An Empirical Study on Disentanglement of Negative-free Contrastive Learning

Jinkun Cao[3] **Ruiqian Nai**[1] **Qing Yang**[4] **Jialei Huang**[1] **Yang Gao**[1,2*]
Tsinghua University[1] Shanghai Qi-Zhi Institute[2]
Carnegie Mellon University[3] Shanghai Jiao Tong University[4]

Abstract

Negative-free contrastive learning methods have attracted a lot of attention with simplicity and impressive performances for large-scale pretraining. However, its disentanglement property remains unexplored. In this paper, we examine negative-free contrastive learning methods to study the disentanglement property empirically. We find that existing disentanglement metrics fail to make meaningful measurements for high-dimensional representation models, so we propose a new disentanglement metric based on Mutual Information between latent representations and data factors. With this proposed metric, we benchmark the disentanglement property of negative-free contrastive learning on both popular synthetic datasets and a real-world dataset CelebA. Our study shows that the investigated methods can learn a well-disentangled subset of representation. As far as we know, we are the first to extend the study of disentangled representation learning to high-dimensional representation space and introduce negative-free contrastive learning methods into this area. The source code of this paper is available at `https://github.com/noahcao/disentanglement_lib_med`.

1 Introduction

Existing disentangled representation learning methods are mostly generative models (25; 15; 24; 10; 8; 31). They are evaluated only on simple synthetic datasets (36; 5) and low representation dimensions, e.g., no more than 20-d. In contrast, contrastive learning is a class of discriminative methods, trained by pulling the representation of two augmentations of the same image close. This usually requires a much higher representation dimension for training, e.g., at least 1000-d.

Despite the success of contrastive learning, it remains unknown whether it can lead to disentangled representations. Recently, some works (50; 53) reveal that contrastive learning can approximately invert the data generation process and allow the learned representation to have identifiability, which is related to the disentanglement property. However, all these advances necessarily rely on the contrast provided by negative samples. The disentanglement property of contrastive learning without negatives, or "non-contrastive self-supervised learning", remains unexplored. And no contrastive learning methods have been evaluated on the standard disentanglement benchmarks (36; 5) yet.

Representation disentanglement requires each single dimension of a representation vector to be correlated to only one independent variable of the input data, which we call a data factor. There lacks a uniform convention of measuring "correlation" here, so multiple disentanglement metrics (9; 15; 13) have been proposed for the quantitative evaluation. Their shared principle is that a well-disentangled representation model should not have a representation dimension responding to the change of more than one data factor. We find although multiple disentanglement metrics' (15; 24; 6; 9; 27) evaluations are in agreement for low-dimensional space (34), they disagree in high-dimensional space. Moreover, their design limitation makes the measurement not meaningful in high-dimensional space. We thus propose a new disentanglement evaluation metric that is robust to representation dimension scaling

*corresponding author

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

and is thus adaptable to a high-dimensional representation model. The new metric is named as "**M**utual information based **E**ntropy **D**isentanglement" and MED for short.

With the proposed metric, we find that negative-free contrastive learning methods can achieve good disentanglement in a subset of latent dimensions. Given that the recent disentanglement studies are limited to simple synthetic datasets but contrastive learning is especially powerful for complicated and large-scale datasets, we also extend the quantitative benchmarking to a real-world dataset CelebA (33). On CelebA, existing low-dimensional generative disentangled learning methods are unable to learn good representations, demonstrating the gap between current disentangled learning research and real-world data complexity. To summarize, our contributions in the work are three-fold:

1. We find that existing disentanglement metrics fail to extend to high-dimensional representation space, and we propose a new metric MED to extend this area to high-dimensional space.

2. We extend the study of disentangled representation learning to real-world complicated datasets and high-dimensional representation space, revealing the gap between current disentangled representation learning and real-world data complexity.

3. We empirically study the disentanglement property of contrastive learning without negatives. We find it can learn a well-disentangled subspace of latent representation.

2 Related Works

Disentangled Representation Learning. Learning a disentangled representation is a long-desired goal in the deep learning community (4; 37; 11; 3; 42; 28; 47). A disentangled representation matches how humans understand the world and allows us to use much fewer labels to solve challenging downstream tasks (49). There are two lines of work related to this goal, i.e., the Independent Component Analysis (ICA) and the Disentangled Deep Representation Learning. ICA (18) usually assumes that the pattern of noise (17; 22) or some additional auxiliary variables (19; 23) can be observed. On the other hand, deep representation learning makes no explicit assumption on the noise distribution or representation prior and usually emphasizes unsupervised learning. Under this setting, deep representation learning is usually based on deep generative models such as VAE-based methods (15; 24; 6; 27) and Generative Adversarial Networks (GAN) (10; 8). These two lines of study focus on different but relevant (26) aspects of a learned representation. In this paper, we focus on empirically studying the disentanglement property of negative-free contrastive learning methods to introduce them into the scope of disentangled representation learning.

Disentanglement Metrics. The variation in the metric used also shows the difference between ICA and Disentangled Deep Representation Learning. ICA aims to achieve good identifiability and uses Mean Correlation Coefficient (MCC) as the common metric. On the other hand, the metrics used in the deep disentangled representation learning community are very diverse. Until now, no widely accepted definition of "disentanglement" is available. So the empirical agreement of metrics makes the basis of quantitatively evaluating disentanglement. DisLib (34) summarizes six popular disentanglement metrics, i.e., DCI (9), SAP (27), MIG (6), BetaVAE score (15), FactorVAE score (24) and Modularity (41). DisLib finds that the five metrics except for Modularity have good agreement in evaluating disentanglement quality. However, all of the metrics are evaluated on low-dimensional latent space, which is around 10 dimensions. We find severe problems when applying those metrics to high-dimensional latent spaces where they show significant disagreement. To make a meaningful evaluation of the disentanglement property of contrastive learning methods, we thus propose a new metric, which is more applicable to high-dimensional representation space.

Contrastive Learning and Representation Disentanglement. Contrastive learning (CL) creates "views" by augmentations over images. Views of the same image serve as positives, and views of other images as negatives. Recently, some works try to understand contrastive learning theoretically (50; 1; 46; 45; 30) or empirically (43; 52; 38). Zimmermann et al. suggests that the contrastive method inverts a data generation process when infinite negative samples are available (50), which is related to the disentanglement property. However, there is still no work connecting CL methods with standard disentanglement benchmarks. On the other hand, negative-free CL methods (40; 7; 51) do not use the contrast between positive and negative samples to encourage discriminative representations. They generate positive "views" of data by applying different augmentations to the same input data, and the pair of views is forwarded into two network streams. Recent works on this line have their own unique designs. BYOL (40) discovers that self-supervised learning can avoid trivial solutions, i.e.,

"model collapse", even without using negative samples to provide contrast. The key of BYOL is to add a predictor layer following the commonly adopted "encoder-projector" network of contrastive learning methods (14). This provides additional asymmetry. As a follow-up, SimSiam (7) removes the momentum update from BYOL and proves that combining the predictor and the stop-gradient design is enough for self-supervised learning to learn non-trivial representation. More recently, Barlow Twins (51) shows that even the predictor or the trick of stop-gradient is not necessary. Negative-free CL methods have their own advantages, such as avoiding model collapse (44; 16; 21), but its disentanglement property remains unexplored, either empirically or theoretically. In this paper, our focus thus is to benchmark the disentanglement property of these methods for the first time which is possible after we propose a new disentanglement metric applicable to the high-dimensional representation models.

3 The Proposed Disentanglement Metric: MED

Positive-negative contrast in self-supervised learning is seen to encourage learned representation to have uniformity on a hyper-sphere (50). When negative views are not available, the disentanglement property of the learned representation remains a mystery. Therefore, we examine the mentioned negative-free contrastive learning methods in an empirical study to reveal this property of interest. However, we demonstrate in Section 3.1 that existing disentanglement metrics can not evaluate CL methods in high-dimensional space fairly or meaningfully. Therefore, we introduce our proposed MI-based Entropy Disentanglement score (MED) in Section 3.2 and its variant to evaluate the disentanglement of a subspace of representation in Section 3.3.

3.1 Failure of Existing Metrics on High-Dimensional Space

Typical contrastive learning methods need a high dimensional representation space ("latent space") to train well. However, the previous study of disentangled representation learning only deals with low-dimensional representation space. For example, in Locatello et al., the latent space dimension is set to 10. So, existing disentanglement metrics (see Appendix C for details) are designed for low-dimensional representation models and have intrinsic flaws when evaluating models in high-dimensional space. To be precise, we have observations below:

- Metrics based on learnable classifiers, such as **BetaVAE score** and **FactorVAE score**, allow unfair advantages to a high-dimensional model whose redundant parameters can trick the classifier more easily. For example, a randomly initialized 1000-d model could reach a FactorVAE score of 61.4 on dSprites, close to many well-trained 10-d VAE-based models' scores (see Table 4 in Appendix).
- Metrics taking only one or two dimensions into score calculation, such as **SAP** and **MIG**, are biased to representations of different dimensions. Because a higher dimension makes it harder for an informative dimension to stand out and enjoy a large informativeness gap over other dimensions.
- **DCI Disentanglement score** uses a learnable regressor to score the importance of each latent dimension to each data factor. The learnable regressor, such as Gradient Boosting Tree (GBT), encourages sparsity in the output importance matrix (see Figure 7 in Appendix). So, it also gives an advantage to high-dimensional models, making it unfair to compare models of different latent dimensions. Moreover, the construction of regressors is time-intensive in high-dimension space. For example, it usually takes hours to evaluate a 1000-d representation model by DCI using GBT.

These flaws are demonstrated by our experiments, showing that existing metrics disagree for high-dimensional representations. Besides our conceptual justification of the failure of these existing disentanglement metrics, we also construct some scenarios where their failure is theoretically demonstrated in Appendix F. Overall, through our experimental evidence and theoretical justification, we show that existing disentanglement metrics can no longer make meaningful disentanglement measurements in the high-dimensional representation space.

3.2 Mutual Information based Entropy Disentanglement

Given the bias and limitations of existing disentanglement metrics and the necessity of high-dimensional representations for contrastive learning, we propose a new disentanglement metric for high dimensional latent spaces, which we name as "**M**utual Information based **E**ntropy **D**isentanglement", or MED in short. The calculation of MED is based on mutual information (MI) between latent dimensions and the set of data factors of input samples. MI is a widely accepted tool to measure the correlation of variables and is not biased to the models of different dimensions. Given

a dataset generated with K ground truth factors $\boldsymbol{v} \in \mathbb{R}^K$ and a representation vector $\boldsymbol{c} \in \mathbb{R}^D$, we construct an importance matrix $R \in \mathbb{R}^{D \times K}$ defined by

$$R_{ij} = I(\boldsymbol{c}_i, \boldsymbol{v}_j) / \sum_{d=0}^{D-1} I(\boldsymbol{c}_d, \boldsymbol{v}_j), \tag{1}$$

where $I(\boldsymbol{c}_i, \boldsymbol{v}_j)$ denotes the mutual information between the ith latent dimension $\boldsymbol{c}_i$ and the jth ground truth factor $\boldsymbol{v}_j$. Here, each row denotes a representation dimension and each column represents a ground truth factor. We normalize the mutual information by columns, such that an entry in the matrix indicates the relative importance of one dimension over all dimensions regarding a certain data factor. This normalization is necessary since different dimensions may have different overall informativeness to all factors. This operation fixes the gap between models of different dimensions by estimating the relative importance of a single dimension.

After normalizing over the columns, we evaluate the contribution of a dimension to different factors, which is described by a row of R. If one dimension is informative to only one ground truth factor, then this dimension is perfectly disentangled. This matches the mechanism of entropy. So we use the entropy to describe the disentanglement level of a dimension. We treat each row as a discrete distribution over factor index by normalization: $P_{ij} = R_{ij} / \sum_{k=0}^{K-1} R_{ik}$, where higher probability P_{ij} indicates that the dimension $\boldsymbol{c}_i$ encodes more information of the factor $\boldsymbol{v}_j$. Then the disentanglement score S_i for a latent dimension $\boldsymbol{c}_i$ is calculated as

$$S_i = 1 - H_K\left(P_{i\cdot}\right), \tag{2}$$

where $H_K\left(P_{i\cdot}\right) = -\sum_{k=0}^{K-1} P_{ik} \log P_{ik}$ is the entropy. S_i will be higher if $\boldsymbol{c}_i$ exhibits more informativeness to one factor while less relevance to other factors. Finally, to summarize the overall disentanglement of a representation model, MED score is the weighted average of the disentanglement scores for all dimensions as

$$\text{MED}\left(\boldsymbol{c}\right) = \sum_{i=0}^{D-1} \rho_i S_i \tag{3}$$

where $\rho_i = \sum_j R_{ij} / \sum_{ij} R_{ij}$ is the relative importance of each dimension.

