19,962 | 64 × 64 × 3 |
| MNIST    | 60,000  | 10,000 | 28 × 28 × 1 |
| dSprites | 737,280 | 0      | 64 × 64 × 1 |

Table 4: Summary of datasets used in experiments.

A.2 Model settings

For FactorVAE, β-VAE and AAE we used the same architectures for the encoder and decoder (see Table 5 and Table 6), following [18]. We trained the models for 300 epochs with mini-batches of size 64. The learning rate is $10^{-3}$ for the encoder/decoder and is $10^{-4}$ for the discriminator over $z$. We used Adam [19] optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.99$. Unless explicitly mentioned, we fixed the following: number of latent variables to 65, coefficient for the TC term in FactorVAE to 50, value for $\beta$ in $\beta$-VAE to 50, and coefficient for the generator loss over $z$ in AAE to 50.

We note that the FactorVAE in [18] only used 10 latent variables for learning factors of variation on the CelebA dataset so our results may look different from theirs. However, by using larger numbers of latent variables, we are able to discover that FactorVAE learns consistent representations (see Appdx. [A.8]).

| Encoder | Decoder | Discriminator Z |
|---------|---------|-----------------|
| $x$ dim: 64 × 64 × 3 | $z$ dim: 65 | $z$ dim: 65 |
| conv (4, 4, 32), stride 2, ReLU | FC 1 × 1 × 256, ReLU | 5 × [FC 1000, LReLU] |
| conv (4, 4, 32), stride 2, ReLU | deconv (4, 4, 64), stride 1, valid, ReLU | FC 1 |
| conv (4, 4, 64), stride 2, ReLU | deconv (4, 4, 64), stride 2, ReLU | $D(z)$: 1 |
| conv (4, 4, 64), stride 2, ReLU | deconv (4, 4, 32), stride 2, ReLU | |
| conv (4, 4, 256), stride 1, valid, ReLU | deconv (4, 4, 32), stride 2, ReLU | |
| FC 65 | deconv (4, 4, 3), stride 2, ReLU | $x$ dim: 64 × 64 × 3 |

Table 5: Model architectures for CelebA.

| Encoder | Decoder | Discriminator Z |
|---------|---------|-----------------|
| $x$ dim: 28 × 28 × 1 | $z$ dim: 65 | $z$ dim: 65 |
| conv (4, 4, 64), stride 2, LReLU | FC 1024, BN, ReLU | 4 × [FC 256, LReLU] |
| conv (4, 4, 128), stride 2, BN, LReLU | FC 7 × 7 × 128, BN, ReLU | FC 1 |
| FC 1024, BN, LReLU | deconv (4, 4, 64), stride 2, BN, ReLU | $D(z)$: 1 |
| FC 128, BN, LReLU | deconv (4, 4, 1), stride 2, sigmoid | |
| FC 65, BN, LReLU | $x$ dim: 28 × 28 × 1 | |
| $z$ dim: 65 | | |

Table 6: Model architecture for MNIST.

5 Only FactorVAE and AAE use a discriminator over $z$
A.3 Reviews of disentanglement learning methods

There has been many works that attempt to learn disentangled representations, which significantly differ in approaches and generalities. However, there are lacks of consensus on many key aspects [26], including definition [15].

**Supervised methods** [22, 43] assume access to the ground truth factors. For example, DC-IGN [22] is a VAE whose latent variables \( z_k \) correspond to different ground truth factors \( y_k \). At each training step, DC-IGN chooses a mini-batch with a ground truth factor \( y_k \) varies while other factors \( y_{\neq k} \) are fixed. Then, they only allow the latent variable \( z_k \) corresponding to the selected factor \( y_k \) to capture the variation in the mini-batch by replacing all other latent variables \( z_{\neq k} \) with their mean values over the mini-batch. This “clamping” strategy was also applied in [33] to improve the disentanglement capability of a higher-order Boltzmann Machine. Mathieu et. al. [29] proposed a conditional VAE that models both labeled factors of variation \( s \) and other unspecified latent representations \( z \). Since \( s \) is given, this model simply learns \( z \), which is assumed to be entangled. To ensure the decoder does not ignore labeled information from \( s \), the authors swap the unspecified latent representations \( z^{(1)}, z^{(2)} \) of two samples \( x^{(1)}, x^{(2)} \) and use an additional GAN to force the images generated from \( (s^{(2)}, z^{(1)}) \) and \( (s^{(2)}, z) \) where \( z \sim \mathcal{N}(0, 1) \) to be similar. The main problem of this method is that none of the generated images are fixed, which results in unstable training of GAN as reported in [29]. Other methods that are derived from [29] include [14, 38].

