Towards an Improved Metric for Evaluating Disentangled Representations

Anonymous

***Abstract*—Disentangled representation learning plays a pivotal role in making representations controllable, interpretable and transferable. Despite its significance in the domain, the quest for reliable and consistent quantitative disentanglement metric remains a major challenge. This stems from the utilisation of diverse metrics measuring different properties and the potential bias introduced by their design. Our work undertakes a comprehensive examination of existing popular disentanglement evaluation metrics, comparing them in terms of measuring aspects of disentanglement (viz. Modularity, Compactness, and Explicitness), detecting the factor-code relationship, and describing the degree of disentanglement. We propose a new framework for quantifying disentanglement, introducing a metric entitled *EDI*, that leverages the intuitive concept of *exclusivity* and improved factor-code relationship to minimize ad-hoc decisions. An in-depth analysis reveals that EDI measures essential properties while offering more stability than existing metrics, advocating for its adoption as a standardised approach.**

***Index Terms*—disentanglement, representation learning**

I. Introduction

The learning of effective representations is crucial for enhancing the performance of downstream tasks in various domains. As defined by Bengio *et al*. [4], representation learning transforms observations into a format that captures the essence of data's inherent patterns and structures. An ideal representation should exhibit five key characteristics: (a) *Disentanglement*, ensuring separate encoding of interpretable factors; (b) *Informativeness*, capturing the diversity of data; (c) *Invariance*, maintaining stability across changes in unrelated dimensions; (d) *Compactness*, summarising essential information efficiently; and (e) *Transferability*, facilitating application across different contexts. These attributes collectively enhance the model's interpretability, efficiency, and adaptability across tasks and domains.

While the literature does not present a unified theory of disentanglement, the consensus leans towards the principle that generative factors of variation ought to be individually encapsulated within distinct latent codes in the representation space. For instance, in an image dataset of human faces, an effective disentangled representation would feature separate dimensions for each identifiable attribute, such as face size, hairstyle, eye colour, and facial expression, among others.

The concept of *modularity* or factor independence stemming from Independent factor analysis [2] supports a commonly accepted view on disentanglement [4, 7, 15]. This notion assumes no causal dependencies among the encoded dimensions, suggesting that in an ideally modularised representation, each generative factor is represented by a unique code or an independent subset of codes. As a result, modifying a specific code or subset within the representation space should ideally influence only its corresponding generative factor, leaving others unchanged.

An alternative perspective on disentanglement, rooted in the concept of compactness, posits that a generative factor should be represented by no more than a single code. This conceptualisation of disentanglement, emphasising the compactness and singularity of representation for each generative factor, has been adopted as a defining criterion by studies such as [12, 6], and is also referred to as *completeness* [7]. Regardless of debates surrounding the desirability of compactness [15, 5], these concepts, along with modularity, have been embraced as part of a more comprehensive yet stringent framework for understanding disentanglement [7, 15]. This integrated approach, which considers modularity, compactness, and explicitness, also known correspondingly as disentanglement, completeness, informativeness, has gained traction in more recent scholarly reviews on the topic [16, 5]. Accordingly, a metric designed to quantify modularity and compactness should also assess informativeness i.e. , the extent to which latent codes encapsulate information about generative factors. When the ground truth factors of variation are identifiable, this informativeness transforms into explicitness, denoting the comprehensive representation of all recognised factors [9].

Despite significant advancements in disentangling latent spaces via deep latent variable models [8, 10, 6], the literature still lacks a reliable and unified metric for evaluation. Traditionally, evaluation has been qualitative, relying on visual interpolation. The quantitative metrics that are available vary across the literature, and it has been demonstrated that the outcomes of these metrics do not consistently align with the findings from qualitative studies of disentangled representations [1, 16, 14]. Due to the variability in outcomes, a common measurement criterion has yet to be established. Furthermore, we observed that most existing metrics fail in certain scenarios and cannot be considered reliable across all settings, even when there is general agreement among them. Through an extensive analysis of the metrics, we identify these shortcomings and propose a new metric that is theoretically sound, reflects the desired properties better and is experimentally more robust.

Concretely, in this work, our contributions can be summarised as:

- We analyse the popular quantitative disentanglement metrics, identify their theoretical underpinnings, elaborate on the differences, and demonstrate their performance under

various simulated conditions.

- Based on the identified shortcomings, we propose a new metric called EDI, built on the novel principle of exclusivity. We show this metric performs better compared to the existing metrics on tests measuring calibration, non-linearity and robustness under noise, while being computationally efficient.
- We present a high-quality open-source codebase for reproducing our results and further research in this direction: https://anonymous.4open.science/r/InnVariant-03A5/README.md.

A. Problem Statement

In subsequent sections, we refer to latent dimensions as 'codes', and to the data generative factors as 'factors'. Generative factors are those attributes that describe the perceptual differences between any two samples from dataset $\mathbf{X}$.

Consider a dataset $\mathbf{X} = \mathbf{x}^{(i)}{}_{i=1}^{N}$ comprising N i.i.d. samples. We assume these samples $\mathbf{x}$ are generated by a random process $g : \mathbb{R}^k \to \mathcal{X}$, which takes the ground truth generative factors $\mathbf{z} \in \mathbb{R}^k$ as input and returns the generated data $\mathbf{x} \in \mathcal{X}$. We now consider a latent variable model capable of inferring the corresponding latent representation $\mathbf{c} \in \mathbb{R}^d$ of the data $\mathbf{x}$. This latent representation $\mathbf{c}$, analogous to $\mathbf{z}$, can be used to generate the corresponding data $\mathbf{x}$. The model simulates the random process of generating data $\mathbf{x}$ as follows: latent variables $\mathbf{c}$ are sampled from some prior distribution $p_\theta(\mathbf{c})$, and then the data $\mathbf{x}$ is sampled from a conditional distribution $p_\theta(\mathbf{x}|\mathbf{c})$. The model aims to approximate the desired data distribution $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{c})p_\theta(\mathbf{c})d\mathbf{c}$.

Given the latent representations $\mathbf{c}$ learned by the trained latent variable model and the known ground truth generating factors $\mathbf{z}$, we aim to obtain a method to quantitatively evaluate the disentanglement of the latent space $\mathbb{R}^d$ by giving a certain score $s \in \mathbb{R}$ according to the identified definitions of disentanglement.

II. Existing Metrics and their Shortcomings

In a recent survey, Carbonneau *et al*. [5] taxonomise the existing metrics into three categories viz. intervention-based, predictor-based and information-based. While this is a significant scholarly work, there appears to be a functional overlap between the intervention and predictor-based, as they both use either accuracy or weights from predictors to determine the factor-code relationships.