Our proposed MED does not use a learnable classifier or a regressor. Further, it inherits a DCI-style normalized importance matrix by taking all dimensions into account instead of using only one or two dimensions. These characteristics allow MED to be more robust to the latent dimensionality and to be better suited to meaningful disentanglement measurement in high-dimensional space. Besides the advantages MED has, it is also very computationally efficient: for a 1000-d BYOL representation model on Cars3D, the evaluation with DCI with Gradient Boost Tree by DisLib takes more than 14 hours, while MED only takes less than 20 seconds on the same machine.

3.3 Partial Disentanglement Evaluation Metric

It is challenging to learn a fully disentangled representation by high-dimensional models without explicitly encouraging disentanglement, especially when there are fewer independent data factors than the number of latent dimensions. However, if a high-dimensional model has a subset of dimensions that disentangle well, it is still worth studying. Such a subset can serve as a proxy to build a more compact representation model when it is hard to train such a model directly. This motivates us to design a version of MED to evaluate the partial disentanglement, which we name “Top-k MED".

The main difference is that we pick the most disentangled k dimensions for each factor and compute MED on this subset of representations. For each ground truth factor $\boldsymbol{v}_j$, $\mathcal{G}_j = \{i | \arg\max_m R_{im} = j\}$ is the set of latent dimensions emphasizing this factor. Then we pick the top k latent dimensions with the highest disentanglement scores in each $\mathcal{G}_j$ to construct a subset $\mathcal{P}_j = \left\{i | S_i \geq S^k,\ i \in \mathcal{G}_j\right\}$. Here, S_i is the disentanglement score of the latent dimension $\boldsymbol{c}_i$ defined in Equation 2. S^k is the kth highest disentanglement score in $\mathcal{G}_j$. Finally we obtain the subset $\mathcal{P} = \cup_j \mathcal{P}_j$. We take the sub-vector after selection $\tilde{\boldsymbol{c}} = \{\boldsymbol{c}_i\}_{i \in \mathcal{P}}$ as the representation to evaluate the MEDscore. The top-k MED is defined as MED($\tilde{\boldsymbol{c}}$).

4 Understanding the learned representation

In this part, we qualitatively study the disentanglement of the learned representation by negative-free contrastive learning with BYOL as an example. Since contrastive methods are not generative models, it is hard to directly do factor-controlled pixel-wise reconstruction for visualization. Instead, we

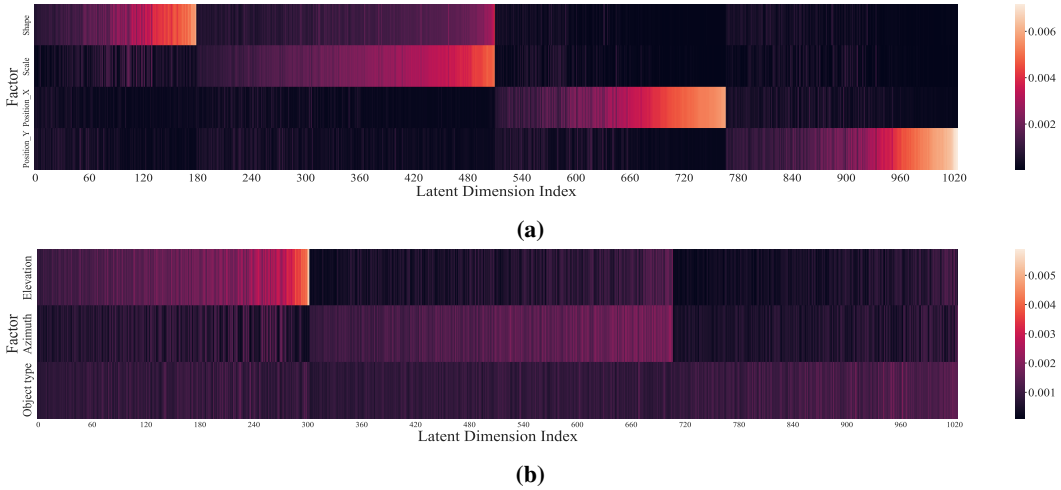


Figure 1: The mutual information heatmap between factors and the BYOL's latent dimensions on dSprites (a) and Cars3D (b). Each row is a ground truth factor and each column is a learned latent dimension. We normalize each row as we do in MED. Brighter straps denote higher MI. Latent dimensions are sorted for visualization.

measure the mutual information between the learned representations and ground truth factors, which is also the foundation of our proposed MED metric. More qualitative study is available in Appendix B.

4.1 Correspondence of Representation and Factor

To understand the disentanglement of a representation model, a basic question is how the representation dimensions correspond to data factors. After encoding an input image to a representation vector, we compute the normalized mutual information (MI) between each ground truth factor and each representation dimension, i.e. R in MED defined as in Equation 1, to measure the correspondence.

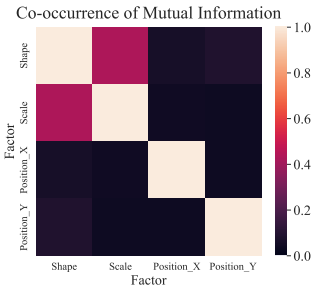


Figure 2: The visualization of normalized co-occurrence of mutual information on dSprites.

The mutual information between dimensions and the factors is included in Figure 1. The heatmaps are actually the transpose of the importance matrix $R^\top$. As shown by the brightness of the entries, the informativeness of the latent dimensions varies greatly. In fact, we can identify three types of columns: columns with *single*, *multiple*, and *no* bright elements. They correspond to three types of latent dimensions: *disentangled* dimensions, *entangled* dimensions, and *uninformative* dimensions. The *disentangled* dimensions only capture information about one factor while the *entangled* dimensions encode multiple factors. In contrast, the *uninformative* dimensions cannot represent factors independently. As there are still *entangled* and *uninformative* dimensions, the representations are not fully disentangled. However, by extracting the subset of *disentangled* dimensions we can derive a well-disentangled subspace.

First, we analyze the model on dSprites dataset (36). dSprites has five factors (shape, scale, orientation, position_x, and position_y). But the orientation is ill-defined with ambiguity. For example, it is impossible to distinguish if a square rotates 0 degrees or 180 degrees. As shown in Figure 1a, the *disentangled* dimensions occupy a significant proportion, indicating an evident partially disentangled pattern. We note that the degree of disentanglement varies on different datasets. An example on Cars3D is shown in Figure 1b. Cars3D is a dataset with 183 different car objects rendered from 4 elevations and 24 azimuths and the ground truth factors are not fully independent of Cars3D (see Appendix B). It is extremely hard to represent its azimuth and the type of cars with few dimensions. Thus its object-type row and azimuth row in Figure 1b are more spread out among multiple latent dimensions. This also shows the difficulty of understanding the disentanglement of the high-dimensional representation model on complicated datasets. Therefore, we will continue to conduct a quantitative evaluation with our proposed MED metric in Section 5. Further quantitative studies on more datasets are available in Appendix B.

4.2 Uniqueness of Factor-Representation Correspondence

In the ideal pattern of disentanglement, a representation dimension should uniquely correspond to only one factor. Now we show to what extent multiple factors are responded to by a single representation dimension in a well-disentangled subspace. Given the mutual information between the representation and the ith factor, noted as I_i, i.e., the ith row in the top-k version of Figure 1a, a good indicator of the uniqueness of factor-representation correspondence is the normalized co-occurrence of mutual information between the i_1^{th} factor and the i_2^{th} factor, which is defined as

$$\widehat{C}_{i_1,i_2} = \frac{\langle I_{i_1}, I_{i_2} \rangle}{||I_{i_1}||_2 \cdot ||I_{i_2}||_2} = \frac{\sum_{d=0}^{K \cdot k-1} I(\tilde{\boldsymbol{c}}_d, \boldsymbol{v}_{i_1}) I(\tilde{\boldsymbol{c}}_d, \boldsymbol{v}_{i_2})}{||I_{i_1}||_2 \cdot ||I_{i_2}||_2}, \tag{4}$$

where $\tilde{\boldsymbol{c}}$ is the representation after the selection process in Section 3.3. We visualize the normalized co-occurrence of mutual information among the four factors by the learned representation in Figure 2. It agrees that a dimension usually encodes only one factor. Moreover, it indicates that the learned representation tends to encode shape and scale together, which also agrees with the intuitive analysis of the independence of factor pairs. For example, the shape and scale of dSprites objects are not disentangled and independent because objects with the same scale value but in different shapes have different pixel areas.

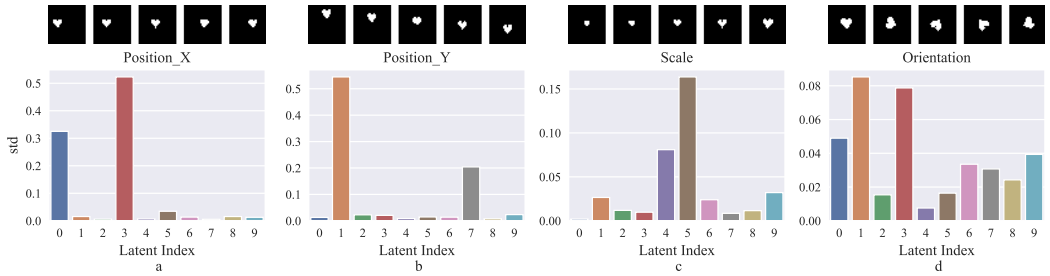


Figure 3: Representation variation when manipulating one factor only in the dimension-reduced version. In (a), (b), (c), *position_x*, *position_y* and *scale* are manipulated respectively and only cause one dimension significantly variate. While, in (d), when manipulating the ill-defined factor *orientation*, multiple dimensions variate.

4.3 Influence by Manipulating Factors

Another intuitive direction to study the relationship between representations and factors is the influence on representations when manipulating the factors. Given that the original representation vector dimension is much higher than the number of factors, we first make the representation more compact to have a more concise illustration. Here, we reduce the representation dimension by the selection process of top-k MED described in Section 3.3.

Figure 3 shows the result of representation vector variation when changing only one factor at once. We take BYOL on dSprites as a representative. Here we set $k = 2$ for top-k MED, i.e., we pick the 2 most disentangled dimensions for each factor and derive a 10-dim representation. We sample a set of images regarding a specific data factor. These images have all possible values of this data factor while having the same value for all other data factors. We get 10-d representations from these generated images by a shared model. Then, we compute the variance of each of the 10 dimensions across the images, leading to 10 scalars. The larger the variance is, the more that dimension responds to the factor. Figure 3(a), (b) and (c) show how the reduced representation vector changes when manipulating the *position_x*, *position_y*, and *scale* factors respectively. Note that we set $k = 2$, therefore we see good disentanglement, with only exactly two representation dimensions having high variation. However, in Figure 3(d) we show a failure mode of the ill-defined factor *orientation* that the change of factor causes multiple dimensions of reduced representation to have large variations, indicating that this factor is represented in an entangled way. From the results, we observe that manipulating different well-defined independent factors causes evident variance on disjoint sets of dimensions. Further, these results demonstrate the existence of a well-disentangled subset of latent dimensions. We also conduct a similar qualitative study where the representation dimension is reduced by the unsupervised PCA technique and observe a similar pattern. The details are provided in Appendix B.4.

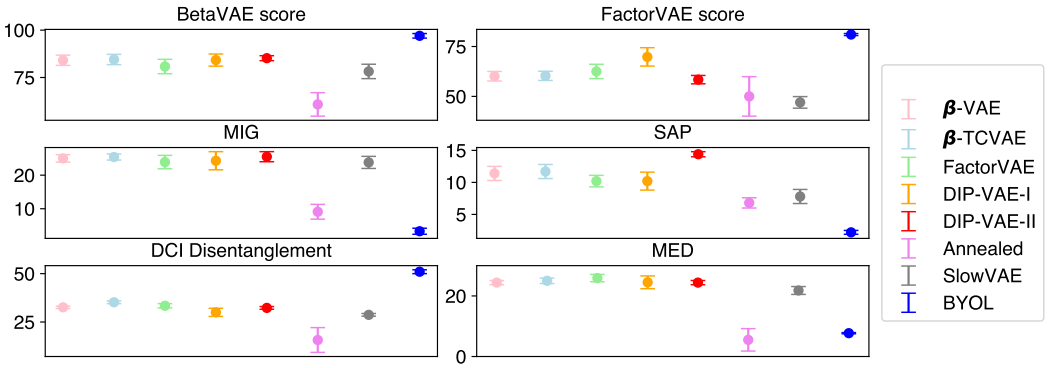


Figure 4: Evaluation with multiple metrics on SmallNORB. Y-axis is the corresponding disentanglement scores. The result shows the disagreement among existing metrics in high-dimensional space (BYOL).

5 Quantitative Evaluation

In this section, we conduct quantitative studies on the disentanglement property of contrastive learning methods. We first introduce the experiment setup in Section 5.1. Then we show quantitative results under both MED and existing disentanglement metrics in Section 5.2 which shows the disagreement of existing metrics to support the necessity of proposing MED. Finally, we make a full quantitative benchmark of methods of interest with MED in Section 5.3 and an ablation study about the dimension of the representation model in Section 5.5. More quantitative studies are available in Appendix C.

5.1 Experiments Setup

The details for reproducibility are introduced in Appendix A. Here we provide a brief description of the setup of the experiments.