**Unsupervised methods** learn disentangled representations directly from raw data without using knowledge about the ground truth factors of variation. Desjardins et. al. [11] made an early attempt at unsupervised disentanglement learning by using a higher-order spike-and-slab RBM with block-sparse connectivity to model the multiplicative interactions between (unknown) factors of variation. Despite some success on the Toronto Face dataset, this method has two main drawbacks that make it impractical: one is its modeling complexity and the other is its oversimplified assumption about the multiplicative interactions between factors. Current state-of-the-art unsupervised methods are based on powerful deep generative models such as GAN [9] or VAE [8, 16, 18, 23]. They show promising disentanglement results on many real datasets and are scalable. The key idea behind these methods is learning independent yet informative representations.

**Semi-supervised methods** have also been proposed. Kingma et. al. [21] proposed two variants of VAE to solve the semi-supervised learning problem. One variant denoted as M1 adds a classifier on top of \( z \sim q_\phi(z|x) \) to predict the label \( y \). The other variant denoted as M2 assumes a generative model \( p_\theta(x, y, z) = p_\theta(x|y, z)p(y)p(z) \) with the inference networks for \( y \) and \( z \) are \( q_\phi(y|x) \) and \( q_\phi(z|x, y) \), respectively. This M2 model is able to separate between style and content by using very little amount of labeled data (about 1-5\% of the total data). Siddharth et. al. [49] replace the variational objective of the M2 model with the importance-weighted loss [5] so that \( q_\phi(y, z|x) \) can have arbitrary conditional dependency between \( y \) and \( z \) instead of just the decomposition \( q_\phi(y, z|x) = q_\phi(y|x)q_\phi(z|x, y) \) used in [21]. However, this does not lead to any significant change in the model architecture.

A.4 Evaluating independence with correlation matrix

For every \( x^{(n)} \) sampled from the training data, we generated \( m = 1 \) latent samples \( z^{(n,m)} \sim q(z|x^{(n)}) \) and built a correlation matrix from these samples for each of the models FactorVAE, \( \beta \)-VAE and AAE. We also built another version of the correlation matrix which is based on the
\(\mathbb{E}_{q(z_i|x_n)}[z_i]\) (called the \textit{conditional means}) instead of samples from \(q(z_i|x_n)\). Both are shown in Fig. 7. We can see that the correlation matrices computed based on the conditional means \textit{incorrectly} describe the independence between representations of FactorVAE and \(\beta\)-VAE. AAE is not affected because it learns deterministic \(z_i\) given \(x\). Using the correlation matrix is not a principled way to evaluate independence in disentanglement learning.

![Figure 7: Correlation matrix of representations learned by FactorVAE, \(\beta\)-VAE and AAE.](image)

**A.5 Trade-off between informativeness, independence and the number of latent variables**

Before starting our discussion, we provide the following fact:

**Fact 2.** Assume we try to fill a fixed-size pool with fixed-size balls given that all the balls must be inside the pool. The only way to increase the number of the balls without making them overlapped is reducing their size.

![Figure 8: Illustration of representations learned by AAE and FactorVAE.](image)

In the context of representation learning, a pool is \(x\) with size \(H(x)\) which depends on the training data. Balls are \(z_i\) with size \(H(z_i)\). Fact. 2 reflects the situation of AAE (see Fig. 8[left]). In AAE, all \(z_i\) are deterministic given \(x\) so the condition “all balls are inside the pool” is met. \(H(z_i) \approx\)
We assume that Rec increases when the number of latent variables in AAE increases, where Rec (Note that we do not count permutation invariance among z). The derivative is non-negative there. This suggests that 65 or 100 latent variables are still not enough to capture all information in the data. FactorVAE, however, handles the increasing number of latent variables in a different way. Thanks to the KL term in the loss function that forces q(z|x) to be stochastic, FactorVAE can break the constraint in Fact and allows the balls to stay outside the pool (see Fig. 8 right). If we increase the number of latent variables but still enforce the independence constraint on them, FactorVAE will keep a fixed number of informative representations and make all other representations “noisy” with zero informativeness scores. We refer to that capability of FactorVAE as code compression.