We take a more nuanced view of the metrics to highlight in depth the key differences in design, interpretation of disentanglement and thus investigate the metrics from a three-fold perspective, namely a) *Aspect of measurement*, b) *Detection of factor-code relationship* and c) *Extent of characterisation*. We identify the good practices employed and the limitations of many of them (cf. Table I). Recognising these weaknesses, we propose a new metric that categorically improves upon each (cf. Section III). Detailed mathematical formulations of the existing metrics consistent with this work are described in the appendix.

1) Aspect of measurement.: A close inspection of the metrics reveals a clear dichotomy in perspectives on disentanglement and consequently in the aspect of its measurement. Metrics that developed in studies with modularity as the key characteristic for disentanglement are designed to test if the factor is encoded by one or more codes, and tend to be calculated from the perspective of each code, On the other hand, metrics with compactness as the identified definition of disentanglement are designed to ensure that a code encodes only one factor at a time. These metrics tend to be calculated from the perspective of the factor.

The *Modularity-centric* metrics include the BetaVAE metric, otherwise known as *Z-diff* [8], and its successor, the FactorVAE metric or *Z-min Variance* [10]. These early metrics are intervention-based i.e. they use a predictor to determine which factor was fixed using statistics learnt from the latent codes.

The *Compactness-centric* metrics include the *Separated Attribute Predictability* metric (*SAP*) [12], and *Mutual Information Gap* (*MIG*) [6], followed by *MIG-sup* [13], and *DCIMIG* [16] that were proposed to augment MIG with the ability to also capture modularity.

Other works propose to use a distinct metric to capture each aspect [15], including explicitness, separately. Eastwood *et al*. [7] continue in this vein and propose using three new metrics to compute modularity, compactness and explicitness, calling them disentanglement (D), completeness (C), and informativeness(I) under a unified framework entitled DCI.

2) Detection of relationship.: The mechanism of detection of factor-code relationships varies across the metrics.

Prediction accuracy of classifiers: The Z-diff and Z-min Variance metrics follow the intuition that code dimensions associated with a fixed factor should have the same value. So they fix one generative factor, while varying all the others, and use a linear classifier to predict the index of the fixed factor, based on the variance in each of the latent codes as in Z-diff or the index of the code with the lowest variance as input in Z-min Variance, such that the resulting classifier is a majority vote classifier. While this approach has the advantage of not making assumptions about factor-code relationships, these metrics require careful discretisation of the factor space (eg., the size and number of data subsets), and other design choices like classifier hyper-parameters and distance function. However, for random classifications, there is no code with the lowest variance, each code would get the same number of votes and so Z-min Variance would assign $\frac{1}{d}$ (d being the number of latent codes) instead of 0. The Explicitness metric in [15] is measured similarly to Z-min Variance, with the difference that it uses the mean of one-vs-rest classification and ROC-AUC instead of accuracy. For discrete factors, SAP uses the classification accuracy of predicting factors using a classifier like Random Forest.

Linear correlation coefficient: For continuous factors, SAP computes for each generative factor, the linear R^2 coefficient with each of the codes, then takes the difference between the largest and the second largest coefficient values to predict

the code encoding it. This ensures that a large value is assigned when only one code is highly informative, and others negligible– an intuition exploited by subsequent works like MIG, which employs mutual information instead of R^2. In the case of SAP, however, this limits the detectable factor-code relationship to a linear one.

Ad-hoc model: DCI utilises feature importance derived from classifiers. The authors [7] originally proposed using a LASSO-based classifier with DCI to predict each generative factor from each latent factor and estimate scores from the weights and accuracy of the trained classifier. Hence the relationship matrix relies heavily on the ad-hoc model, requiring careful selection of the model and hyperparameters [5]. Naturally, this metric thus may be prone to stochastic behaviour, which is less than ideal.

Mutual information (MI): The use of mutual information to describe relationships was first proposed in MIG, and has since been adopted by many subsequent metrics, including Modularity score [15], MIG-sup, and DCIMIG. While this choice offers the advantage of not varying by implementation, and making no assumptions about the relationship between factors and codes, all these methods compute mutual information by binning and suffer from several challenges. We elaborate on this further in the next subsection on shortcomings.

3) Extent of characterisation.: The ability of metrics to express the degree of modularity or compactness depends on the extent of characterisation. The Z-diff metric uses maximum value to describe the extent of disentanglement. Consequently, it would not be capable of distinguishing whether a code captures primarily one factor or multiple factors. SAP and MIG take the difference between the top two entries to express the degree of completeness, which would not allow distinguishing whether a factor is encoded by two codes or by more than two codes. This yields limitations in functionality, discussed in the next subsection. MIG-sup, furthermore, is not affected by low information content, as it normalises mutual information by dividing by the entropy of the code, making it ignorant to information loss. DCI, in contrast, is designed well in this regard as it can express the degree of relationship by calculating $1 - entropy$ (where entropy is estimated from the probabilities derived from feature importance). Modularity is also equipped to express the degree well by calculating the deviation of all items from the maximum value.

4) Shortcomings.: Abdi *et al*. [1], in a first attempt, reported inadequacies in the disentanglement metrics, noting discrepancies without delving into the underlying reasons. This observation spurred further investigations within the research community. Chen *et al*. [6] examined metrics through the lens of robustness to hyperparameter selection during experiments and showed that the early modularity-centric metrics overestimated disentanglement. Sepliarskaia *et al*. [16], in subsequent work, provided an initial theoretical analysis, unveiling specific cases of failures in the metrics, but lacked a controlled study. Carbonneau *et al*. [5] showed some controlled evidence of measurement of different properties and reported that the metrics differ in terms of measured properties and overall agreement. Surveying the literature, we identified the following major functional issues, that support our argument to have improved metrics:

a) *Several metrics designed for a particular aspect fail in efficiently reflecting that aspect in all cases.* This is observed strongly in Z-diff and Z-min Variance which penalise modularity violations weakly [16]. We conducted a systematic analysis to test metric calibration to confirm this and identify other discrepancies (cf. Section IV-A). This was also observed in the case of compactness-centric metrics like SAP and DCIMIG[1]. In MIG, it was observed that it assigns a 0, when a factor is encoded by just two codes [5], indicating too strong a penalisation in partial entanglement.

b) *Modularity-centric metrics are generally not equipped to capture compactness and disregard explicitness.* Since these metrics align z_i, with a corresponding set of codes, c_i, this strategy does not ensure that distinct codes are dedicated to unique factors. Further, they do not capture the extent of the factor-code relationship, and consequently cannot be reliably used to reflect disentanglement.

c) *Predictor-based methods can overfit and can be computationally expensive.* Metrics that use predictors to determine factor-code relationships can overfit when there are too few samples, resulting in overestimating explicitness [5]. Furthermore, the complexity of the chosen model can result in undesirable computational complexity (cf. Section IV-E).

d) *Existing information-based metrics are fraught with computational challenges.* The existing metrics that use mutual information using maximum likelihood estimators, that require quantisation of both spaces and parameterised sampling procedures. The existing formulations[2] expect a discretisation of spaces into bins, with the mutual information value estimation being sensitive to binning considerations. These pose further a challenge in scenarios dealing with high-dimensional or non-linear data [11, 5], discussed further in Section IV-B.