Datasets. Representation disentanglement is usually evaluated on synthetic datasets, such as dSprites (36), Cars3D (39), Shapes3D (5), and SmallNORB (29). Besides those datasets, we also include a real-world dataset CelebA (32). CelebA contains human face images with 40 binary attributes. The attributes include fine-grained properties of the human face, such as whether wearing glasses or having wavy hair. We include the details of dataset factors in the appendix.

Evaluation Protocol. We conduct experiments with both MED and the existing metrics to reveal the disagreement between existing metrics. Then we use MED as the main metric to study the disentanglement of contrastive learning methods. The implementation of evaluation metrics is adapted from the protocol provided by DisLib (34). All results are calculated with three random seeds and we report both the average score and the standard deviation. More details are introduced in Appendix A.2.

Reference Methods. We investigate most of the popular disentangled representation learning methods as studied in the standard benchmark of DisLib (34). Besides, we also compare with a recently proposed ICA method called ICE-BeeM (23). Since we do not assume the ground truth factors are known during the training, we use its unconditional version. We term it EBM (energy-based model). For the contrastive learning methods, we evaluate not only negative-free methods such as BYOL (40), Barlow Twins (51) and SimSiam (7), but also those using negative samples such as MoCo and MoCov2 (14).

Model Implementation. All methods use a shared architecture of encoder network as explained in the appendix. The latent dimension of contrastive learning methods is set to 1000 since they require a high-dimensional latent space to work. For other methods, the latent dimension is set to be 10 on synthetic datasets as in DisLib and 128 on CelebA dataset when evaluating with MED. For the evaluation with Top-k MED, the dimension of all methods is set to 1000-d for fairness. On dSprites, Cars3D and SmallNORB, we acquire checkpoints from DisLib if they are provided. We train our checkpoints on CelebA and Shapes3D.

5.2 Disagreement of Existing Metrics

As there is not a uniform definition of "disentanglement", existing disentanglement metrics are motivated by different desired properties of a disentangled representation. These metrics had previously shown good agreement in the large-scale experiments of DisLib (34). However, when we extend the

disentanglement study beyond low-dimensional scenarios, these metrics disagree significantly. We select representative low-dimensional VAE-based methods and a high-dimensional BYOL model to evaluate on a representative dataset, the SmallNORB dataset. The results are in Figure 4, which show a significant disagreement among metrics on BYOL, while they agree on the low-dimensional VAE methods. Aligned with our analysis in Section 3.2, BetaVAE score, FactorVAE score, and DCI overestimate the disentanglement degree of the high-dimensional model while MIG and SAP underestimate it. Since existing metrics fail to evaluate high-dimensional models, we opt to use MED as the main evaluation metric in the following sections. Please refer to Appendix C and E for more evidence of the disagreement of existing metrics. Moreover, in Appendix F, we provide the analysis by constructing scenarios where MED can still output the result aligned with human intuition while existing metrics fail to conduct a meaningful measurement of disentanglement degree.

Table 1: MED and Top-k MED scores on multiple datasets. Methods in gray are contrastive self-supervised learning methods. The latent dimension of VAE methods with * is 1000.

Metrics	Model	dSprites	Shapes3D	Cars3D	SmallNORB	CelebA
MED	β-VAE	32.6 (10.0)	52.5 (9.4)	29.0 (2.2)	24.4 (0.7)	3.3 (0.5)
	β-TCVAE	31.8 (7.4)	53.2 (4.9)	**33.0 (3.8)**	25.0 (0.9)	4.7 (0.1)
	FactorVAE	32.5 (10.1)	55.9 (8.0)	29.1 (3.0)	**25.9 (1.2)**	0.6 (0.6)
	DIP-VAE-I	18.8 (5.6)	43.5 (3.7)	19.4 (3.3)	24.5 (2.1)	3.7 (0.2)
	DIP-VAE-II	14.7 (5.5)	52.6 (5.2)	16.7 (4.1)	24.4 (0.6)	–
	AnnealedVAE	**35.8 (0.8)**	**56.1 (1.5)**	15.5 (2.5)	5.5 (3.7)	–
	EBM	6.8 (4.0)	2.1 (2.6)	–	2.3 (1.7)	–
	MoCo	4.2 (0.5)	6.1 (0.1)	8.6 (0.4)	4.9 (0.1)	**5.8 (0.1)**
	MoCov2	3.5 (1.4)	4.2 (0.4)	6.5 (0.4)	3.3 (0.2)	4.8 (0.2)
	BarlowTwins	6.0 (0.3)	6.4 (0.3)	5.6 (1.4)	6.1 (0.1)	4.2 (0.2)
	SimSiam	26.5 (0.1)	12.3 (0.9)	10.4 (0.1)	10.7 (1.1)	5.3 (0.4)
	BYOL	31.3 (0.4)	6.0 (0.5)	9.7 (0.5)	7.7 (0.2)	4.8 (0.4)
Top-k MED	β-VAE*	16.6 (6.2)	19.2 (1.4)	29.2 (2.0)	15.8 (2.1)	4.5 (0.3)
	β-TCVAE*	11.2 (0.4)	25.0 (0.5)	20.0 (1.8)	23.0 (0.6)	3.6 (0.2)
	FactorVAE*	3.2 (3.8)	8.2 (4.0)	7.8 (1.6)	4.8 (1.0)	5.0 (0.3)
	DIP-VAE-I*	7.0 (1.3)	16.2 (0.9)	24.6 (2.2)	20.9 (2.7)	2.5 (0.9)
	MoCo	16.1 (2.0)	18.1 (0.6)	26.6 (1.6)	17.9 (0.8)	**7.9 (0.1)**
	MoCov2	14.7 (1.0)	13.6 (1.7)	24.5 (2.1)	15.1 (0.9)	6.6 (0.7)
	BarlowTwins	21.7 (1.3)	20.0 (0.3)	23.8 (2.5)	24.5 (1.5)	5.7 (0.2)
	SimSiam	39.1 (0.4)	**30.0 (2.0)**	**32.7 (2.3)**	**28.4 (1.9)**	7.2 (0.6)
	BYOL	**53.7 (0.7)**	19.7 (1.3)	31.8 (1.3)	25.7 (0.3)	6.8 (0.7)

5.3 Disentanglement Benchmark with Contrastive Learning Methods

For the benchmarking of disentanglement, we use both MED and the partial version of MED, i.e. Top-k MED. For Top-k MED, we set the MED partial evaluation hyperparameter $k = 2$ for dSprites, Shape3D, and SmallNORB, and $k = 3$ for Cars3D and CelebA. The values of k are chosen such that the selected dimensions are roughly close to the latent space dimension of the low-dimensional reference methods. And we extend the latent dimension of all methods to 1000 when evaluating Top-k MED for fairness. We note that despite our hyperparameter search, we were unable to train good EBM weights on Cars3D and CelebA, so we keep that section empty. The results are in Table 1. We also encourage readers to read the results of existing metrics in Table 4 in Appendix C.

In the upper part of Table 1, we show the MED score for previous disentangled methods as well as contrastive methods. We find that contrastive methods achieve significantly lower disentanglement scores on 3 of the 5 datasets (Shapes3D, Cars3D, SmallNORB). Contrastive methods achieve slightly higher disentanglement scores on CelebA. On dSprites, some negative-free contrastive methods (SimSiam, BYOL) achieve scores close to SOTA, but the other contrastive methods' score is much lower. In summary, in most cases, contrastive methods have inferior disentanglement properties compared to the best methods; only in a few settings do contrastive methods achieve scores comparable to SOTA scores.

These results are disappointing but not surprising since contrastive methods are not explicitly designed to maximize the feature disentanglement. Further, since the underlying number of factors is usually quite small, on the order of 10, the 1000-d feature space will likely have dimensions that are either not related to the ground truth factors or capture a combination of the ground truth factors. This result

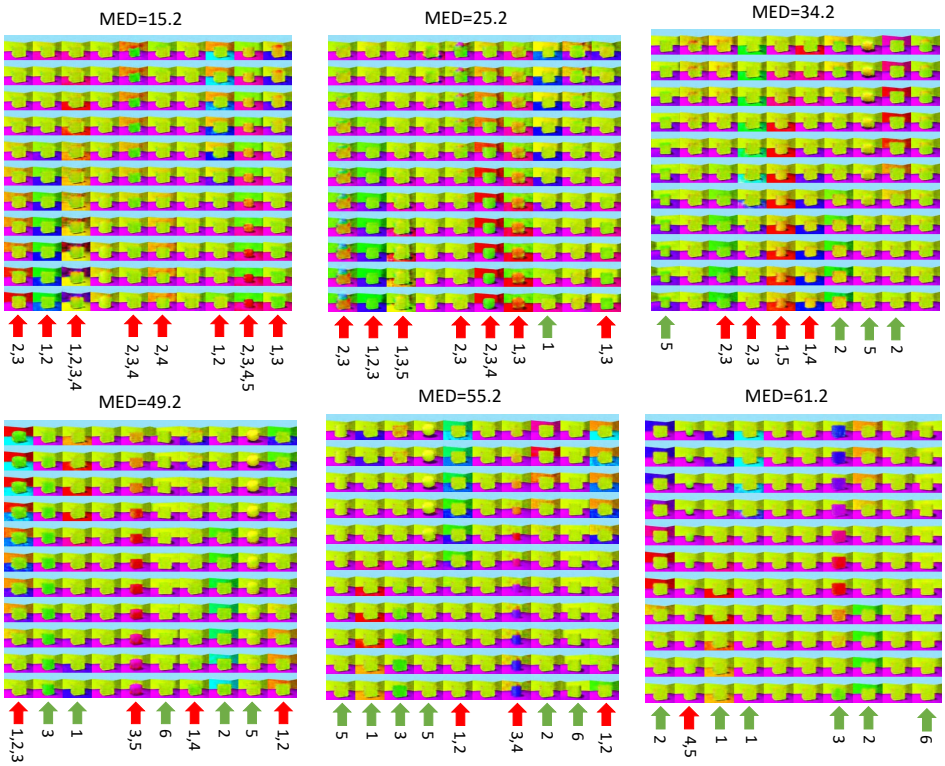


Figure 5: The visualization of latent traversing by VAE models from Van Steenkiste et al.. Each subfigure is the visualization for one model, annotated with the MED score of that model. The involved data factors include (1) Floor color; (2) Wall color; (3) Object color; (4) Object scale; (5) Object shape and (6) Orientation. For each column in the subfigure, the value of only one latent dimension changes, i.e. latent traversal for that dimension. Below each column, we also annotate the actual changing factors in this column, obtained with human manual inspection. If a model is fully disentangled, we expect each column to correspond to only one ground truth factor. The visualization shows that if a model achieves a higher MED score, it has more dimensions disentangled and responsive to only one factor while fewer dimensions entangled to multiple factors.

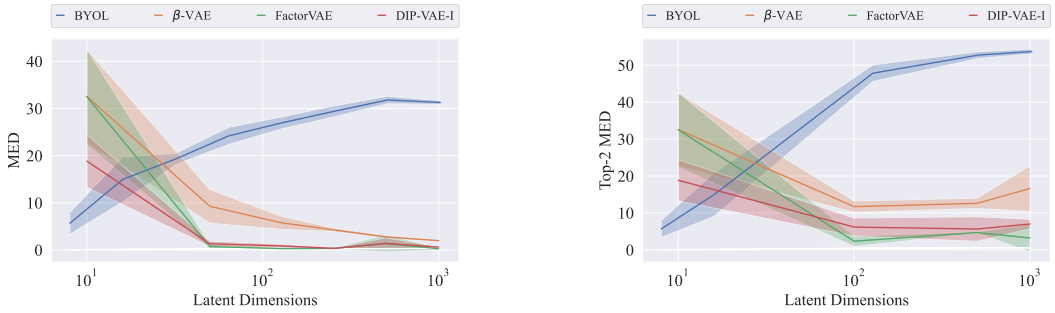
does not contradict with Zimmermann et al. (53) since (1) they find that the learned feature is a linear transformation of the ground truth factors, which doesn't necessarily disentangle, and (2) they use augmentations on factors that cannot be done in practice.

The lower part of Table 1 shows top-k MED measurements on various methods. We find that contrastive learning methods (especially the negative-free ones) in general show a better disentanglement in a selected subspace and the disentanglement is stronger than the reference methods. This shows that there exists a subspace in the learned representation that is well disentangled. Moreover, when we compare the subspace in contrastive methods (gray part in the lower section of Table 5.3) to the traditional approach that directly trains a low-dimensional latent space (non-gray part in the upper section of Table 5.3), we find that the disentanglement of the former is usually better than or on par with that of the latter. This means that we probably should not constrain the dimension of the latent space and require it to be fully disentangled, but rather should use high dimensional latent spaces and only require it to have a subset with good disentanglement properties.

To conclude, we find that the high-dimensional contrastive methods, including negative-free ones, do not learn a fully disentangled representation. However, there exists a subspace in the learned representation that is well disentangled. Such a subspace can show much better disentanglement properties than previous SOTA approaches. Despite the fact that these methods require a high dimension to train, such a subspace can serve as a proxy between contrastive learning methods and a more compact and disentangled low-dimensional representation.