A.6 Why FactorVAE can learn consistent representations?

Inspired by the variational information bottleneck theory, we rewrite the standard VAE objective in an equivalent form as follows:

\[ \min_{q(z|x)} I(x, z) \quad \text{s.t.} \quad \text{Rec}(x) \leq \beta \]  

where Rec(x) denotes the reconstruction loss over x and \( \beta \) is a scalar.

In the case of FactorVAE, since all latent representations are independent, we can decompose \( I(x, z) \) into \( \sum_{i} I(x, z_i) \). Thus, we argue that FactorVAE optimizes the following information bottleneck objective:

\[ \min_{q(z|x)} \sum_{i} I(x, z_i) \quad \text{s.t.} \quad \text{Rec}(x) \leq \beta \]  

We assume that Rec(x) represents a fixed condition on all \( q_i(z|x) \). Because \( I(x, z_i) \) is a convex function of \( q(z_i|x) \) (see Appdx. A.7), minimizing Eq. 14 leads to unique solutions for all \( q(z_i|x) \) (Note that we do not count permutation invariance among \( z_i \) here).

To make Rec(x) a fixed condition on all \( q_i(z|x) \), we can further optimize \( p(x|z) \) with z sampled from a fixed distribution like \( N(0, I) \). This suggests that we can add a GAN objective to the original FactorVAE objective to achieve more consistent representations.

A.7 \( I(x, z) \) is a convex function of \( p(z|x) \)

Let us first start with the definition of a convex function and some of its known properties.

**Definition 3.** Let \( X \) be a set in the real vector space \( \mathbb{R}^D \) and \( f : X \rightarrow \mathbb{R} \) be a function that output a scalar. \( f \) is convex if \( \forall x_1, x_2 \in X \) and \( \forall \lambda \in [0, 1] \), we have:

\[ f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2) \]

**Proposition 4.** A twice differentiable function \( f \) is convex on an interval if and only its second derivative is non-negative there.
We first trained several FactorVAE models with 3 latent variables on the CelebA dataset. After training, for each model, we performed 2D interpolation on every pair of latent variables $z_i, z_j$ ($i \leq j$) and decoded the interpolated latent representations back to images for visualization. We found that the learned representations from these models share visually similar patterns, which is illustrated in Fig. [10]. It is apparent that all images in Fig. [10] are derived from a single one (e.g. we can choose the first image as a reference) by switching the rows and columns and/or flipping the whole image vertically/horizontally. The reason why switching happens is that all latent variables of FactorVAE are permutation invariant. Flipping happens due to the symmetry of $q(z_i)$ which is forced to be similar to $p(z_i) = \mathcal{N}(0, 1)$.

Proposition 5 (Jensen’s inequality). Let $x_1, \ldots, x_n$ be real numbers and let $a_1, \ldots, a_n$ be positive weights on $x_1, \ldots, x_n$ such that $\sum_{i=1}^{n} a_i = 1$. If $f$ is a convex function on the domain of $x_1, \ldots, x_n$, then

$$f \left( \frac{1}{n} \sum_{i=1}^{n} a_i x_i \right) \leq \sum_{i=1}^{n} a_i f(x_i)$$

Equality holds if and only if all $x_i$ are equal or $f$ is a linear function.

Proposition 6 (Log-sum inequality). Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ be non-negative numbers. Denote $a = \sum_{i=1}^{n} a_i$ and $b = \sum_{i=1}^{n} b_i$. We have:

$$\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \geq a \log \frac{a}{b}$$

with equality if and only if $\frac{a_i}{b_i}$ are equal for all $i$.

Armed with the definition and propositions, we can now prove that $I(x, z)$ is a convex function of $p(z|x)$. Let $p_1(z|x)$ and $p_2(z|x)$ be two distributions and let $p_*(z|x) = \lambda p_1(z|x) + (1 - \lambda) p_2(z|x)$ with $\lambda \in [0, 1]$. $p_*(z|x)$ is a valid distribution since $p_*(z|x) > 0 \ \forall z$ and $\int_x \int_z p_*(z|x)p(x) \ dz \ dx = 1$. In addition, we have:

$$p_*(z) = \int_x p_*(z|x)p(x) \ dx = \int_x (\lambda p_1(z|x) + (1 - \lambda) p_2(z|x)) p(x) \ dx = \lambda \int_x p_1(z|x)p(x) \ dx + (1 - \lambda) \int_x p_2(z|x)p(x) \ dx$$