III. Exclusivity Disentanglement Index (EDI)

Having identified the best practices in design and their shortcomings, we exploit them to define the the disenglement aspects in a more intuitive and simple way, using the principle of exclusivity. In this section, we introduce our proposed metric EDI. First, we define impact intensity that measures the factor-code relationship. Next, we define exclusivity which we subsequently use as the criteria to define and mathematically construct both modularity and compactness metric formulations.

A. Impact Intensity

We measure the influence each of the factors z_i have on the latent codes c_i using a relationship matrix we call *Impact Intensity*. We introduce two improvements in the computation

[1]When a factor is encoded by two codes, DCIMIG yields a score of $(\max(I(c_0; z_0), I(c_1; z_0)) + I(c_2; z_1))/(H(z_0) + H(z_1)) = 1$.

[2]it is commonly estimated as, $I(c, z) = \sum_{i=1}^{z} \sum_{j=1}^{c} P(i, j) \log\left(\frac{P(i,j)}{P(i)P(j)}\right)$.

Table I: Summary of metrics in regards to measurement aspects viz. modularity (Mod), compactness (Comp) and explicitness (Expl), detection of relationship and extent of characterisation. Identified strengths and weaknesses in design are marked with +/- accordingly.

Metric	Mod	Comp	Expl	Relationship Detection	Extent Characterisation
Z-min Variance [10]	✓			- Majority vote classifier accuracy	- Maximum value
SAP [12]		✓	✓	- Linear correlation (continuous); Predictive accuracy (categorical)	- Difference between top two
MIG [6]		✓	✓	+ Mutual information	- Difference between top two
Modularity [15]	✓		✓	+ Mutual information	+ 1 - avg. squared deviations
DCI [7]	✓	✓	✓	- Feature importance	+ 1 - entropy
MIG-sup [13]	✓			+ Mutual information	- Difference between top two
DCIMIG [16]	✓		✓	+ Mutual information	- Difference between top two

of relationship matrix, namely, a) an improved estimator and b) no reliance on ad-hoc decision model.

As pointed out earlier, existing implementations of MI in metrics are unsuited to high-dimensional continuous variables and fraught with computational challenges [11]. Naturally, a non-parametric estimator with no dependence on discretisation is more suitable. A recently proposed method called MINE [3] operates by training a small neural network to maximise a lower bound on the mutual information between two variables. As it involves no density estimation using maximum likelihood it is flexible and has been shown to converge to the true mutual information between high-dimensional variables [3]. Linearly scalable in both dimensionality and sample size, it offers a significant advantage (cf. Sections IV-B and IV-E).

Thus, we propose computing the relationship matrix as follows: First, we calculate the following required variables: a) $I(\boldsymbol{c}_i; \boldsymbol{z}_j)$, signifying the mutual information computed between each factor $\boldsymbol{c}_i$ and each code $\boldsymbol{z}_j$; b) $I(\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_d; \boldsymbol{z}_j)$, signifying the mutual information between all codes $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_d$ and each factor $\boldsymbol{z}_j$; and c) $H(\boldsymbol{z}_j)$, representing the entropy of each factor. We establish the relationship as $R(\boldsymbol{c}_i; \boldsymbol{z}_j) = \frac{I(\boldsymbol{c}_i; \boldsymbol{z}_j)}{I(\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_d; \boldsymbol{z}_j)}$, denoting the impact intensity of factor $\boldsymbol{z}_j$ on the code $\boldsymbol{c}_i$ among all codes. This, we argue, offers a more accurate representation of the relationship, as latent codes are learned from generative factors.

B. Exclusivity

The concept of exclusivity is crucial in both modularity and compactness. In modularity, we desire a code to capture a singular factor and exclude others. In compactness, it is expected that a factor is represented by a code without overlapping with others. This principle is fundamentally the inverse of impurity.

We propose an intuitive method to quantify the extent of exclusivity, which is defined as the difference between correctness (the maximum value) and incorrectness (the root mean square error of all other values). The objective is to maximise the difference between correctness and incorrectness.

Given a set of attributes $\{a_1, a_2, \ldots, a_n\}$, the exclusivity is mathematically represented as:

$$\text{Exclusivity}(a_1, a_2, \ldots, a_n) = a_{i^*} - \sqrt{\frac{1}{n-1} \sum_{\substack{i=1 \\ i \neq i^*}}^{n} a_i^2},$$
$$\text{where} \quad i^* = \text{argmax}_i \, a_i.$$

The aim is for the maximum value to be as high as possible, with the remainder as minimal as possible.

C. Formulation

By applying the aforementioned concepts of exclusivity to better depict the "extent", and impact intensity to capture factor-code relationships, we formulate the following metrics to measure modularity, compactness, and explicitness.

Modularity. We formalize the metric for modularity, or disentanglement, of a latent code $\boldsymbol{c}_i$ as:

$$D(\boldsymbol{c}_i) = \text{Exclusivity}\left(\boldsymbol{R}(\boldsymbol{c}_i; \boldsymbol{z}_1), \boldsymbol{R}(\boldsymbol{c}_i; \boldsymbol{z}_2), \ldots, \boldsymbol{R}(\boldsymbol{c}_i; \boldsymbol{z}_k)\right).$$

The aggregate modularity score is then calculated as $D = \frac{1}{k} \sum_{i=1}^{d} D(\boldsymbol{c}_i)$, where d denotes the code dimensionality, and k, representing the number of factors, signifies the maximum potential influence a single factor can exert. Notably, this framework may encounter complications due to correlated effects, wherein multiple codes capture a single factor. To address this challenge, we allocate to each code $\boldsymbol{c}_i$ and its predominantly associated factor $\boldsymbol{z}_{j^*}$ a score $S_{ij^*} = D(\boldsymbol{c}_i)$, while assigning $S_{ij} = 0$ for $j \neq j^*$. Accumulating these scores across all factors yields $S_j = \sum_i S_{ij}$ for each factor j. Ensuring that the score for each factor does not exceed 1 (the maximum conceivable impact intensity for each factor is 1), the final score is thus recalculated as $D = \frac{\sum_j \max(S_j, 1)}{k}$, facilitating an accurate assessment of modularity.