5.4 Latent Traverse on Shapes3D

Now, we provide visualizations to show that MED can provide evaluation results aligned with human intuition about the disentanglement degree regarding data factors. Because CL methods



(a) MED score w/ dimension change.

(b) Top-2 MED score w/ dimension change.

Figure 6: The influence of the representation dimension on the (a) MED score and (b) Top-2 MED score on dSprites dataset. The disentanglement property regarding both metrics is enhanced with increasing representation dimension for the BYOL method but decreased for all tested VAE-based methods. The minimum dimension is 10 for VAE-based methods, and 8 for BYOL.

are not generative models and have high dimensions, they are hard to be adopted for visualization with latent traversal. We thus adopt VAE-based generative models here to perform latent traverse. We follow the practice in DisLib (34) to use Shapes3D for the latent traverse. For fairness and reproducibility, we trained VAE models on Shapes3D with the published configurations provided by Van Steenkiste et al. because DisLib (34) does not provide the raw experiment logs on Shapes3D. The results are shown in Figure 5. For each column in each subfigure, only the value of one dimension of the latent code is manipulated. The manipulation is performed the same way as the default setup for traverse visualization in DisLib. Given the six factors on Shapes3D with index, we indicate under each subfigure if (1) red arrow: a dimension is entangled to more than one factor or (2) green arrow: a dimension is disentangled and responsive to only one factor. Through the visualizations and the corresponding MED scores, we can clearly see that MED can well represent the disentanglement degree. We could observe a clear pattern that the model with a higher MED score has more disentangled representations. This demonstrates that the results from MED scores are aligned with the intuition of humans.

5.5 Influence of Dimension

As representation dimension is found as a core variable in disentangled representation learning, we evaluate the influence of dimension over disentanglement by both MED and top-k MED on dSprites with BYOL as an example versus VAE-based methods. The results are shown in Figure 6. We find that the BYOL's MED score and top-2 MED score increase along with the latent dimension. The scores plateau at around 512 dimensions. This is consistent with previous literature on the difficulty to train informative contrastive models with low latent dimensions (12). We further show that a lower latent dimension leads to a less disentangled subspace as well. On the contrary, we also note that the VAE methods fail to scale to higher latent dimensions. We find a large gap between higher-dimensional VAEs and their 10-dim versions. This suggests the gap between existing disentangled representation methods and the real-world data complexity, which can not be represented in a limited dimension.

6 Conclusion

In this paper, we provide an empirical study of the disentanglement property of contrastive learning without negatives for the first time. In the high-dimensional space, we find the difficulty of adopting the existing disentanglement metrics. Therefore, we propose MED and Top-K MED to evaluate disentanglement based on mutual information. The evaluation shows that even without negative samples, contrastive learning can learn a well-disentangled subset of representation. Recently, the study of contrastive learning, or general self-supervised learning, is still motivated by empirical observations. We hope our work can reveal some clues to motivate future theoretical justifications.

Acknowledgement

We appreciate the help from Jinhyung Park on paper writing. This work is supported by the Ministry of Science and Technology of the People's Republic of China, the 2030 Innovation Megaprojects "Program on New Generation Artificial Intelligence" (Grant No. 2021AAA0150000). This work is also supported by a grant from the Guoqiang Institute, Tsinghua University.

References

[1] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

[2] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[3] Y. Bengio, Y. LeCun, et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5):1–41, 2007.

[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[5] C. Burgess and H. Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

[6] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018.

[7] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[8] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016.

[9] C. Eastwood and C. K. Williams. A framework for the quantitative evaluation of disentangled representations. In *ICML*, 2018.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[11] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[12] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[13] H. Hälvä, S. L. Corff, L. Lehéricy, J. So, Y. Zhu, E. Gassiat, and A. Hyvarinen. Disentangling identifiable features from noisy data with structured nonlinear ica. *arXiv preprint arXiv:2106.09620*, 2021.

[14] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[15] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *arXiv preprint*, 2016.

[16] T. Hua, W. Wang, Z. Xue, S. Ren, Y. Wang, and H. Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608, 2021.

[17] A. Hyvarinen and H. Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in Neural Information Processing Systems*, 29:3765–3773, 2016.

[18] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[19] A. Hyvarinen, H. Sasaki, and R. Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019.

[20] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 4020–4031, 2018.

[21] L. Jing, P. Vincent, Y. LeCun, and Y. Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[22] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

[23] I. Khemakhem, R. P. Monti, D. P. Kingma, and A. Hyvärinen. Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. *arXiv preprint arXiv:2002.11537*, 2020.

[24] H. Kim and A. Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[26] D. Klindt, L. Schott, Y. Sharma, I. Ustyuzhaninov, W. Brendel, M. Bethge, and D. Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

[27] A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

[28] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[29] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:II–104 Vol.2, 2004.

[30] J. D. Lee, Q. Lei, N. Saunshi, and J. Zhuo. Predicting what you already know helps: Provable self-supervised learning. *arXiv preprint arXiv:2008.01064*, 2020.

[31] Z. Lin, K. Thekumparampil, G. Fanti, and S. Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *International Conference on Machine Learning*, pages 6127–6139. PMLR, 2020.

[32] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[33] Z. Liu, P. Luo, X. Wang, and X. Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

[34] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*, 2019.

[35] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020.

[36] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[37] J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[38] S. Purushwalkam and A. Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *arXiv preprint arXiv:2007.13916*, 2020.

[39] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. Deep visual analogy-making. *Advances in neural information processing systems*, 28:1252–1260, 2015.

[40] P. H. Richemond, J.-B. Grill, F. Altché, C. Tallec, F. Strub, A. Brock, S. Smith, S. De, R. Pascanu, B. Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020.

[41] K. Ridgeway and M. C. Mozer. Learning deep disentangled embeddings with the f-statistic loss. *arXiv preprint arXiv:1802.05312*, 2018.

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

[43] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020.

[44] Y. Tian, L. Yu, X. Chen, and S. Ganguli. Understanding self-supervised learning with dual deep networks. *arXiv preprint arXiv:2010.00578*, 2020.

[45] C. Tosh, A. Krishnamurthy, and D. Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pages 1179–1206. PMLR, 2021.

[46] Y.-H. H. Tsai, Y. Wu, R. Salakhutdinov, and L.-P. Morency. Demystifying self-supervised learning: An information-theoretical framework. *arXiv e-prints*, pages arXiv–2006, 2020.

[47] M. Tschannen, O. Bachem, and M. Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[48] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6924–6932, 2017.

[49] S. Van Steenkiste, F. Locatello, J. Schmidhuber, and O. Bachem. Are disentangled representations helpful for abstract visual reasoning? *Advances in Neural Information Processing Systems*, 32, 2019.

[50] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[51] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[52] N. Zhao, Z. Wu, R. W. Lau, and S. Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020.

[53] R. S. Zimmermann, Y. Sharma, S. Schneider, M. Bethge, and W. Brendel. Contrastive learning inverts the data generating process. *arXiv:2102.08850*, 2021.

A Reproducibility

In this section, we provide the information required to reproduce our results reported in the main text. And we commit to making the code implementation and evaluating checkpoints public. Our experiments are run on a machine with an AMD Ryzen Threadripper 3970X 32-Core Processor and a GeForce RTX 3090 GPU.

VAE methods implementation. For the results on synthetic datasets, i.e., dSprites, Cars3D, SmallNORB, and Shapes3D, the disentanglement score is from the original logs of DisLib (34) [2]. In the released logs, each method has different training configurations, and our reported result is from the configuration with the highest average performance overall the provided random seeds. For the evaluation on the CelebA dataset, we follow an open-sourced implementation in Pytorch [3] and align the encoder architecture of all methods to be the same as described in Appendix A.1. For the results on Shapes3D, because DisLib does not release the pretrained checkpoints, we use the same open-sourced implementation to reproduce with the configuration indicated by DisLib. Parameters are kept as the default well-tuned version in the provided implementation. When the latent dimension is 1000, training of BetaTC VAE will collapse with the default hyperparameters, we have to decrease the β to 3.0 to work it around.

GAN methods implementation. Limited by the text length, we do not include the performance of GAN methods in the main text, but we will report some in the following appendix content. It is hard to include GAN methods' performance in the benchmark as the training is not always stable and the discriminator weights are usually not provided in many public codebases. When evaluating the methods on synthetic datasets, the FactorVAE scores of InforGAN, IB-GAN, and InfoGAN-CR are provided in the paper of Lin et al.. But the evaluation of other metrics in Lin et al. uses a not aligned settings with Locatello et al., so we check its officially release [4] to evaluate the provided implementation and model weights under the unified evaluation setup. We perform the same evaluation process for results on the CelebA dataset.

Energy-based Model (EBM). We refer to the implementation of ICE-BeeM (23) for this method. We use the officially released codebase for it [5]. The encoder implementation has been aligned with our default already. The only modification we make is to use the unconditional version instead of its default conditional version in loss computation to satisfy the fully unsupervised settings.

Contrastive Learning implementation. Our implementations are based on the public and official implementations of MoCo/MoCov2 [6], BYOL/SimSiam [7] and Barlow Twins [8]. The details of implementation are explained in Appendix A.1.

Evaluation Protocol. For MED, we first compute MI following the implementation of MIG by DisLib (34). Then we calculate the entropy disentanglement score in the same way as the DCI Disentanglement score in DisLib. For other disentanglement metrics evaluation, we use the implementation of DisLib. The settings of some important parameters are provided in Appendix A.2.

A.1 Implementation of contrastive learning model

Architecture. To make a fair comparison with previous methods, we follow the encoder architecture in Factor VAE (24). The pipeline details are shown in Table 2. After each convolutional layer in the figure, there is a ReLU activation layer and a group normalization (group number = 4) layer for BYOL. So, the encoder is a stack of (Conv-ReLU-GN) blocks. For other contrastive learning methods, we keep the default batch normalization to replace GN. By default, the final output channel number is 1000, i.e, $D = 1000$. For other details of contrastive learning methods, we follow the convention in their official implementations.

[2] https://github.com/google-research/disentanglement_lib
[3] https://github.com/AntixK/PyTorch-VAE
[4] https://github.com/fjxmlzn/InfoGAN-CR
[5] https://github.com/ilkhem/icebeem
[6] https://github.com/facebookresearch/moco
[7] https://github.com/lucidrains/byol-pytorch
[8] https://github.com/facebookresearch/barlowtwins

Besides the representation network (encoder), BYOL also has a projector network and a predictor network. Both of them consist of a pipeline "Linear $\longrightarrow$ BN $\longrightarrow$ ReLU $\longrightarrow$ Linear". The projection dimension is 256, and the hidden dimension of the projector is 4096. The predictor keeps a 256-dimensional feature vector in its pipeline.

Table 2: The encoder architecture for our implemented contrastive learning methods on synthetic datasets. Besides, there is a ReLU activation layer and a possible normalization layer following each convolutional layer to create a stack of (Conv-ReLU-Norm) blocks.

Encoder
input: 64×64 images
pipeline:
4×4 conv, stride 2, 32-channel
4×4 conv, stride 2, 32-channel
4×4 conv, stride 2, 64-channel
4×4 conv, stride 2, 64-channel
4×4 conv, stride 2, 128-channel
1×1 conv, stride 1, D-channel

Training settings. We make minor modifications to the training setting of default BYOL to apply to contrastive learning methods without negative samples. For training on all datasets, the images are resized to 64x64. For data preprocessing, we copy 1-channel images of dSprites and SmallNORB to be 3-channel. During the training stage, we use such a pipeline of augmentation (in *PyTorch*-style):

1. *RandomApply(transforms.ColorJitter(0.8, 0.8, 0.8, 0.2), p=0.3)*
2. *RandomHorizontalFlip()*
3. *RandomApply(transforms.GaussianBlur((3,3), (1.0, 2.0)), p=0.2)*
4. *RandomResizeCrop(size=(64, 64), scale=(0.6,1.0))*
5. normalization.

For the normalization, the pixel value of images from dSprites and SmallNORB is uniformly normalized from [0,255] to [0,1.0]. For Cars3D, Shapes3D, and CelebA, we adopt the commonly used Imagenet-statistic normalization for preprocessing the RGB image pixel values.

During training, we use Adam optimizer by default, whose learning rate is $3e-4$ without weight decay. The batch size is set to 512 by default. For evaluation on dSprites, Shapes3D, and CelebA, we select the weights after training for 15 epochs for evaluation. We select the weights after training for 140 epochs for evaluation on Cars3D and the weights of the 200th epoch on SmallNORB considering the small scale of these two datasets.

To decrease the influence of randomness, we train each model configuration multiple times with different random seeds (seed=0, 1, 2). We report the average and standard deviation. To be precise, as our implementation is based on Pytorch, we initialize the libraries of *numpy*, *torch*, *torch.cuda*, and *random* with the same random seeds.