$$= \lambda p_1(z) + (1 - \lambda) p_2(z)$$

We write $I(x, z) = \lambda I_1(x, z) + (1 - \lambda) I_2(x, z)$ as follows:

$$I(x, z) = \lambda \int_x p(x) \int_z p_1(z|x) \log \frac{p_1(z|x)}{p_1(z)} \ dz \ dx + (1 - \lambda) \int_x p(x) \int_z p_2(z|x) \log \frac{p_2(z|x)}{p_2(z)} \ dz \ dx$$

$$= \int_x p(x) \int_z \left( \lambda p_1(z|x) \log \frac{p_1(z|x)}{\lambda p_1(z)} + (1 - \lambda) p_2(z|x) \log \frac{(1 - \lambda)p_2(z|x)}{(1 - \lambda)p_2(z)} \right) \ dz \ dx$$

$$\geq \int_x p(x) \int_z p_*(z|x) \log \frac{p_*(z|x)}{p_*(z)} \ dz \ dx$$

(15)

where the inequality in Eq. [15] is the log-sum inequality. This completes the proof.

A.8 Experiments to show that FactorVAE learns consistent representations

We first trained several FactorVAE models with 3 latent variables on the CelebA dataset. After training, for each model, we performed 2D interpolation on every pair of latent variables $z_i, z_j$ ($i \leq j$) and decoded the interpolated latent representations back to images for visualization. We found that the learned representations from these models share visually similar patterns, which is illustrated in Fig. [10]. It is apparent that all images in Fig. [10] are derived from a single one (e.g. we can choose the first image as a reference) by switching the rows and columns and/or flipping the whole image vertically/horizontally. The reason why switching happens is that all latent variables of FactorVAE are permutation invariant. Flipping happens due to the symmetry of $q(z_i)$ which is forced to be similar to $p(z_i) = \mathcal{N}(0, 1)$. 

17
Figure 10: Random traversal on the latent space of FactorVAE. We can easily see the visual resemblance among image regions corresponding the same number.

Figure 11: Top 10 representations sorted by the variance of the distribution of $E_{q(z_i|x(n))}[z_i]$ over all $x^{(n)}$.

We then repeated the above experiment on FactorVAE models containing 65, 100, 200 latent variables, but replacing 2D interpolation on pairs of latent variables with conditional 1D interpolation on individual latent variables to account for large numbers of combinations. We sorted the latent variables $z_i$ of each model according to the variance of the distribution of $E_{q(z_i|x^{(n))}}[z_i]$ over all data samples $x^{(n)} \sim p_D(x)$ in descending order. Fig. 11 shows results for the top 10 latent variables of each model. We can see that some factors of variation are consistently learned by these models, for example, those that represent changes in color of the image background. Because these factors usually appear on top, we hypothesize that the learned factors should follow some fixed order. However, many pronounced factors do not appear at the top, suggesting that the sorting criterion is inadequate. We then used the informativeness metric defined in Sec. 4.1 to sort the latent variables. Now the “visual consistency” and “ordering consistency” patterns emerge, (see Fig. 12). We also observed that the number of learned factors is relatively fixed (around 38-43) for all models despite that the number of latent variables varies significantly from 65 to 200.

Figure 12: Top 10 representations sorted by informativeness scores. We can clearly see the consistency of representations across different runs.
Figure 13: Random traversal on the space of two latent components. Columns correspond to different runs. Rows correspond to different pairs of latent components (or latent pairs). The number of latent components is 3 in all runs, hence, there are only 3 possible latent pairs (regardless of the order of latent components in each pair). Note that in each row, we group visually similar latent pairs across different runs together under the title “Between $z_i$ and $z_j$” but in fact, $z_i$ and $z_j$ are different across runs. In the first 2 runs, TC=50 and in the last two runs, TC=10.

A.8.1 More visual results

See Fig. [13] for the progression along two latent factors, Fig. [14] for unsorted informative representations, and Fig. [15] for the representations sorted by their informativeness.

A.9 Analysis of existing metrics for interpretability

In this section, we analyze recent metrics, including Z-diff score [16, 18], Separated Attribute Predictability (SAP) [23] and Mutual Information Gap (MIG) [8].