We then assign a score $S_{ij^*} = D(\boldsymbol{c}_i)$ to each code $\boldsymbol{c}_i$ and its most effective factor $\boldsymbol{z}_{j^*}$, and mark the others as $S_{ij} = 0$ for $j \neq j^*$. The overall disentanglement is finally calculated as:

$$D = \frac{\sum_j \min(S_j, 1)}{k}, \text{ where } S_j = \sum_i S_{ij}.$$

Compactness. The compactness of a generative factor $\boldsymbol{z}_j$ is calculated as:

$$C(\boldsymbol{z}_j) = \text{Exclusivity}\left(\boldsymbol{R}(\boldsymbol{c}_1; \boldsymbol{z}_j), \boldsymbol{R}(\boldsymbol{c}_2; \boldsymbol{z}_j), \ldots, \boldsymbol{R}(\boldsymbol{c}_d; \boldsymbol{z}_j)\right).$$

Accordingly, the overall compactness score, C, is determined by the average compactness across all factors:

$$C = \frac{1}{k}\sum_{j=1}^{k} C(\boldsymbol{z}_j).$$

Explicitness. For a generative factor $\boldsymbol{z}_j$, explicitness or informativeness is calculated as the ratio of the combined information content of the codes relative to $\boldsymbol{z}_j$ to the entropy of $\boldsymbol{z}_j$ itself:

$$I(\boldsymbol{z}_j) = \frac{I(\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_d; \boldsymbol{z}_j)}{H(\boldsymbol{z}_j)}.$$

Hence, the aggregate measure of informativeness is the mean informativeness across all generative factors:

$$I = \frac{1}{k}\sum_{j=1}^{k} I(\boldsymbol{z}_j).$$

IV. Experiments

In the following sections, we model the relation $c = \gamma(g(\boldsymbol{z}))$ as $\boldsymbol{c} = f(\boldsymbol{z})$. Here, $f(\boldsymbol{z})$ represents a fully-parameterised function controlling the factor-code relationship. For the experiments, factors $\boldsymbol{z}$ are sampled i.i.d from a discrete uniform distribution in Sections IV-A and IV-E, and from a continuous uniform distribution $\mathcal{U} \in \{0, 1\}$ in Section IV-B to Section IV-D. Following [5], we generate N factors to form a set Z and compute the corresponding set of codes $\boldsymbol{c}$ using the experiment-specific $f(\boldsymbol{z})$ parameterised by α, resulting in one representation. Unless otherwise specified, the factor and code dimensionality are kept equal ($k == d$). For each α within the chosen discrete range, we generate M representations and aggregate over these for s random seeds. In Section IV-F, representations are learnt using real latent variable models on a real-world dataset.

A. Are the metrics well calibrated?

Motivated by discrepancies in our exploratory analysis, we first systematically assess metric behaviour via discrete boundary test cases for each of the aspects i.e. modularity, compactness, and explicitness. Codes are arranged in that order with 1 denoting a perfect aspect and 0 completely imperfect. For example, $\#101$ indicates perfect disentanglement and explicitness, but imperfect compactness.

To form the factor space, we sample $N = 50,000$ points from a discrete uniform distribution with a one-to-one encoding. Each category is assigned a distinct code ($k = d = 2$) unless: a) when modularity is low, we encode two factors into one code; b) when compactness is low, we encode a factor into two codes; or c) when informativeness is low, we randomly drop categories within the factors. We simulate a total of $2^3 = 8$ representative cases[3]. Results, reported in Table II using $s = 50$ random seeds, confirm some intuitions, and previously reported observations while revealing interesting insights.

[3]detailed description in supplementary material

As discussed in Section II-4, not all metrics designed for specific aspects are well-calibrated. Z-min Variance, for instance, which is modularity-centric, fails to penalise modularity violations, with scores larger than 0.5 in low modularity scenarios ($\#000, \#001, \#010, \#011$). This stems from its assigning of the minimum score as $\frac{1}{d}$. The Modularity metric, while performing perfectly in high modularity cases, unexpectedly assigns high scores of 0.75 in low modularity scenarios too ($\#010$ and $\#011$). This is likely due to an error introduced by dividing the maximum term in the formula. DCI Mod correctly assigns low scores in low modularity cases of $\#000$ and $\#001$, however, it assigns relatively large scores of $> 50\%$ in $\#010$ and $\#011$, indicating some influence of high compactness, which is not ideal. In contrast, EDI Mod assigns 0.43, reflecting low modularity relatively better.

The discrepancies appear in compactness-centric metrics as well. SAP, for instance, assigns a relatively low score of 0.33 in both high compactness scenarios ($\#010$ and $\#110$) but a higher score of 0.45 in the low compactness case of $\#101$, suggesting greater influence from other aspects. MIG also assigns relatively low scores of 0.41 and 0.45 in high compactness scenarios ($\#010$ and $\#110$), but a higher score of 0.49 in the less compact scenario of $\#101$. Its successor, MIG-sup, assigns a large score of 0.99 in both low ($\#100$, $\#101$) and high compactness scenarios ($\#110$, $\#111$) while tends to assign intermediate scores of about 0.5 in high compactness, low modularity scenarios ($\#010$). This shows a high influence from modularity but yields no clear interpretation of the captured aspects. Furthermore, the DCIMIG metric assigns a higher score to the low compactness case of $\#101$ confirming weak penalisation, as a consequence of two codes capturing different information extent about the factor. DCI Comp assigns very high scores to scenarios $\#100$ and $\#101$, which are highly modular despite low compactness. EDI Comp, in contrast, assigns lower scores. In terms of explicitness, EDI and DCI perform comparably. Overall the results indicate EDI to be better calibrated in comparison to the existing metrics.

B. How do the metrics deal with non-linearity?

The ability to attribute accurate scores when factor-code relationships are non-linear as in realistic data is a crucial property. A robust metric should exhibit negligible effect with increasing non-linearity. In this experiment, we simulate representations which are perfectly compact and modular, but the encoding function becomes increasingly non-linear. We use $f(z) = 1000 - \alpha + 0.25\tan\big(\omega(z - 0.5)\big) + 0.5$ where $\omega = 2\arctan\big(1000\alpha - \frac{0.25}{2}\big)$. As α increases, the curve becomes more steep but remains monotonic for $z \in [0, 1]$. Using $k = d = 6$, we simulate $M = 50$ representations with $N = 20,000$ points sampled from $\mathcal{U} \in [0, 1]$. For $s = 50$ seeds, we report the aggregated scores in Figure 1.