Table 3: The factors on all the datasets we investigate the disentanglement on.

	dSprites	Shapes3D	Cars3D	SmallNORB	CelebA
Factors (# of values)	Shape (3) Scale (6) Orientation (40) Position X (32) Position Y (32)	Floor hue (10) Wall hue (10) Object hue (10) Scale (8) Orientation (15) Shape (4)	Elevation (4) Azimuth (24) Object id (183)	category (10) Elevation (9) Azimuth (18) Lighting (6)	40 attributes (2 for each)

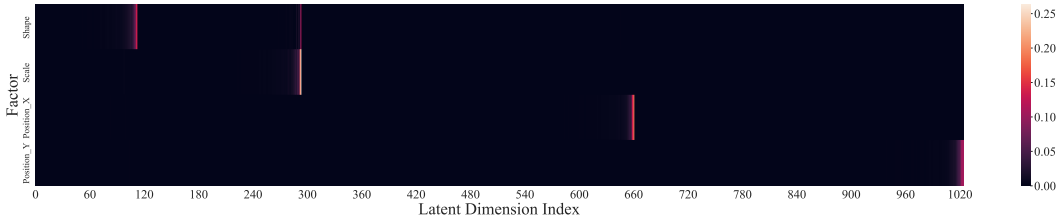


Figure 7: The importance distribution for the representation learned from BYOL on dSprites. Here, we follow the practice of DisLib to use a Gradient Boosting Tree (GBT) regressor to determine the importance matrix of each latent dimension in predicting each factor. Compared with the Mutual Information distribution shown in Figure 1a, the importance distribution is significantly more sparse. Sparsity is encouraged when constructing the GBT regressor. This makes it hard to study the true representation pattern.

A.2 Evaluation Metrics

In the main text, we compare the evaluation metrics provided in the DisLib protocol with our proposed MED metric. Here we provide more details about them. Moreover, we would conduct evaluations under all of them in the next section.

BetaVAE Metrics. Introduced in Higgins et al. (15), BetaVAE score assumes each dimension corresponds to one category in a linear classifier. Representations are obtained after the generated samples with only one factor fixed. Then we calculate the summation of the divergence between different representations and put it into a linear classifier. The classifier is trained to predict the index k for the fixed data factor. The accuracy of this linear model is the value of the BetaVAE score.

FactorVAE Metrics. Kim and Mnih (24) argues the BetaVAE score has the tendency to fail into a spurious disentanglement and proposes a new metric based on a majority vote classifier. Representations are obtained after the generated samples with only the k-th factor fixed. Normalizing each dimension in representations in terms of standard deviation. The index of dimension with the lowest variances of normalized representation and the factor index k are the input and the output of the linear classifier. The accuracy of the classification is the FactorVAE score.

Mutual Information Gap. Chen et al. (6) assumes the disentanglement model has the property that most information of one specific factor is contained in one dimension or a group of certain dimensions. The mutual information gap is the summation of the difference between the highest and second-highest normalized mutual information between a fixed factor and the dimensions of the output representation vector. The formula can be illustrated below:

$$\frac{1}{K}\sum_{k=1}^{K}\frac{1}{H_{z_k}}(I(v_{j_k}, z_k) - \max_{j \neq j_k} I(v_j, z_k)), \tag{5}$$

where K is the overall number of ground truth factors. v is the latent representation and z_k is the factors of latent variables and $j_k = \arg\max_j I(v_j, z_k)$.

DCI disentanglement. As Eastwood and Williams (9) suggest, the disentanglement can also be measured by the entropy of relative importance for each dimension in predicting factors. First, we have to know the importance of each dimension of the representation for predicting each factor. The importance is determined by a regressing model such as Lasso or Random Forest in the original DCI implementation (9) or Gradient Boosting Tree in DisLib implementation (34). We note the importance matrix R where R_{ij} is the importance of the i-th dimension in prediction the j-th factor. Then the disentanglement score for the i-th dimension is defined as $D_i = (1 - H_K(P_i))$ where $H_K(P_i) = -\sum_{k=0}^{K-1} P_{ik} log_K P_{ik}$ denotes the entropy and $P_{ij} = R_{ij}/\sum_{k=0}^{K-1} R_{ik}$ denotes the normalized importance of the i-th dimension in prediction the j-th factor. Finally the overall disentanglement score is calculated as $D = \sum_i \rho_i D_i$ where $\rho_i = \sum_j R_{ij}/\sum_{ij} R_{ij}$ is the weighting factor of the each dimension's informativeness in representing factors.

SAP. Kumar et al. (27) proposes the Separated Attribute Predictability (SAP) score. SAP is computed with classification score of predicting j^{th} factors on i^{th} dimension as the ij^{th} entry. SAP is the mean of the difference between the highest and second-highest scores for each column.

We follow the implementation provided by DisLib (34) for the evaluation protocol. Despite exceptions, the evaluation batch size is 64, the *prune_dims.threshold* is 0.06. If a classifier is required to be trained

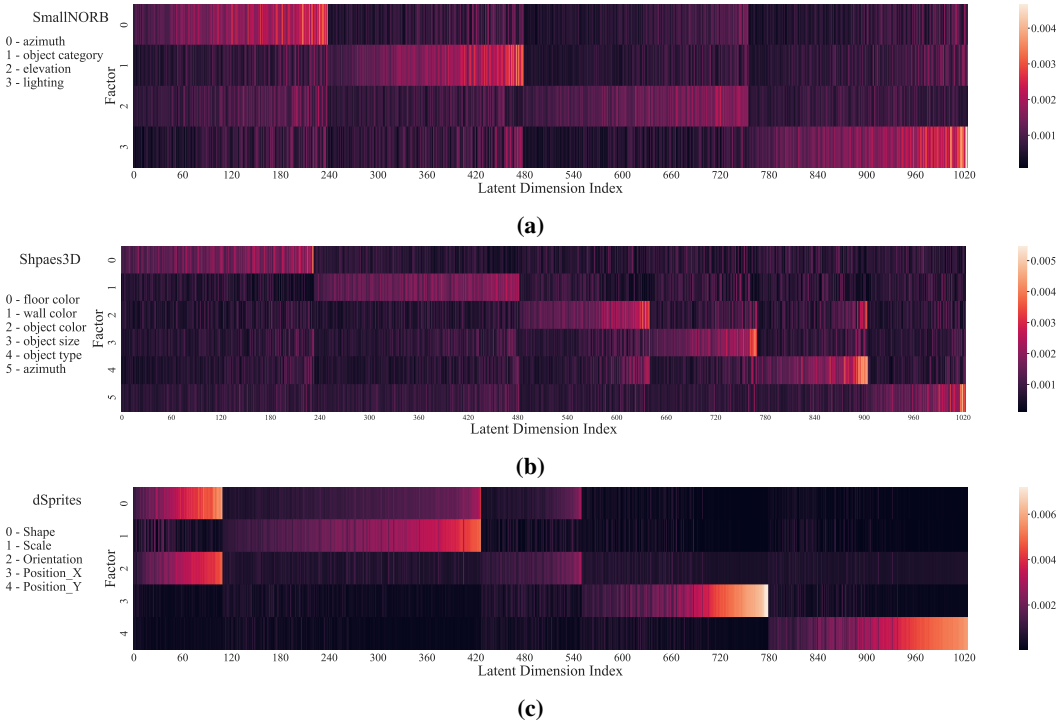


Figure 8: The mutual information distribution on SmallNORB(a), Shapes3D(b) , and dSprites including all factors(c).

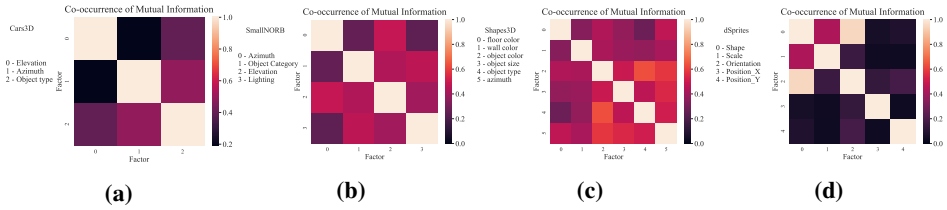


Figure 9: The co-occurrence of factors in the mutual information relationship among BYOL representations on Cars3D(a), SmallNORB(b) , Shapes3D(c), and dSprites including all factors(d).

during evaluation, *num_train* is 10000, and *num_eval* is 5000. For Mutual information computation, the discretizer function is the histogram discretizer, and the number of bins in the discretization is 20. For the evaluation of MIG and SAP on dSprites, SmallNORB, Cars3D, and Shapes3D, BYOL representation vectors are reduced to 10 dimensions by PCA to be aligned with other methods. For the evaluation of MIG and SAP on CelebA, to have a fair comparison, the representation vectors of all methods are reduced to 40 dimensions. For the implementation of our proposed MED, the basic logic is the same as DCI Disentanglement, but we replace the classifier output with the mutual information based scores.

B More Qualitative Study

We provide more qualitative studies about the disentanglement property shown by the contrastive learning here. We still use BYOL as an example of the negative-free contrastive learning methods.

B.1 Importance Distribution by DCI

In the main text, we concisely talked about the potential variables introduced by the learnable model under some metrics. Here we show an example for the widely used DCI Disentanglement metric. We follow DisLib to use Gradient Boosting Tree to estimate the importance matrix between each factor and each latent dimension. All parameters are set the same as its default protocol. The visualization is shown in Figure 7. Compared with the mutual information distribution shown in

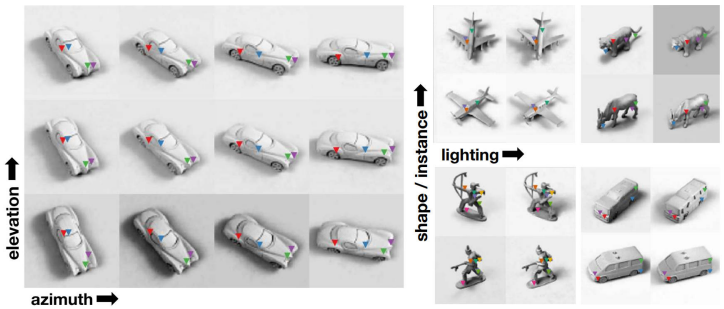


Figure 10: Some samples from SmallNORB dataset. The variance is controlled by the factor indicated on the axis. The image is from Jakab et al. (20).

Figure 1a, the importance distribution is much more sparse. This is because the construction of the GBT regressor encourage the sparsity of the output importance matrix. Such a sparsity can lead to the misunderstanding that the correlation between factors and latent dimensions is also sparse which is not true. By using pure measurement without involving additional adaptive models, such a problem will not be raised in the proposed MED metric.

B.2 Mutual Information Heatmaps

We compute MI between each latent dimension and each data factor and visualize them by heatmaps. The heatmaps offer us an intuitive picture of the learned representation space. For completeness, we show the MI heatmaps on SmallNORB, Shapes3D, and dSprites with all factors in Figure 8a, Figure 8b and Figure 8c respectively. We can see that the disentangled pattern described in the main text still emerges. There is a group of columns brighter than others in each row, and these groups do not overlap for most rows. However, we find that some latent dimensions emphasize more than one factor. We provide a more detailed analysis from the perspective of factor co-occurrence on this phenomenon in B.3 below.

B.3 Co-occurrence of Factors

To understand to what extent one dimension of the learned representation would respond to more than one factor, we make the co-occurrence of mutual information to factors on more datasets here. The visualizations are shown in Figure 9 for the results on Cars3D, SmallNORB, Shapes3D, and dSprites respectively. Moreover, we now analyze the definition of data factors of these datasets to discuss whether they are defined to be fully independent or not.

SmallNORB Though most non-diagonal entries have very low co-occurrence of mutual information, two pairs of factors show slightly higher co-occurrence. They are "azimuth-elevation" and "instance category-lighting". After investigating the dataset, we find the two pairs of factors are not fully independent. Figure 10 show some samples with corresponding factors manipulated. We could see that the elevation and azimuth are not fully independent. And the correlation between the instance category and the lighting factor is even more obvious because the lighting condition is sensibly related to the shadow around the object, whose distribution and shape are highly determined by the instance category.

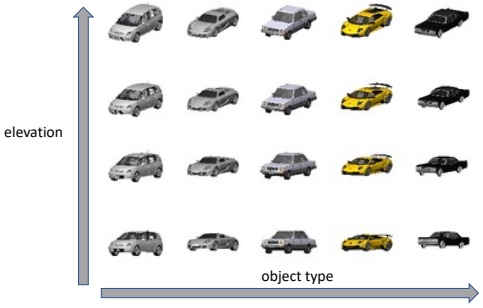


Figure 11: Samples from Cars3D. The variables of the object type and the elevation are controlled. It shows that the two factors are not independent.

Cars3D Only one pair of factors show some co-occurrence, i.e. "elevation-object type". We randomly selected samples from Cars3D by different object types and elevations, as shown in Figure 11. It shows that with the same value of

elevation, samples of different object types have different visual elevations. So these two factors are not fully independent. This explains the slightly higher co-occurrence of mutual information between this pair of factors.