The main idea behind the Z-diff score [16, 18] is that if a ground truth generative factor $y_k$ ($k \in \{1, 2, ..., K\}$) is well aligned with a particular disentangled representation $z_i$ (although we do not know which $i$), we can use a simple classifier to predict $k$ using information from $z$. Higgins et al. [16] use a linear classifier while Kim et. al. [18] use a majority-vote classifier. The main drawback of this metric is that it assumes knowledge about all ground truth factors that generate the data. Hence, it is only applicable for a toy dataset like dSprites. Another drawback lies in the complex procedure to compute the metric, which requires training a classifier. Since the classifier is sensitive to the chosen optimizer, hyper-parameters and weight initialization, it is hard to ensure a fair comparison.
Figure 14: All informative representations without sorting. Each subfigure corresponds to a particular model.

The SAP score [23] is computed based on the correlation matrix $C$ between the latent variables $z$ and the ground truth factors $y$. If a latent $z_i$ and a factor $y_k$ are both continuous, the (square) correlation $C_{i,k}$ between them is equal to $\frac{\text{Cov}(z_i, y_k)}{\text{Var}(z_i) \text{Var}(y_k)}$ and is in $[0, 1]$. However, if the factor $y_k$ is discrete, computing the correlation between continuous and discrete variables is not straightforward. The authors handled this problem by learning a classifier that predicts $y_k$ given $z_i$ and used the balanced 6 prediction accuracy as a replacement. Then, for each factor $y_k$, they sorted $C_{i,k}$ in the descending order and computed the difference between the top two scores. The mean of the difference scores for all factors was used as the final SAP score. The intuition for this metric is that if a latent $z_i$ is the most representative for a factor $y_k$ (due to the highest correlation score), then other latent variables $z_{\neq i}$ should not be related to $y_k$, and thus, the difference score for $y_k$ should be high. We believe the SAP score is more sensible than Z-diff but it is only suitable when both the ground truth factors and the latent variables are continuous as no classifier is required. Moreover, if we have $K$ discrete ground truth factors and $L$ latent variables, the number of classifiers we need to learn is $L \times K$, which is unmanageable when $L$ is large.

6To achieve balance, the classifier uses the same number of samples for all categories of $y_k$ during training and testing
The MIG score [8] shares the same intuition as the SAP score but is computed based on the mutual information between every pair of $z_i$ and $y_k$ instead of the correlation coefficient. Thus, the MIG score is theoretically more appealing than the SAP score since it can capture nonlinear relationships between latent variables and factors while the SAP score cannot. The MIG score, to some extent, reflects the concept “interpretability” that we discussed in Section 3 in the main text.

A.10 Computing RMIG

For simplicity, here we only discuss how to compute RMIG for unsupervised learning. RMIG requires the value of $\hat{I}(z_i, y_k)$ for every pair of $z_i, y_k$. We quantize $I(z_i, y_k)$ as follows:

$$I(z_i, y_k) = \sum_{s_i \in S} \sum_{u_k \in U_k} Q(s_i, u_k) \left( \log Q(s_i, u_k) - \log Q(s_i) - \log Q(u_k) \right)$$

where $s_i$ and $u_k$ are quantized bins for $z_i$ and $y_k$, respectively; $S$ is shared among different $z_i$. Based on our probabilistic assumption $p(z_i, y_k, x^{(n)}) = q(z_i|x^{(n)})p(x^{(n)}|y_k)p(y_k)$, we compute $Q(s_i, u_k)$ as follows:

$$Q(s_i, u_k) = \frac{1}{N} \sum_{n=1}^{N} Q(s_i|x^{(n)})Q(u_k|x^{(n)})$$

where $Q(s_i|x^{(n)})$ and $Q(u_k|x^{(n)})$ are quantized versions of $q(z_i|x^{(n)})$ and $p(y_k|x^{(n)})$, respectively. Then, we compute $Q(s_i)$ and $Q(u_k)$ from $Q(s_i, u_k)$ as follows:

$$Q(s_i) = \sum_{u_k \in U_k} Q(s_i, u_k) \quad \text{and} \quad Q(u_k) = \sum_{s_i \in S} Q(s_i, u_k)$$

(16)
We argue that the small differences between RMIG and MIG1 scores in some models are caused within the range.