This experiment perfectly challenges the complexity of the predictors employed by the metrics, and highlights the potential issues inherent to mutual information computation using density estimation, yielding interesting insights. Metrics that calculate MI using binning methods generally perform

Table II: Measurement results of all boundary test cases. Representative codes follow a (m,c,i) format with binary values indicating high or low. Results are reported as mean scores for 50 random seeds. Standard deviations are not included due to limited space, however, most values are close to 0.

Nr.	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
Z-min Variance	0.57	0.55	0.62	0.67	1.00	1.00	1.00	1.00
SAP	0.04	0.03	0.33	0.88	0.22	0.45	0.33	0.88
MIG	0.06	0.034	0.41	0.82	0.23	0.49	0.45	0.99
MIG-sup	0.11	0.03	0.54	0.63	0.99	0.99	0.99	1.00
DCI Mod	0.08	0.00	0.57	0.57	0.99	1.00	0.99	1.00
DCI Comp	0.08	0.00	0.99	1.00	0.75	0.68	0.99	1.00
DCI Expl	0.44	1.00	0.44	1.00	0.44	1.00	0.44	1.00
Modularity	0.25	0.25	0.75	0.75	1.00	1.00	1.00	1.00
DCIMIG	0.05	0.02	0.17	0.46	0.38	0.75	0.46	1.00
EDI Mod	0.11	0.02	0.43	0.43	0.99	0.99	0.99	0.99
EDI Comp	0.12	0.02	0.99	1.00	0.61	0.57	1.00	1.00
EDI Expl	0.45	0.99	0.45	0.99	0.45	0.99	0.45	0.99

inadequately. To illustrate this, we contrasted MIG to its alternative variant implemented with a non-parametric estimator, KSG [11]. MIG-ksg demonstrates greater stability until reaching $\alpha = 0.6$, after which it gradually declines. Metrics using linear models like SAP, exhibit instability as non-linearity increases. This is also observed in DCI, which in its original implementation uses a LASSO classifier. Both DCI Mod and Comp decline and become more variable as non-linearity increases. In contrast, Z-diff and Modularity scores exhibit stability throughout the experiment. EDI Mod and Comp consistently assign a perfect score of 1 throughout too, indicating robustness in this setting. For explicitness, a slight reduction in mutual information is expected.

C. How do the metrics behave on decreasing disentanglement?

Next, we evaluate the performance of the metrics as a perfectly disentangled representation gradually transitions to an entangled state. We conduct an experiment where we linearly reduce the modularity and compactness of the representation while maintaining explicitness. To describe the factor-code relationship, we employ $f(\boldsymbol{z}) = \boldsymbol{z}\boldsymbol{R}$, with

$$\boldsymbol{R} = \begin{pmatrix} 1-\alpha & \alpha & 0 & \cdots & 0 \\ 0 & 1-\alpha & \alpha & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1-\alpha & \alpha \\ \alpha & 0 & \cdots & 0 & 1-\alpha \end{pmatrix}.$$

Like in the previous experiment, we use $k = d = 6$, and simulate $M = 50$ representations with $N = 20,000$ points. For $s = 50$ seeds, we report the aggregated scores in Figure 2. As the parameter α increases, we expect a linear decrease in all metrics dealing with modularity or compactness, though not reaching 0 entirely[4]. Z-diff and Z-min Variance metrics fail completely in this regard. Conversely, both modularity and compactness components of EDI and DCI respectively demonstrate robust performance. DCI Expl, which does not represent true mutual information remains largely unaffected. It is also prone to overfitting and hence may overestimate explicitness [5]. In comparison, EDI Expl exhibits a drop when one factor becomes equally represented by two codes. Most information-based metrics also perform well, however, assign zero value already when only one factor or code becomes fully entangled.

D. How do the metrics deal with noise?

In this segment, we investigate how the metrics behave when we keep modularity and compactness intact, but gradually reduce explicitness. Choosing $f(z) = (1-\alpha)z + \alpha n$, and keeping the setting consistent as before, we report the results in Figure 3. In this simulation, we expect the metrics representing explicitness to decrease gradually. In this regard, both EDI Expl and DCI Expl perform adequately, however unlike DCI, EDI Expl does not assign a perfect score of 1 here due to the true mutual information being less than 1. Metrics representing modularity and compactness should exhibit unchanged behaviour under noise. Here we see a larger contrast between the metrics. While MIG, SAP, and Modularity metric decrease gradually and reach 0, Z-min Variance collapses rapidly after the middle mark. DCI Mod and Comp also decrease, first slowly, then quite rapidly as α approaches 0.8. Here we can see strikingly more stability in EDI Mod and Comp. In fact, even Z-diff appears to be robust here.

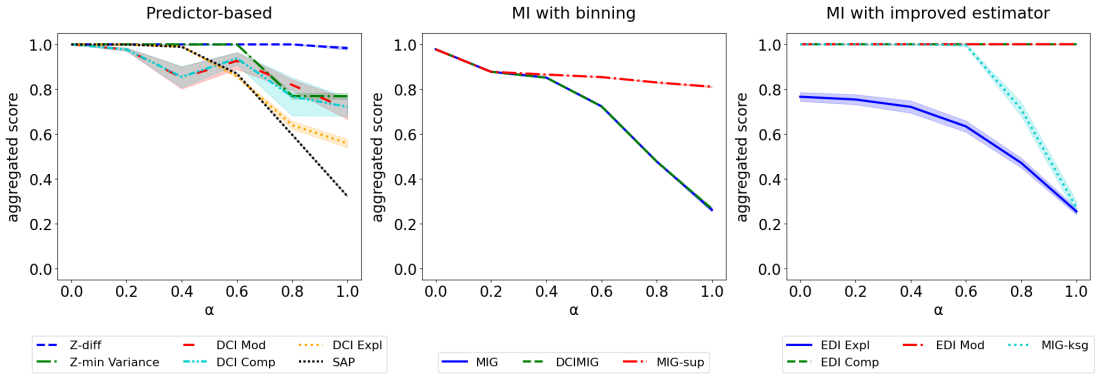

Figure 1: As α increases, the factor-code relationship becomes more non-linear. We see a decline in most metrics computing MI using binning, as well as metrics that use predictors. EDI, in comparison, exhibits good stability.