Shapes3D The result shows relatively bad disentanglement. To be precise, some factor pairs show low mutual information co-occurrence as expected, such as the color factors of floor, wall, and object and the pair of "object color - azimuth". But the MI co-occurrence of "wall color - object size" and "object color - object size;" are higher than we expected as we did not recognize their high dependence. This result might relate to our model's relatively poor performance on Shapes3D.

dSprites We omit the factor *orientation* in the visualization at main text for clarity. Here we show the results including all factors. For dSprites dataset, the value of *orientation* is highly correlated with *shape*. This is because that different *shape* has different symmetry properties for rotation. As a result, dimensions that encode information of *shape* tend to capture information of *orientation* simultaneously. This explains that, in Figure 8c, some dimensions have two brighter entries at "shape" and "orientation" rows. Thus, it is not surprising that the "shape - orientation" cell is brighter than other off-diagonal elements in Figure 9d.

B.4 Manipulating Factors

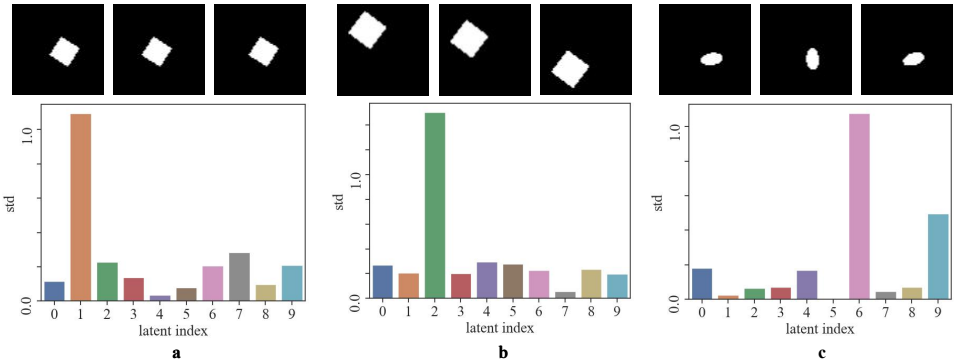


Figure 12: Representation variation when manipulating one factor only in the dimension-reduced version. In (a) and (b), *position_x* and *position_y* are manipulated respectively and only cause one dimension significantly variate. While, in (c), when manipulating the ill-defined factor *orientation*, two dimensions variate.

In the main paper, we studied the influence on the generated representation by manipulating the factors, where the representation is reduced by selecting dimensions as in calculating Top-k MED. Here, we do the qualitative study of the influence on representation by manipulating factors in another way but still on dSprites. To make the original high-dimensional representation space more compact, we use the unsupervised dimension reduction by PCA instead, which is more general when the data factor pattern is unknown. Here, we reduce the representation dimension by PCA to 10. Note that since the PCA operation mixes the original latent space with a linear combination, it might destroy the existing disentanglement property in the high dimensional space, or enhance the disentanglement if the original high dimensional space is a linear combination of the ground truth factors. But such influence is usually considered secondary to the disentanglement learned by a model. No matter which case, if the dimension-reduced representation shows disentangled properties, the original space at least captures linearly transformed ground truth factors, and the dimension reduction techniques such as PCA can make the representation more compact in a qualitative study.

Figure 12 shows the result of representation vector variation when changing only one factor at once. Given three images with only one factor's value being different, we generate the 10-dim representation vectors from them. Then, we compute the variance across the three vectors, leading to 10 scalars. The larger the variance is, the more that dimension responds to the factor change. Figure 12(a) and (b) show how reduced representation vector changes when manipulating *position_x* and *position_y* factor respectively. It shows good disentanglement that only one representation dimension has high variation. However, in Figure 12(c) we show a failure mode of the ill-defined factor *orientation* that change of factor causes both the 6th and the 9th dimensions of reduced representation to have large

variations. From the results, we observe that manipulating one well-defined independent factor causes evident variance in only one dimension. And it shows that we could make the learned representation vector more compact by unsupervised dimension reduction.

C More Quantitative Results

C.1 Full Disentanglement Benchmark

In the main text, we evaluate the disentanglement under our proposed MED metric for full space and top-k subspace on multiple datasets. In this section, to provide a more complete understanding of the disentanglement property of contrastive learning without negatives, we report the disentanglement scores with other metrics, including FactorVAE score, BetaVAE score, MIG, SAP, and DCI Disentanglement.

For the results of VAE-based methods, as the large-scale benchmark of Locatello et al. (34) provides the original logs on dSprites, Cars3D, and SmallNORB datasets, we simply report the performance of the best configuration. The original logs on Shapes3D are not available, so we train and evaluate on Shapes3D by ourselves for the MED scores. For scores under other metrics, we report the median disentanglement scores. Some results are from Locatello et al. (35) but the std error is not available. The median performance of SlowVAE is from its original paper (26). For the results of CelebA, the result of InfoGAN-CR is from its officially released checkpoint without availability to the std error. For other methods, we report the mean value of our trained weights over three random seeds as default. Because the evaluation of DCI is extremely time-consuming, around 14 hours for a 1000-d model, we only take BYOL as an example here for negative-free contrastive learning methods. All results are combined and shown in Table 4. We also include traditional metrics results on their selected top-k subspace for contrastive learning methods.

Aligned with the analysis we provide in the main text, the results show significant disagreement among the existing metrics. To be precise, for those metrics (BetaVAE score, FactorVAE score, DCI Disentanglement) using a learnable model such as a regressor or a classifier, the high-dimensional BYOL model achieves a significant advantage. However, for the metrics relying on only one or two dimensions to reveal the connection between a latent dimension and a factor (MIG and SAP), BYOL's performance is not that impressive anymore.

Finally, the result on CelebA shows the great robustness of BYOL's learned representations to be disentangled on real-world datasets. Yet, the large gap between the score of CelebA and those on synthetic datasets emphasizes the difficulty of learning disentangled factors on real-world images. It is hard to empirically study whether it is the high dimension that gives BYOL advantages on some metrics because the nature of BYOL makes it hard to be trained with a small latent dimension to make a comparison.

In the last rows for each dataset, we show previous disentanglement metrics scores on the top-k subspaces of BYOL. We find that BYOL (top-k) has performed better than or on par with unsupervised methods with low latent dimensions. Note that Ada-GVAE, SlowVAE, and EBM are (weakly) supervised methods. The SOTA performance justifies our claim of the well-disentangled subspaces. MIG and SAP require a large gap between the two most vital representation dimension-factor responses, measured by mutual information and classification accuracy. But out of 1000 dimensions, we choose the k most relative ones. Hence, the gap ought to be minimal. Consequently, BYOL (top-k) receives lower MIG and SAP scores. Except for dSprites and CelebA, BYOL (top-k) have lower DCI scores. The primary cause of this is the massive information loss during the selection of the top-k dimensions, which completely obliterates the original encoding pattern. As a result, the gradient boosting tree (GBT) developed throughout the DCI process tends to "borrow" information from latent dimensions that emphasize additional relative factors to classify some complex factors. The "information-lending" dimensions then become less disentangled since they are also important for predicting those complex factors. Taking the Cars3D dataset as an example, we set $k = 3$ for top-k MED. To predict the factor *object type*, the GBT needs to classify 183 types of cars from only 9 dimensions with only 3 emphasizing *object type*. This forces GBT to utilize information from *elevation* since these two attributes are correlated (see Appendix B.3). However, we keep all they have learned from training for other low-dimensional methods. Therefore, the information loss problem does not apply to them.

Table 4: Evaluation results on multiple datasets with different disentanglement metrics. The best results of all models are **bolded**. The best results on low-dimensional space are underlined.

	Model	BetaVAE	FactorVAE	MIG	SAP	DCI	MED
dSprites	β-VAE	82.3 (7.6)	65.8 (9.2)	26.3 (11.0)	5.2 (2.7)	39.3 (13.2)	32.6 (10.0)
	β-TCVAE	86.7 (2.4)	76.6 (7.8)	23.8 (6.8)	6.9 (0.9)	36.3 (7.1)	31.8 (7.4)
	FactorVAE	84.9 (2.8)	75.3 (7.4)	18.4 (9.0)	6.8 (0.8)	28.8 (10.6)	32.5 (10.1)
	DIP-VAE-I	82.7 (3.3)	59.1 (4.8)	9.6 (5.1)	5.2 (2.6)	14.4 (4.6)	18.8 (5.6)
	DIP-VAE-II	81.5 (4.9)	58.6 (7.6)	7.4 (3.4)	3.6 (2.2)	12.3 (5.2)	14.7 (5.5)
	AnnealedVAE	86.5 (0.1)	60.1 (0.0)	**35.2 (1.3)**	7.6 (0.5)	37.9 (2.1)	35.8 (0.8)
	Ada-GVAE	88.0 (2.7)	73.1 (3.9)	17.3 (4.7)	6.6 (2.0)	32.3 (4.6)	–
	SlowVAE	87.0 (5.1)	75.2 (11.1)	28.3 (11.5)	4.4 (2.0)	47.7 (8.5)	–
	EBM	82.3 (2.0)	65.7 (12.5)	1.7 (0.5)	3.0 (1.2)	19.1 (1.8)	6.8 (4.0)
	InfoGAN-CR	85.5 (1.0)	88.0 (1.0)	19.8 (3.2)	6.0 (1.0)	14.0 (5.2)	–
	BYOL	**93.2 (0.4)**	**91.6 (0.8)**	29.3 (0.4)	**8.0 (0.4)**	**66.9 (0.2)**	31.3 (0.4)
	BYOL (top-2)	90.0 (0.9)	79.2 (2.3)	3.3 (0.9)	0.8 (0.2)	45.0(0.1)	**53.7 (0.7)**
Cars3D	β-VAE	**100.0 (0.0)**	89.3 (1.2)	11.7 (1.1)	1.4 (0.9)	38.7 (4.6)	29.0 (2.2)
	β-TCVAE	**100.0 (0.0)**	92.2 (2.7)	**15.5 (2.9)**	1.7 (0.3)	42.7 (3.5)	**33.0 (3.8)**
	FactorVAE	**100.0 (0.0)**	91.7 (4.1)	10.6 (2.2)	**2.0 (0.5)**	29.0 (6.7)	29.1 (3.0)
	DIP-VAE-I	**100.0 (0.0)**	90.5 (5.0)	5.9 (2.8)	1.9 (1.4)	22.6 (5.6)	19.4 (3.3)
	DIP-VAE-II	**100.0 (0.0)**	85.0 (6.1)	5.1 (2.7)	1.3 (0.8)	20.8 (5.4)	16.7 (4.1)
	AnnealedVAE	**100.0 (0.0)**	85.0 (4.3)	7.6 (1.0)	1.5 (0.5)	18.5 (4.3)	15.5 (2.5)
	SlowVAE	**100.0 (0.0)**	90.4 (0.5)	15.4 (2.2)	1.6 (0.5)	48.0 (2.4)	–
	BYOL	**100.0 (0.0)**	**95.8 (1.2)**	7.6 (0.9)	1.8 (0.7)	**48.5 (2.3)**	9.7 (0.5)
	BYOL (top-3)	**100.0 (0.0)**	95.2 (0.8)	3.8 (0.5)	1.1 (0.8)	15.8 (3.6)	31.8 (1.3)
SmallNORB	β-VAE	84.1 (2.7)	60.1 (2.4)	25.0 (1.1)	11.4 (1.1)	32.6 (0.6)	24.4 (0.7)
	β-TCVAE	84.5 (2.7)	60.3 (2.3)	25.4 (0.9)	11.7 (1.1)	35.2 (0.7)	25.0 (0.9)
	FactorVAE	80.8 (3.8)	62.5 (3.6)	23.9 (2.0)	10.2 (0.9)	33.4 (1.1)	**25.9 (1.2)**
	DIP-VAE-I	84.2 (3.2)	69.8 (4.6)	24.3 (2.7)	10.2 (1.4)	30.0 (2.1)	24.5 (2.1)
	DIP-VAE-II	85.2 (1.3)	58.4 (2.1)	**25.5 (1.5)**	**14.4 (0.4)**	32.3 (0.7)	24.4 (0.7)
	AnnealedVAE	60.8 (6.2)	50.0 (9.9)	9.1 (2.2)	6.8 (0.8)	15.7 (6.4)	5.5 (3.7)
	SlowVAE	78.2 (3.8)	47.0 (2.9)	23.8 (1.8)	7.8 (1.1)	28.7 (0.7)	21.8 (1.3)
	EBM	79.0 (4.4)	57.9 (3.5)	1.7 (0.5)	1.9 (0.1)	13.9 (2.2)	2.3 (1.7)
	BYOL	**97.0 (0.8)**	**81.0 (0.5)**	3.3 (0.9)	2.2 (0.3)	**51.0 (1.0)**	7.7 (0.2)
	BYOL (top-2)	86.7 (0.4)	65.6 (3.7)	3.3 (1.4)	1.5 (0.2)	13.6 (0.3)	25.7 (0.3)
Shapes3D	β-VAE	100.0 (0.0)	92.4 (4.5)	37.8 (16.0)	11.3 (3.2)	77.3 (3.2)	52.4 (9.4)
	β-TCVAE	100.0 (0.0)	90.5 (5.5)	46.4 (15.4)	12.4 (6.1)	78.4 (5.2)	53.2 (4.9)
	FactorVAE	98.1 (3.2)	90.6 (6.4)	48.2 (15.2)	11.1 (4.3)	71.8 (8.6)	55.9 (8.0)
	DIP-VAE-I	98.3 (3.3)	84.2 (11.3)	20.1 (8.4)	6.0 (1.1)	69.0 (3.6)	43.5 (3.7)
	DIP-VAE-II	99.6 (0.04)	94.9 (4.1)	22.1 (3.54)	8.2 (1.8)	49.8 (10.6)	52.6 (5.2)
	AnneledVAE	95.1 (4.4)	88.8 (4.6)	46.2(5.4)	8.5(1.6)	56.2(4.7)	**56.1 (1.5)**
	Ada-ML-VAE	**100.0**	**100.0**	50.9	12.7	94.0	–
	Ada-GVAE	**100.0**	**100.0**	56.2	**15.3**	**94.6**	–
	SlowVAE	**100.0 (0.1)**	97.3 (4.0)	**64.4 (8.4)**	5.8 (0.9)	82.6 (4.4)	–
	EBM	75.9 (11.2)	53.2 (8.7)	5.2 (2.2)	2.8 (1.1)	21.8 (11.0)	2.1 (2.6)
	BYOL	91.5 (3.9)	82.5 (2.4)	5.2 (1.7)	2.8 (0.3)	53.1 (1.5)	6.0 (0.5)
	BYOL (top-2)	95.5 (1.1)	83.0 (3.6)	2.8 (0.8)	1.4 (0.5)	27.2 (3.8)	19.7 (1.3)
CelebA	VAE	21.5 (3.2)	6.1 (3.8)	0.8 (0.1)	0.9 (0.2)	11.2 (2.3)	3.8 (0.2)
	β-VAE	19.1 (1.9)	5.8 (1.8)	0.1 (0.1)	0.6 (0.2)	8.7 (1.9)	3.3 (0.1)
	β-TCVAE	19.9 (2.3)	9.8 (2.4)	0.6 (0.2)	1.2 (0.3)	3.5 (1.1)	4.7 (0.1)
	FactorVAE	25.3 (3.0)	**12.0 (2.1)**	0.4 (0.1)	0.6 (0.2)	7.1 (0.7)	0.6 (0.6)
	DIP-VAE-I	21.0 (1.9)	9.3 (1.1)	0.2 (0.1)	0.9 (0.3)	13.8 (2.2)	3.6 (0.2)
	InfoGAN-CR	16.8	11.3	1.6	2.8	22.0	–
	BYOL	**35.7 (2.1)**	11.5 (1.1)	**2.6 (0.7)**	**8.2 (0.9)**	**41.0 (1.3)**	4.8 (0.4)
	BYOL (top-3)	25.8 (0.8)	9.7 (0.5)	0.7 (0.2)	0.2(0.02)	15.8(0.3)	**6.8 (0.7)**