Although erf is denoted as

Assume that we have a Gaussian distribution $N(\mu, \sigma)$. The definite integral of its density function within the range $[a, b]$ denoted as $G(a, b)$ can be computed as follows:

$$G(a, b) = \int_a^b \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) dx$$

$$= \frac{1}{2} \left( \text{erf}\left(\frac{b - \mu}{\sigma \sqrt{2}}\right) - \text{erf}\left(\frac{a - \mu}{\sigma \sqrt{2}}\right) \right)$$

Although erf($\cdot$) does not have analytical form, we can compute its values with high precision using polynomial approximation. For example, the following approximation provides a maximum error of $5 \times 10^{-4}$:

$$\text{erf}(x) \approx 1 - \frac{1}{\left(1 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4\right)^4}, \quad x > 0$$

where $a_1 = 0.278393, a_2 = 0.230389, a_3 = 0.000972, a_4 = 0.078108$. 

---

Figure 16: **Left:** Correlation between our RMIG (#bins=100) and the original MIG [8] (#samples=10000). **Right:** Correlation between our RMIG (#bins=100) and the implementation of MIG in [26] (#bins=100). Experiments are conducted on the dSprites dataset.

An interesting thing here is that $Q(s_i)$ computed in Eq. 16 is equal to $\frac{1}{N} \sum_{n=1}^{N} q(s_i|x^{(n)})$ which means $Q(s_i)$ is truly a quantized version of $q(z_i)$ and is not affected by any ground truth factors.

### A.11 Comparing RMIG with other MIG implementations

RMIG has several advantages compared to the original MIG [8] which we refer as MIG1: i) RMIG works on real datasets, MIG1 does not; ii) RMIG supports continuous factors, MIG1 does not. On toy datasets such as dSprites, RMIG produces almost the same results as MIG1 (Fig. 16 (left)).

Locatello et. al. [26] provided an implementation of MIG which we refer as MIG2. MIG2 is **theoretically not correct** in two points: i) it only uses the mean of the distribution $q(z_i|x^{(n)})$ instead of the whole distribution $q(z_i|x^{(n)})$, and ii) the bin range and width varies for different $z_i$. The performance of MIG2 is, thus, unstable. We can easily see this problem by comparing the right plot with the left plot in Fig. 16. MIG2 usually overestimates the true MIG1 when evaluating $\beta$-VAE models with a large $\beta$ (e.g. $\beta \in \{20, 30, 50\}$). We guess the reason is that in these models, $q(z_i|x^{(n)})$ usually has high variance, hence, using the mean of $q(z_i|x^{(n)})$ like MIG2 leads to the wrong estimation of $I(z_i, y_k)$.

### A.12 Definite integral of a Gaussian density function

Assume that we have a Gaussian distribution $N(\mu, \sigma)$. The definite integral of its density function within the range $[a, b]$ denoted as $G(a, b)$ can be computed as follows:

$$G(a, b) = \int_a^b \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) dx$$

$$= \frac{1}{2} \left( \text{erf}\left(\frac{b - \mu}{\sigma \sqrt{2}}\right) - \text{erf}\left(\frac{a - \mu}{\sigma \sqrt{2}}\right) \right)$$

Although erf($\cdot$) does not have analytical form, we can compute its values with high precision using polynomial approximation. For example, the following approximation provides a maximum error of $5 \times 10^{-4}$:

$$\text{erf}(x) \approx 1 - \frac{1}{\left(1 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4\right)^4}, \quad x > 0$$

where $a_1 = 0.278393, a_2 = 0.230389, a_3 = 0.000972, a_4 = 0.078108$. 

---

https://github.com/google-research/disentanglement_lib
Figure 17: Normalized informativeness scores (bins=100, 100% data) of all latent variables sorted in descending order. For InfoGAN, we only plot the informativeness value of factorial latent variables. The red column corresponds to the categorical latent variables.

|                             | $H(z_{i^*}, y_k)$ | $I(z_{i^*}, y_k)$ | $I(z_{j^*}, y_k)$ | RMIG | JEMMI |
|-----------------------------|-------------------|-------------------|-------------------|------|-------|
| FactorVAE                  | 5.7022            | 0.4359            | 0.4236            | 0.0054 | 0.8239 |
| AAE                        | 6.1790            | 0.0635            | 0.0591            | 0.0019 | 0.8940 |
| InfoGAN                    | 2.5954            | 2.008             | 0.006             | **0.8699** | **0.1289** |

Table 8: Normalized RMIG and JEMMI scores (bins=100, 100% data). Entropy of the digit class is $H(y_k) = \log(10) \approx 2.3012$.