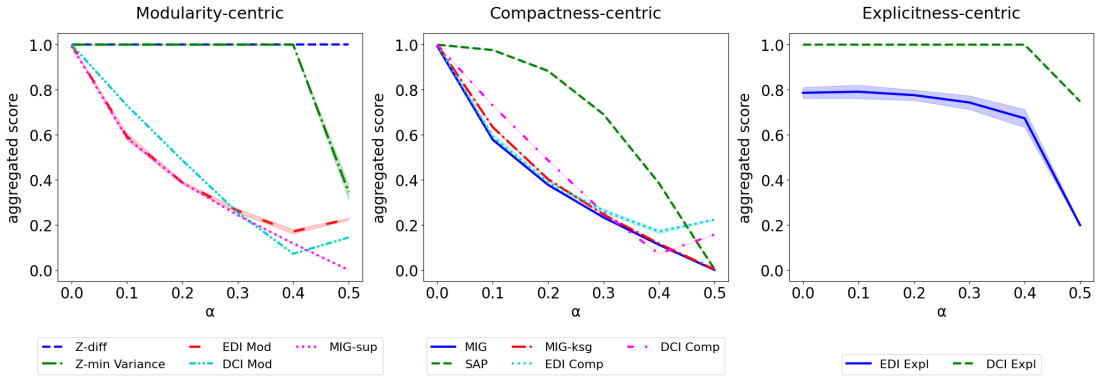

Figure 2: As α increases, the representation becomes less modular and compact. EDI and DCI perform adequately, whereas MIG, SAP assign 0 with partial entanglement and Z-diff, Z-min Variance fail to observe any difference.

E. How do the metrics compare on resource efficiency?

Here, we evaluate and compare the metrics in terms of sample efficiency and time complexity. To test sample efficiency, we compute the difference in estimated scores when using subsets of data $N \in \{100, 1000, 10000, 10000\}$ against the full sample size of $N = 100,000$. We keep the experimental setup as in section IV-A, with a difference that we use only 10 random seeds, and report the mean differences in Figure 4 (left). We observe the minimum samples required to reliably estimate scores vary across metrics, as a result of design choices. While most metrics converge around the 10,000 sample mark, it becomes evident that classifiers-based metrics such as DCI necessitate larger sample sizes for optimal performance, whereas metrics reliant on MI require fewer samples. In this regard, EDI generally is more sample-efficient than DCI, with the exception of its explicitness component, which needs more samples to reliably compute mutual information.

In terms of time complexity, most metrics are constant or (sub)linear. We observed EDI to be linear, and for DCI, it depends on the complexity of the ad-hoc model. If one were to choose complex models like random forest or XGBoost (DCI-xgb) to model non-linearity better as recommended [5], this would come with a serious disadvantage of the curse of dimensionality (cf. Figure 4 (right)).

F. Metric Agreement on Real Dataset

It was observed in previous works that metrics do not correlate on complex datasets, and the correlations may not be consistent across datasets [14]. While we do not test consistency in this regard, we test general agreement of the metrics on a popular dataset used in the domain, namely Shapes3D[5], in order to test if EDI can be applied in real settings. We heuristically opted to utilise FactorVAE [10] and BetaVAE [8] for learning representations. For FactorVAE, we chose $\gamma \in \{2, 4, 6, 8, 10\}$, and for BetaVAE, $\beta \in \{2, 3, 4, 5\}$. For 5 random seeds, this resulted in 45 representations in total. Next, we produce a ranking of the learned representations on the scores and calculate the agreement between the rankings for each pair of metrics using Spearman's coefficient (cf. Section IV-F).

We observe EDI to display strong correlations with SAP, DCIMIG, MIG-sup, Z-min Variance and perfect correlation with Modularity, indicating general agreement on both modularity and compactness aspects. The exception in this case

[4]since increasing α does not lead to perfect entanglement, i.e. a factor equally represented by all codes. Instead, only two codes capture a factor.

[5]https://github.com/google-deepmind/3d-shapes

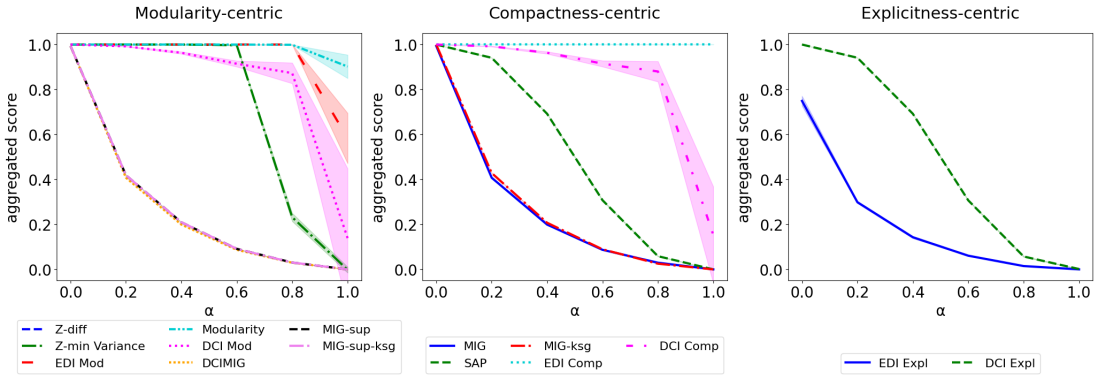

Figure 3: As α increases the representation becomes more noisy. We expect explicitness measuring metrics to gradually reach 0, but modularity and compactness metrics should stay unaffected. EDI exhibits greater stability here in comparison to others.

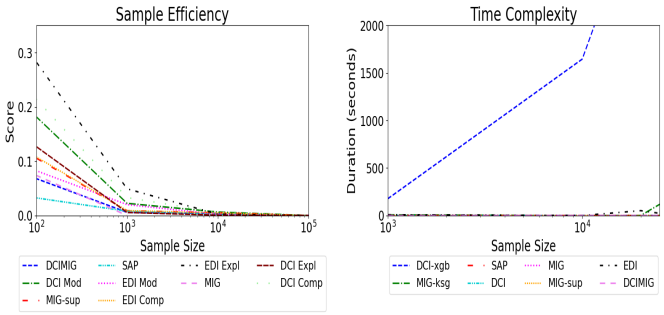

Figure 4: Comparing sample efficiency (left) and time complexity (right). A metric is more sample efficient if it shows a smaller difference in its score as sample size increases. Time complexity is assessed by examining the rate of change in computation duration as the sample size increases. Here we see a clear downside of using complex predictors to model factor-code relationships.