D Ablation Study about the Normalization Choice

In this section, we put ablation studies here about the choice of normalization layers in the model architecture.

Table 5: Results of using different normalization strategies on dSprites.

normalization	w/o norm	BN	GN	LN	IN
MED	23.8 (0.6)	29.4 (0.5)	31.3 (0.4)	31.3 (0.8)	0.0 (0.0)

We experiment with five types of normalization layers in the encoder network on the dSprites dataset. The results are shown in Table 5. For the group normalization, we set the group number to 4. On the dSprites dataset, we find the commonly used BN decreases the disentanglement performance. By keeping the batch norm in the projector and the predictor, removing the batch norm in the encoder will not cause the model to collapse, which agrees with the observation in previous works (40). On the contrary, replacing batch norm in the encoder with group norm or layer norm will increase the representation disentanglement while achieving similar accuracy in downstream factor prediction. We notice that a similar phenomenon has been discovered before in supervised representation disentanglement. For example, Bau et al. (2) discovered that a network trained with batch normalization layers has less interpretable (disentangled) neurons. On the other hand, the instance norm (48) completely breaks the contrastive learning process. We still do not fully understand this behavior, but we hypothesize that it may be caused by the shared batch statistics that make it hard for a feature to be aligned with the ground truth factor.

E Disagreement of Existing Metrics

Without a uniformly recognized definition of "disentanglement", the validity of existing metrics is backed up with their shared expectation of "disentanglement" and consistency of experimental results, as suggested in DisLib (34). However, the results in Figure 4 break the belief. The disagreement arises when the high-dimensional representation model joins the comparison. We provide an in-depth view of the disagreement by ranking different methods. The results are shown in Figure 13. If all metrics agree perfectly, there should be no relative ranking switch. However, the frequent switch, especially with BYOL being taken into comparison, strongly suggests their disagreement. Results in Table 4 further show that different metrics may make completely different rankings with large value gaps. We also provide logistic analysis in Appendix F and quantitative results in Appendix C. Given the fact that self-supervised representation learning always requires a high dimension to train, MED is necessary to extend the study of disentangled representation learning to complicated real-world datasets and self-supervised representation learning methods.

F The Superiority of MED

Here, we provide both experimental observations and theoretical analysis in synthetic scenarios to show the superiority of MED: in what cases other metrics outputs meaningless or even opposite results but MED can still perform a meaningful evaluation.

F.1 Experimental Observations

As we discuss throughout the paper, the five existing metrics we investigate, i.e. BetaVAE score, FactorVAE score, SAP, MIG, and DCI Disentanglement, are unfair to models of different dimensions. So we extend the results mentioned in Section 3.2 to show the superiority of MED by confirming its stability under different dimensions. We conduct a sanity check on randomly initialized models. For a good disentanglement metric, we do not expect a high score whatever the representation dimension is. The following results reveal the malfunction of these metrics.

- **MED vs MIG/SAP**: First of all, we emphasize that SAP/MIG is designed not just for *disentanglement* but also *completeness*. They encourage a strict one-to-one mapping between latent dimensions and factors. So they tend to reward models whose dimension is close to the factor number. On the popular synthetic datasets with only a limited number of factors, low-dimensional models have an unfair advantage with MIG/SAP metrics. We compare a well-trained 1000-d BYOL model and a 10-d randomly initialized encoder in Table 6. The results show that the 10-d randomly initialized model can achieve higher scores than the

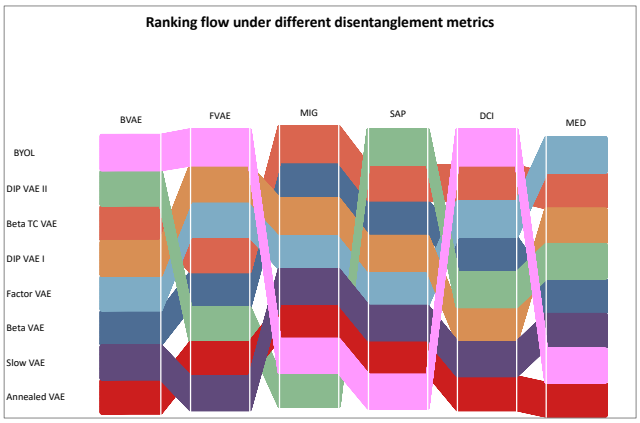


Figure 13: The flow chart of rankings of disentanglements scores. From top to down, the ranking decreases, indicating high to low disentanglement properties. If all metrics agree perfectly, there should be NO switch of relative ranking among these methods with different metrics. However, we notice obvious ranking variance, especially regarding the ranking of BYOL. With the analyses of the bias of existing metrics when the representation dimension varies, this figure emphasizes the necessity of proposing a new disentanglement metric that is fair to compare models of different dimensions.

1000-d BYOL model by SAP and MIG. But MED keeps a reasonable comparison that the trained high-dimensional model can still enjoy higher scores than low-dimensional random weights.

Table 6: The mean scores of different metrics on Shapes3D.

	MIG	SAP	MED
10-d randomly initialized	**6.7**	**3.0**	2.9
1000-d BYOL	5.2	2.8	**6.0**

- **MED vs BetaVAE/FactorVAE score**: BetaVAE and FactorVAE prefer high-dimensional models as they have more parameters to trick the learnable classifier. In Table 7, we find even randomly initialized high-dimensional (1000-d) model can achieve higher BetaVAE/FactorVAE score than a well-trained 10-d model (DIP-VAE-II). In contrast, MED provides results in line with our expectations.

Table 7: The mean scores of different metrics on dSprites.

	BetaVAE score	FactorVAE score	DCI	MED
10-d DIP-VAE-II	81.5	58.6	12.3	**14.7**
1000-d randomly initialized	**82.7**	**61.4**	**19.8**	3.8

- **MED vs DCI**: Similar to BetaVAE/FactorVAE scores, DCI also overestimate high-dimensional models as they have a higher chance to get lottery dimensions, especially with the sparsity encouraged by GBT. Also, another huge drawback of DCI is the computation efficiency: for a 1000-d model, DCI takes more than 14 hours for evaluation while MED only takes less than 20 seconds. The computation of MED is more than 2000 times faster than DCI.

Through the experimental observations, we find the abnormals from existing metrics when comparing models of different dimensions. Compared to them, the results of MED are consistent with human knowledge.

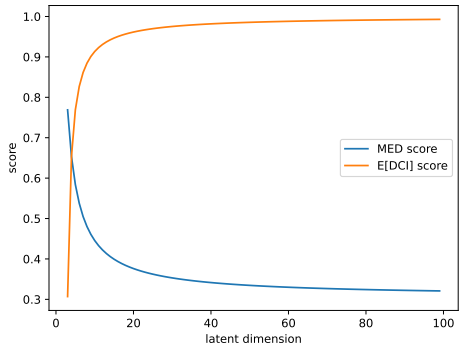


Figure 14: How MED score and DCI score change along the latent dimension of the model in our constructed linear scenarios.

F.2 Theoretical Analysis

In this part, we construct synthetic scenarios to investigate the validity of MED and existing metrics. We will show that in such situations, existing metrics will fail to reflect the real disentanglement quality but MED can show superiority to maintain reasonable evaluations. For the following cases, we assume there are two data factors F_0, F_1 on the data. We note their value as $v_0, v_1 \in \{0, 1\}$. And they are sampled independently and uniformly, i.e., $p(v_j = 1) = p(v_j = 0) = 0.5$. Then we construct linear cases to simplify the analysis that the value of latent dimensions is the combination of factor values.

F.2.1 MED vs MIG/SAP.

As mentioned, MIG and SAP require not just *disentanglement* but also *completeness*. So, for a 1000-d model whose latent code value is $c_i = v_{i\text{mod}2}, 1 \leq i \leq D, D = 1000$. I.e. half of the latent code is the true first factor, and the other half latent code is the second factor. It is clear that each dimension perfectly correlates to a single factor. But its MIG/SAP scores turn out to be 0, which is significantly opposite to our intuition. But in this case, for MED we have

$$S_i = 1, \quad \rho_i = \frac{1}{D}. \tag{6}$$

So $\text{MED}(c) = 1$, indicating that every dimension is perfectly disentangled to a single factor and fully responsive to it. Also, this conclusion applies to different values of D, showing the robustness of MED with models of different dimensions.

F.2.2 MED vs DCI.

With a constructed latent code as

$$c_i = \begin{cases} v_0 & , i = 0 \\ v_1 & , i = 1 \\ \frac{1}{2}(v_0 + v_1) & , i > 1 \end{cases}. \tag{7}$$

To simplify, we assume the importance matrix output by GBT in DCI metric is the absolute of first-order derivative, namely $|\frac{\partial c_i}{\partial v_j}|$ of the dimension i with respect to the factor F_j. GBT encourages sparsity in the output importance matrix. To reflect this, we assume the desired sparsity of the GBT is 2. Thus, we have the dimensions beyond the sparsity limit are given 0 importance. Then the importance matrix value R_{ij} is

$$R_{ij} = \begin{cases} 1 & , i = j, i \leq 1 \\ \frac{1}{2} & , i = d_j \\ 0 & , otherwise \end{cases}. \tag{8}$$

d_j is a randomly selected dimension between $[2, D-1]$. For the two factors F_0 and F_1, d_j is selected independently. So that chance that $d_0 = d_1$ is $p(d_0 = d_1) = \frac{1}{D-2}$.