A.13 Experiments on MNIST

In this section, we analyze the performances of FactorVAE, AAE and InfoGAN on MNIST. They share the same architecture and settings described in Appdx. A.2. All the three methods have 65 latent variables. In FactorVAE and AAE, all the latent variables are continuous. In InfoGAN, the factorial part consists of 1 categorical latent variable with 10 classes and 2 continuous latent variables, and the noisy part consists of the remaining 62 continuous latent variables.

A.13.1 Informativeness

We show the informativeness of the representations of the three models in Fig. 17. Again, we can see that all representations of AAE have equally high informativeness while in FactorVAE, only a fraction of representations are informative. In InfoGAN, the categorical representation $z_{i^*}^{\text{cat}}$ is the most informative with the normalized score is very close to 1. It means that the conditional distribution $q(z_{i^*}^{\text{cat}} | x)$ is nearly one-hot and the marginal distribution is $q(z_{i^*}^{\text{cat}})$ is well balanced.

A.13.2 Independence

The results of FactorVAE and AAE under the indepedence metric are similar to the results on CelebA so we do not provide them here.

A.13.3 Interpretability

From Fig. 18 it is clear that InfoGAN is very good at disentangling the digit class since this model achieves very high RMIG score and very low JEMMI score. We believe the main reason is that InfoGAN has a categorical latent variable that reflects the categorical structure of the digit class factor. FactorVAE and AAE only use continuous latent variables so they are unable to disentangle the digit class. The quantitative results are consistent with the visual results shown in Fig. 19 and Fig. 20.

A.13.4 Visual results

See Fig. 18 for top most informative representations and Fig. 19 for top most interpretable representations. Some random interpolation of InforGAN representations is listed in Fig. 20.
Figure 18: Top 10 representations sorted by their informativeness.

Figure 19: Top 10 representations sorted by their mutual information with the digit class.

Figure 20: Random 2D interpolation InfoGAN’s representations.
A.14 Experiments on dSprites

Here, we analyze the performances of FactorVAE, $\beta$-VAE and AAE on the dSprites dataset. All models use the same architecture and settings in Appdx A.2 and have 10 latent variables. This dataset has 5 ground truth factors which are “shape” (3 values), “scale” (6 values), “rotation” (40 values), “x-position” (32 values), “y-position” (32 values).

A.14.1 Informativeness

From Fig. 21, we see that 5 representations of AAE have equally high informativeness scores while the remaining 5 representations have nearly zeros informativeness scores. This is because AAE needs only 5 representations to capture all information in the data. FactorVAE also needs only 5 representations but some are less informative than those of AAE. Note that the number of ground truth factors of variation in dSprites dataset is also 5.

A.14.2 Independence

Fig. 22 shows heat maps of MISJED scores for the three models.

A.14.3 Interpretability

From Tables 9 and 10, we see that FactorVAE is very good at disentangling “scale”, “x-position” and “y-position” but fails to disentangling “shape” and “rotation”. However, FactorVAE still performs much better than $\beta$-VAE and AAE. These results are consistent with the visual results in Fig. 23.

Also note that in FactorVAE, the RMIG scores for “scale” and “x-position” are quite similar but the JEMMI score for “scale” is higher than that for “x-position”. This is because the quantized distribution (with 100 bins) of a particular representation $z_i$ fits better to the distribution of “x-position” (having 32 possible values) than to the distribution of “scale” (having only 6 possible values).
Table 9: Normalized RMIG scores (bins=100, 100% data).

| Shape | Scale | Rotation | Pos X | Pos Y |
|-------|-------|----------|-------|-------|
| FactorVAE | 0.2412 | 0.7139 | 0.0523 | 0.7198 | 0.7256 |
| β-VAE | 0.0481 | 0.1533 | 0.0000 | 0.4127 | 0.4193 |
| AAE | 0.0053 | 0.0786 | 0.0098 | 0.3932 | 0.4509 |

Table 10: Normalized JEMMI scores (bins=100, 100% data).

| Shape | Scale | Rotation | Pos X | Pos Y |
|-------|-------|----------|-------|-------|
| FactorVAE | 0.6841 | 0.3422 | 0.7204 | 0.2908 | 0.2727 |
| β-VAE | 0.8642 | 0.8087 | 0.9199 | 0.5629 | 0.5576 |
| AAE | 0.8426 | 0.8143 | 0.8665 | 0.5738 | 0.5258 |

Figure 23: Top 3 representations sorted by their mutual information with different ground truth factors.