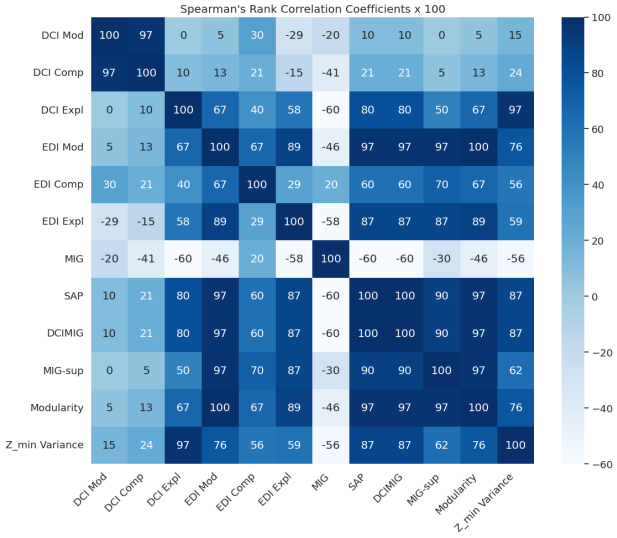

Figure 5: Metric correlations on Shapes3D using Spearman's rho.

is DCI which does not appear to correlate with most metrics. It might be that DCI required more hyperparameter tuning. MIG demonstrates a negative correlation with all metrics except EDI compactness. indicating that both measure similar properties to an extent.

V. Conclusion

In this study, we conducted a comprehensive analysis of existing metrics for evaluating disentanglement, elucidating differences in their assumptions, design, and functionality. By focusing on best practices, we formulated a novel metric, EDI, grounded in the intuitive and novel concept of exclusivity. Through controlled simulations, we demonstrated EDI to be well-calibrated, and better in comparison to existing metrics on non-linearity, resource efficiency and robustness under noise. These observations indicate a better suitability of EDI in supervised disentanglement measurement. However, it is essential to acknowledge that several pertinent questions remain open. Specifically, the development of unsupervised metrics has not progressed well, which has restricted the evaluation of disentangled representations in real-world scenarios. We hope and aim for further research in this direction to address this gap, as it holds promise for enhancing the practical utility of disentanglement evaluation in diverse contexts.

References

[1] Amir H. Abdi, Purang Abolmaesumi, and Sidney Fels. *A Preliminary Study of Disentanglement With Insights on the Inadequacy of Metrics*. 2019. arXiv: 1911.11791 [cs.LG].

[2] Hagai Attias. "Independent factor analysis". In: *Neural computation* 11.4 (1999), pp. 803–851.

[3] Mohamed Ishmael Belghazi et al. *MINE: Mutual Information Neural Estimation*. 2021. arXiv: 1801.04062 [cs.LG].

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. *Representation Learning: A Review and New Perspectives*. arXiv:1206.5538 [cs]. Apr. 2014. DOI: 10.48550/arXiv.1206.5538.

[5] Marc-André Carbonneau et al. *Measuring Disentanglement: A Review of Metrics*. arXiv:2012.09276 [cs]. May 2022.

[6] Ricky T. Q. Chen et al. *Isolating Sources of Disentanglement in Variational Autoencoders*. arXiv:1802.04942 [cs, stat]. Apr. 2019.

[7] Cian Eastwood and Christopher K. I. Williams. "A Framework for the Quantitative Evaluation of Disentangled Representations". en. In: Feb. 2022.

[8] Irina Higgins et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". en. In: (Apr. 2017).

[9] Irina Higgins et al. *Towards a Definition of Disentangled Representations*. arXiv:1812.02230 [cs, stat]. Dec. 2018.

[10] Hyunjik Kim and Andriy Mnih. *Disentangling by Factorising*. arXiv:1802.05983 [cs, stat]. July 2019.

[11] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. "Estimating mutual information". In: *Phys. Rev. E* 69 (6 2004), p. 066138. DOI: 10.1103/PhysRevE.69.066138.

[12] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. *Variational Inference of Disentangled Latent Concepts from Unlabeled Observations*. arXiv:1711.00848 [cs, stat]. Dec. 2018. DOI: 10.48550/arXiv.1711.00848.

[13] Zhiyuan Li et al. *Progressive Learning and Disentanglement of Hierarchical Representations*. arXiv:2002.10549 [cs, stat]. Feb. 2020.

[14] Francesco Locatello et al. *Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations*. arXiv:1811.12359 [cs, stat]. June 2019.

[15] Karl Ridgeway and Michael C. Mozer. *Learning Deep Disentangled Embeddings with the F-Statistic Loss*. arXiv:1802.05312 [cs, stat]. May 2018.

[16] Anna Sepliarskaia, Julia Kiseleva, and Maarten de Rijke. *How to Not Measure Disentanglement*. arXiv:1910.05587 [cs, stat]. Mar. 2021.

APPENDIX

A. Existing Metric Formulations

1) Z-diff (BetaVAE) Metric: Higgins *et al*. [8] introduced the BetaVAE based on the notion that dimensions capturing the constant generative factor should match, while others vary. This metric aims to capture modularity by computing the following steps:

(a) Selecting a generative factor z_k.
(b) Choosing a pair of samples, s_1 and s_2, with z_k constant while other factors vary.
(c) Generating latent codes c_1 and c_2.
(d) Calculating pairwise distortion:

$$e = (|c_{1,i} - c_{2,i}|), 1 \leq i \leq |z| \tag{1}$$

(e) Repeating the above steps to train a linear classifier predicting the fixed generative factor, with Z-diff indicating classifier precision.

2) Z-min Variance (FactorVAE) Metric: Kim *et al*. [10] proposed a metric similar to Z-diff, based on the assumption that latent codes capturing a constant generative factor should remain consistent. The method normalises each latent code by its dataset-wide standard deviation. The latent dimension with the least variance and the index of the constant factor form a sample for a linear classifier, assessing the classifier's precision.

3) Separated Attribute Predictability (SAP): Kumar *et al*. [12] developed the SAP metric, based on a matrix of informativeness I, with each entry $I_{i,k}$ representing a linear regression from latent code c_i to generative factor z_k. The SAP score is:

$$SAP(c,z) = \frac{1}{D}\sum_j \left(I_{i_k,k} - \max_{l \neq i_k} I_{l,k} \right), ; i_k = \arg\max_i I_{i,k} \tag{2}$$

4) Modularity Score: Ridgeway *et al*. [15] proposed a modularity metric of a latent cod c_i as:

$$modularity = 1 - \frac{\sum_{k \in \Omega_{\neq *}} I(k, c_k)}{I(z_*, c_k)^2 \times (M-1)}, \tag{3}$$

where z_* represents the factor that has the highest mutual information, $\Omega_{\neq *}$ denotes the set of all the generative factors except z_*, and M represents the number of factors.

5) MIG: The Mutual Information Gap (MIG), as detailed by Chen *et al*. [6], estimates disentanglement through the empirical mutual information between latent codes and generative factors:

$$\frac{1}{K}\sum_k \frac{1}{H(z_j)} \left(I(c_{i^*}; z_j) - \max_{i \neq i^*} I(c_i; z_j) \right) ,$$

where $i^* = \text{argmax}_i I(c_i; z_j)$, $\text{H}(z_j)$ is the entropy of z_j.