(1) If $d_0 = d_1 = k$, we have

$$\rho_i = \begin{cases} \frac{1}{3} & , i = 0, 1, k \\ 0 & , otherwise \end{cases}, \qquad S_i = \begin{cases} 1 & , i = 0, 1 \\ 1 - log2 & , i = k \\ 0 & , otherwise \end{cases} \tag{9}$$

(2) If $d_0 \neq d_1$, we have

$$\rho_i = \begin{cases} \frac{1}{3} & , i = 0, 1 \\ \frac{1}{6} & , i = d_0, d_1 \\ 0 & , otherwise \end{cases}, \qquad S_i = \begin{cases} 1 & , i = 0, 1, d_0, d_1 \\ 0 & , otherwise \end{cases}. \tag{10}$$

And DCI Disentanglement score is $\text{DCI}(c) = \sum_{i=0}^{D-1} \rho_i S_i$. We have $\text{DCI}(c) = 1 - log2 = 30.7\%$ in situation (1) and $\text{DCI}(c) = 1$ in situation (2). So, in such a simplified version, the expectation of DCI Disentanglement score is

$$\mathbb{E}[\text{DCI}(c)] = \frac{1}{D-2}(1 - log2) + (1 - \frac{1}{D-2}) = 1 - \frac{log2}{D-2}. \tag{11}$$

On the other hand, similarly we can derive the MED score is

$$\text{MED}(c) = 1 - \frac{D-2}{D} log2. \tag{12}$$

which is quite consistent along the change of D. Note that the drastic difference between MED and DCI is caused by the GBT used in DCI. The GBT leads to a sparse importance matrix M, while the mutual information in MED won't. This difference doesn't make much difference in low-dimension latent codes, but DCI will be high even if the latent code doesn't disentangle at all in the high-dimensional case.

We plot the curves of MED score and the expectation of DCI score in Figure 14. It shows that when the latent dimension D increases, the expectation of the DCI score increases and approaches 1. For instance, if $D = 3$, we have $\mathbb{E}[\text{DCI}(c)] = 30.7\%$; if $D = 1000$, we have $\mathbb{E}[\text{DCI}(c)] = 99.9\%$. Such an abnormality from DCI Disentanglement alerts us that it can not give a meaningful and fair comparison between methods of different dimensions. However, MED is much more robust with the change of model dimension. For instance, if $D = 3$, we have $\text{MED}(c) = 76.9\%$ and if $D = 1000$, we have $\text{MED}(c) = 30.8\%$.

F.2.3 MED vs BetaVAE/FactorVAE score.

Situation 1 We construct a situation where the first two dimensions are a weighted combination of the factors and the value of all following dimensions are the average of the factor values. Thus we have the latent code as

$$c_i = \begin{cases} \frac{1}{3}v_0 + \frac{2}{3}v_1 & , i = 0 \\ \frac{1}{3}v_1 + \frac{2}{3}v_0 & , i = 1 \\ \frac{1}{2}(v_0 + v_1) & , i > 1 \end{cases}. \tag{13}$$

We can justify it as a very entangled representation. But the latent code value change with respect to the factor value is what BetaVAE and FactorVAE scores care about.

FactorVAE calculates the variance of latent code, i.e. $Var(c)$, when changing the factors to represent the factor-dimension response degree. During calculating the FactorVAE score, one factor is always fixed. So we have

$$Var_{\text{fix}v_0}(c_i) = \begin{cases} \frac{1}{9} & , i = 0 \\ \frac{1}{36} & , i = 1 \\ \frac{1}{16} & , i > 1 \end{cases}, \quad Var_{\text{fix}v_1}(c_i) = \begin{cases} \frac{1}{36} & , i = 0 \\ \frac{1}{9} & , i = 1 \\ \frac{1}{16} & , i > 1 \end{cases}, \tag{14}$$

so for each factor, we have a single dimension that gives the minimum variance. In this case, the FactorVAE score uses a majority vote to determine the dimension with the smallest variance with respect to the value change of a factor as the "most disentangled" dimension to it. And the vote accuracy is regarded as the FactorVAE score. So, it is clear that because of the existence of the lucky dimension $j = i$, the FactorVAE score can be expected to approach 100, indicating "perfectly disentangled"! With the increase of model dimension, the chance of having such a lucky dimension increases as well in real cases. The bias is that FactorVAE only needs one dimension which has a low response to all other factors except for one factor to get high scores.

Table 8: *Logistic Regression* accuracy results of the full representation space and top-k subspace. Methods in gray are contrastive self-supervised learning methods.The best results of all models are **bolded**. The best results on low-dimensional space are underlined.

Space	Model	dSprites	Shapes3D	Cars3D	SmallNORB	CelebA
Full Space	β-VAE	34.5 (3.0)	35.6 (2.6)	65.0 (1.6)	48.6 (0.5)	82.1 (0.1)
	β-TCVAE	32.1 (3.4)	40.3 (2.1)	68.3 (1.4)	47.4 (0.7)	**85.2 (0.1)**
	FactorVAE	30.4 (2.5)	20.7 (2.3)	56.1 (1.3)	42.6 (1.3)	80.4 (0.2)
	DIP-VAE-I	30.3 (0.9)	37.6 (1.5)	68.5 (1.2)	49.6 (0.5)	**85.2 (0.1)**
	DIP-VAE-II	30.9 (1.9)	–	67.2 (1.7)	48.5 (0.5)	–
	MoCo	46.3 (1.3)	78.3 (2.1)	61.8 (1.4)	56.3 (0.7)	**85.2 (0.3)**
	MoCov2	52.0 (0.3)	76.5 (0.3)	63.1 (1.2)	53.8 (0.5)	84.9 (0.1)
	BarlowTwins	41.7 (0.2)	74.5 (0.8)	79.0 (0.2)	**62.0 (0.1)**	83.1 (0.2)
	SimSiam	40.3 (0.1)	**80.9 (3.4)**	45.3 (4.6)	54.5 (0.5)	84.0 (0.2)
	BYOL	**54.6 (0.2)**	71.5 (1.3)	**81.2 (0.4)**	57.2 (1.1)	81.9 (0.2)
Top-k Subspace	MoCo	22.7 (1.4)	43.7 (1.1)	38.1 (1.2)	34.5 (0.2)	85.1 (0.2)
	MoCov2	26.0 (1.0)	33.4 (0.2)	32.6 (0.8)	31.4 (1.1)	84.9 (0.1)
	BarlowTwins	29.1 (0.7)	44.4 (1.3)	43.1 (1.4)	34.5 (0.4)	82.9 (0.2)
	SimSiam	38.0 (0.6)	61.8 (1.0)	29.4 (3.6)	37.7 (2.5)	83.8 (0.2)
	BYOL	37.9 (0.6)	46.3 (0.5)	43.3 (0.9)	40.2 (0.8)	81.9 (0.2)

On the other hand, the BetaVAE score calculates the latent code value change to represent the response between factors and dimensions. This is represented by the absolute value of the first-order derivative. So for the i-th dimension with respect to the j-th factor as R_{ij}, we have

$$R_{ij} = \begin{cases} \frac{1}{3} & , i = j, i \leq 1 \\ \frac{2}{3} & , i \neq j, i \leq 1 \\ \frac{1}{2} & , otherwise \end{cases} . \tag{15}$$

BetaVAE score uses a learnable linear classifier to determine the dimension. In this case, prediction j from $R_{\cdot j}$ can achieves 100% accuracy. So similarly, it also requires one "lucky" dimension to make the prediction, offering an advantage to high-dimensional models. Moreover, because of the powerful capacity of a learnable linear classifier, the BetaVAE score is close to 100 for many cases as in Table 4. It thus has another flaw of low distinguishability of methods' disentanglement properties.

On the other hand, we could now calculate the MED score for this situation. The results turns out to be MED$(c) = 1 - log2 = 30.7\%$ for any value of D. This is a relatively low score and we think it makes better sense in such entangled cases compared to the falsely high score from BetaVAE and FactorVAE scores.

Situation 2 Let us build another case as in the discussion of DCI above

$$c_i = \begin{cases} v_0 & , i = 0 \\ v_1 & , i = 1 \\ \frac{1}{2}(v_0 + v_1) & , i > 1 \end{cases} . \tag{16}$$

FactorVAE and BetaVAE scores can still give a "perfect" score over such a representation. However, when the dimension D is very large, only a small subset of the representation dimensions is disentangled while most are highly entangled. As a higher-dimensional model has a better chance to get a lucky dimension, maintaining the perfect scoring with increasing dimensions is not aligned with our intuition and expectation. From this sense, Top-K MED has its special use. When $D = 1000$, we know MED$(c) = 30.8\%$, but we can still get a subset of the 1000 dimensions to get a high Top-K MED score. Such a subset of representations is more compact and has better explainability to the factor variance.

With all the situations constructed and analyzed above, we could notice the failure of existing metrics to achieve evaluation results (1) meaningful, (2) fair, and (3) aligned with the human sense to disentanglement in some scenarios. On the other hand, our proposed MED always keeps the evaluation and comparison reasonable and aligned with the expectation from the human high-level institution.

G Properties of the Top-k Subspace

We propose the top-k selection process to study the partial disentangle properties. Here we provide more details. We first visualize the top-k selection process based on the heatmap of the importance matrix. Then we will discuss the effects of k and the downstream performance comparison of the top-k subspace and full space.

Visualization of the top-k Process We describe the top-k process in the Section 3.3. Correspondingly, here we colored the picked dimensions yellow in the Figure 15. It shows that most picked dimensions only have one entry much brighter than the others, indicating they are well-disentangled. Since *orientation* is ill-defined in dSprites, BYOL tends to encode it with *shape*. Hence the most disentangled dimensions for *shape* also capture information of *orientation*.

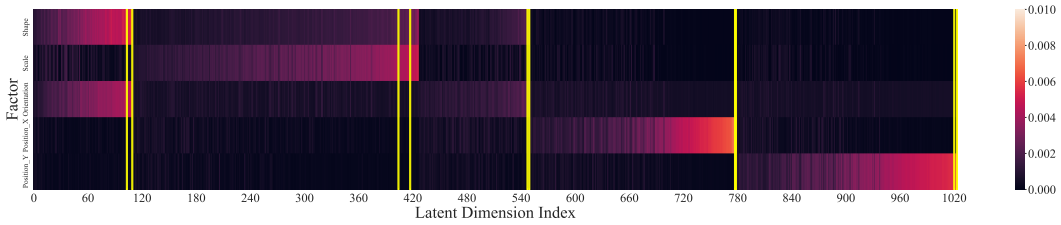


Figure 15: The visualization of the top-k process. Based on Figure 8c, the selected dimensions are highlighted with yellow boxes. For dSprites we set $k = 2$. The indices picked are 103, 109, 404, 417, 547, 549, 777, 778, 1020, and 1023.

Ablation on k The parameter k is designed as an evaluation choice. For analytical purposes, we choose the value of k to match the dimension of VAE methods in the main text. To investigate the influence of k, we set $k = 1, 2, 10, 50, 100$ and evaluate the top-k MED scores of 1000-d VAEs and BYOL on dSprites. The results are shown in Figure 16. We find that with different values of k, the ranks of top-k MED scores remain unchanged. Since the top-k process is a supervised greedy selection, as k grows the scores decrease. In conclusion, top-k MED can measure the subspace disentangle properties with a wide range of k. Therefore we can choose k according to actual needs.

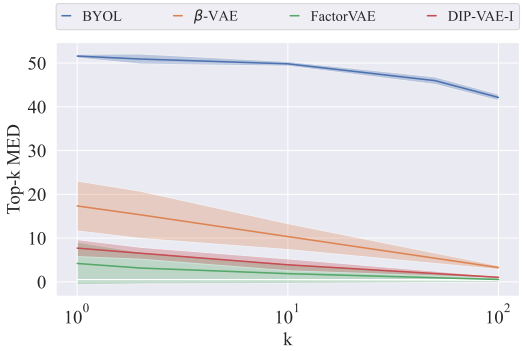


Figure 16: The influence of k in top-k MED on dSprites dataset. The evaluation results are consistent with multiple choices of k.

Downstream performance of top-k subspace We study the disentangle properties of the full space and top-k subspace in 5.3. Here we analyze the downstream performance of these spaces. We train a *Logistic Regressor* to predict each factor from the full representation or the top-k picked representation. Then we take the mean prediction accuracy of all factors as the measure of downstream performance. We compare the downstream performance of the full space of VAEs and BYOL, and top-k subspace of BYOL on multiple datasets. Here we set $k = 2$ for dSprites, Shape3D, and SmallNORB, and $k = 3$ for Cars3D and CelebA to match the dimension of VAEs. The results are in Table 8. We note that, except on CelebA, performance drops non-negligibly when completing downstream tasks with the top-k representations. This suggests that the top-k subspace is well-disentangled but other

dimensions also contain factor-related information. It is unsurprising because we cut down the dimension drastically (from 1000 to 10 on synthetic datasets and from 1000 to 100 on CelebA). However, we find that the top-k spaces learned by contrastive methods achieve better or compatible performance compared with the full spaces of VAEs on 3 out of 5 datasets (dSprites, Shapes3D, and CelebA). It is reasonable that a high-dimensional model can be better for downstream tasks as the downstream model can be trained to leverage each little piece of information from the input representations. But we get a message from the results that by drawing a subspace from the powerful CL-trained model, its performance on downstream tasks can be comparable to or superior to the low-dimensional models. And the selected low-dimensional model can have much better compactness and interpretability.

H Limitations

Our work still has some limitations. Firstly, the design of contrastive learning methods still depends on empirical practice. Therefore there lacks a widely accepted theoretical framework to analyze its disentanglement properties. As for the experiment setups, the inductive biases of different methods may potentially affect the results (such as normalization in Section D). Instead of an exhaustive search, we inherit the available best settings, i.e., the settings from DisLib (34) and the public official implementations of other methods. All hyperparameters and details have been indicated in Section A.