6) MIG-sup: As a complement to MIG, MIG-sup, introduced by Li *et al*. [13], addresses MIG's limitation regarding modularity. It averages differences between the top two mutual information values for each code and factor.

7) DCI: The idea behind DCI ([7]) is that it is possible to recover generative factors from latent units. Therefore, in order to compute *disentanglement*, *completeness* and *informativeness*, a model M trained to reconstruct generative factors from latent units is needed. The sub-model for predicting the generative factor z_j from latent codes c should be able to calculate the feature importance of each input latent code unit c_i and the feature importance is denoted as R_{ij}.

Disentanglement: The "probability" that c_i being important for predicting z_j in all factors is simulated as $P_{ij} = \frac{R_{ij}}{\sum_k R_{ik}}$. The disentanglement score for the code c_i is then calculated as:

$$D_i = 1 - \text{H}(P_{c_i}), \text{ where } \text{H}(P_{c_i}) = -\sum_k P_{ik} \log_K P_{ik}$$

Completeness: Similarly, the "probability" that c_i is important in all codes for predicting z_j is $P_{ij} = R_{ij}$. The completeness score for the generative factor z_j is then calculated as:

$$C_j = 1 - H(P_{z_j}), \text{ where } H(P_{z_j}) = -\sum_d P_{dj} \log_D P_{dj}$$

Informativeness: The informativeness of the generative factor z_j is estimated as the prediction error of z_j from the latent codes c.

$$I_j = \text{Error}(z_j, \hat{z}_j) = \text{Error}(z_j, M_j(c))$$

8) DCIMIG: DCIMIG or 3CharM claim to satisfy the three characters of disentanglement simultaneously ([16]). It is calculated as follows:

(a) calculate the disentanglement score for each latent code c_i as $D(c_i) = I(c_i; z_{j^*}) - \max_{j \neq j^*} I(c_i; z_j)$, where $j^* = \text{argmax}_j I(c_i; z_j)$.
(b) calculate the disentanglement score for each generative factor z_j as
$D(z_j) = \max_i D(c_i)$, where $j = j^*$ in calculating $D(c_i)$. That is, $D(z_j)$ is maximum value among the disentanglement scores of the codes that capture z_j. If no code capture z_j, $D(z_j) = 0$.
(c) 3CharM is then defined as $\frac{\sum_j D(z_j)}{\sum_j H(z_j)}$

B. Data

Shapes3D

Shapes3D [6]is a dataset of 3D shapes procedurally generated from 6 ground truth independent latent factors. These factors are:

- Floor (colour) hue: 10 values linearly spaced in $[0, 1]$
- Wall (colour) hue: 10 values linearly spaced in $[0, 1]$
- Object (colour) hue: 10 values linearly spaced in $[0, 1]$
- Scale: 8 values linearly spaced in $[0, 1]$
- Shape: 4 values in $[0, 1, 2, 3]$
- Orientation: 15 values linearly spaced in $[-30, 30]$

All possible combinations of these latents are present exactly once, generating $N = 480,000$ total images. All factors are sampled uniformly and independently of each other.

C. Model

We select latent variable models that enforce disentanglement by regularizing the encoding distribution $q_\phi(z|x)$ in the VAE. Theoretically, latent representations learned by the selected model should have better disentanglement than those learned by VAE. We select BetaVAE and FactorVAE for experimentation. For a fair comparison, we applied a common encoder/decoder architecture for all VAE variants, as described in Table III. For the discriminator in FactorVAE, we used the same model architecture as in FactorVAE: a feed forward neural network that has six hidden layers with 1000 neurons each, using a leaky ReLU of factor 0.2 as activation, and an output layer with two output units.

[6]https://github.com/google-deepmind/3d-shapes

Table III: Details of encoder and decoder architecture in the experiments

Encoder	Decoder
Input: $n_i \times 64 \times 64$	Input: n_c
Conv: $32 \times 4 \times 4$ (stride 2), ReLU	Linear: 256, ReLU
Conv: $32 \times 4 \times 4$ (stride 2), ReLU	Linear: 1024, ReLU
Conv: $64 \times 4 \times 4$ (stride 2), ReLU	ConvTranspose: $64 \times 4 \times 4$ (stride 2), ReLU
Conv: $64 \times 4 \times 4$ (stride 2), ReLU	ConvTranspose: $64 \times 4 \times 4$ (stride 2), ReLU
Linear: 256, ReLU	ConvTranspose: $32 \times 4 \times 4$ (stride 2), ReLU
Linear: $2 \times n_c$	ConvTranspose: $32 \times 4 \times 4$ (stride 2)

D. Training

Again for better comparison, we fixed all the training hyperparameters used to train VAE, as detailed in Table IV. All parameters are set as closely as possible to previous works [14, 10], while also taking into account actual training speed and performance as much as possible. In addition, we use Adam optimizer with learning rate 1e-4, $\beta_1 = 0.5$, $\beta_2 = 0.9$ for training the discriminator of FactorVAE.

Table IV: Training hyper-parameters

Parameter Key	Value
training epochs	128
batch size	64
optimizer	Adam: β_1 0.9, β_2 0.999
learning rate	1e-4
reconstruction loss	binary cross entropy

Of the data, 90% is used for training and the remaining 10% is used for testing. Considering the robustness, we train each model with 10 different random seeds. The metrics will examine the representation learned by each model, and finally we aggregate the evaluation results.

Table V: The basic test cases used in the experiment on calibration, where columns Mod, Comp and Expl indicate the low or high in modularity, compactness, and explicitness, respectively.

Mod	Comp	Expl	factors	codes	description
0	0	0	$z_1 z_2$	$c_1 c_2$	same as #001, with reduced information
0	0	1	$z_1 z_2$	$c_1 c_2$	$c_1 c_2$ together encode z
0	1	0	$z_1 z_2 z_3$	$c_1 c_2$	same as #011, with reduced information
0	1	1	$z_1 z_2 z_3$	$c_1 c_2$	c_1 encodes $z_1 z_2$, c_2 encodes z_3
1	0	0	$z_1 z_2$	$c_1 c_2 c_3$	same as #101, with reduced information
1	0	1	$z_1 z_2$	$c_1 c_2 c_3$	$c_1 c_2$ together encode z_1, c_3 encodes z_2
1	1	0	$z_1 z_2$	$c_1 c_2$	same as #111, with reduced information
1	1	1	$z_1 z_2$	$c_1 c_2$	c_1 encodes z_1, c_2 encodes z_2