# CAN YOU TRUST YOUR DISENTANGLEMENT?

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

There has been growing interest, in recent years, in learning disentangled representations of data. These are representations in which distinct features, such as size or shape, are represented by distinct neurons. Measuring disentanglement, i.e., quantifying the extent to which a given representation is disentangled, is not straightforward. Multiple metrics have been proposed. In this paper, we identify two failings of existing metrics, and show how they can assign a high score to a model which is still entangled. We then propose two new metrics which redress these problems. Additionally, we introduce the task of recognizing novel combinations of familiar features (NCFF), which we argue is doable if and only if the model is disentangled. As well as being desirable in itself, NCFF provides a tangible downstream task that can help focus the field of disentanglement research, in contrast to the set of bespoke metrics that are currently used. We then show empirically that existing methods perform poorly on our proposed metrics and fail at recognizing NCFF and so, we argue, are not disentangled.

## 1 INTRODUCTION

Learning to produce an effective vector representation of input data has long been a central question for the field of deep learning. Early proponents argued that a significant advantage of neural networks was that they could form distributed representations, where each input is represented by multiple neurons, and each neuron is involved in the representation of multiple different inputs (Hinton et al., 1986). Compared to using a separate neuron for each input, distributed representations are exponentially more compact (Bengio, 2009). An extension of distributed representations is the idea of disentangled representations. These are a particular type of distributed representation in which each neuron represents a single human-interpretable factor of variation in the input data, such as colour, size or shape. These are also sometimes called "generative factors". In this paper, we mostly refer to them as "factors" or "features". Similarly, the individual scalar values in the vector representation, which are sometimes called "latent variables", we refer to as "neurons". In the strongest case, each factor is represented by a single neuron so that, e.g., changing the colour of the object in the image would cause a single neuron to change its value while all other neurons remain unchanged. (We discuss further below the ambiguity as to whether this stronger condition is required.) Disentanglement (DE) was originally formulated by Bengio (2009) (see also (Bengio et al., 2013; Bengio, 2013)). More recently, beginning with (Higgins et al., 2016), there have been many unsupervised DE methods proposed based on autoencoders.

In this paper, we make the case that existing DE methods largely fail to produce disentangled representations. We makes this case in two ways. Firstly, we examine the commonly used DE metrics and show how they fail to pick up certain forms of entanglement, and that representations can score highly on such metrics while being strongly entangled. Specifically, we expose two problems with existing metrics: that they incorrectly align ground-truth factors to neurons, as they do not require distinct variables to be assigned to distinct factors; and that they only consider the effect of single latent variables at a time, and so they fail to detect when the representation of the ground-truth factors is distributed, in an entangled way, over neurons. To address these problems, we present two new DE metrics, based on the ability of a classifier to predict the generative factors from the encoded representation. If a representation is truly disentangled, then all the relevant information should be contained in a single neuron (or possibly a few neurons, see discussion in Section 2) and so a classifier using only this/these neuron(s) should be just as accurate as one using all neurons, and using all other neurons should be no more accurate than random guessing. Our first metric is the accuracy of

the single-neuron classifier, this should be high. Our second metric is the accuracy of the classifier using all other neurons, this should be low. We then demonstrate empirically that existing methods all perform poorly on these two metrics.

The second way in which we challenge existing DE methods is by proposing a task to judge the performance of a disentangled model, namely, the identification of novel combinations of familiar features. Humans can accomplish this easily, because we understand each separately. We can clearly picture a purple giraffe, and could recognize one if we saw it, even if we have never seen one or even heard the phrase "purple giraffe" before, because we have disentangled the concepts of colour and shape. The ability to form and understand novel combinations is a deep, important aspect of human cognition, and an essential feature of the oldest examples of human creativity. Folklore, across many diverse cultures, abounds with mythical creatures that combine features of multiple animals, such as the Minotaur or Pegasus in Greek mythology (Stratton, 2004; Hansen & Hansen, 2005). Looking even further back in our past, therianthropic cave art that combines human and animal features includes some of the earliest examples of human creative drawing (Conard, 2003). Scientific creativity, too, draws on the conception of novel combinations, such as Mendeleev's periodic table predicting the existence and properties of silicon and other then-unknown elements, through novel combinations of the features of existing elements (Mendeleev, 1871).

Such creative ability is a direct consequence of humans being able to disentangle the relevant features of the objects we encounter. Having a disentangled representation implies the ability to understand novel combinations, so, contrapositively, a system that cannot understand novel combinations is not disentangled. This is the basis for our proposed task. For example, we test whether a network trained to identify small squares, small circles and large squares can, at test time, correctly identify large circles. If it had learned to disentangle colour from shape, then it could simply identify "large" and "circle" separately, each of which is familiar. However, it turns out that existing models fail badly at this task. For every method we test, on every dataset, the accuracy for novel combinations is far below that for familiar combinations, mostly at random guessing. This is the strongest piece of evidence that the previously proposed DE models are exhibiting hidden entanglement.

Our contributions are briefly summarized as follows.

- We identify and describe two shortcomings of existing DE metrics: the incorrect alignment of neurons to factors and the failure to pick on distributed entanglements.
- We propose two alternative metrics, single-neuron classification and neuron knockout, that do not suffer from the problems that existing metrics suffer from.
- We empirically test existing DE models on our proposed metrics of single-neuron classification and neuron knockout, and show that they all perform poorly.
- We introduce the task of identifying novel combinations of familiar features, and describe why it is suitable for evaluating DE models.
- We show empirically that all existing models fail at this task on all datasets.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the shortcomings of existing DE metrics, and proposes our new metrics. Section 5 introduces the task of recognizing novel combinations. Section 6 presents the results of applying our metrics and proposed task to existing DE models, and Section 7 summarizes our work.

## 2 Related Work

**Disentanglement Models.** After the initial proposal by Bengio (2009), disentangled representations received new interest beginning with the proposal by Higgins et al. (2016) of $\beta$-VAE, which was an adaption of the variational autoencoder (VAE) (Kingma & Welling, 2013) to produce disentangled representations. By taking the prior in the VAE framework to be multivariate normal with a diagonal covariance matrix, $\beta$-VAE encourages the model representations to have diagonal covariance too, which the authors claim enforces DE. This instigated a series of VAE-based disentangled models, which, like the original VAE and like $\beta$-VAE, were fully unsupervised. Kumar et al. (2017) further encouraged a diagonal covariance matrix by introducing an additional loss term consisting of the Euclidean of the model's covariance matrix from the identity matrix. Burgess et al. (2018) propose to gradually increase the reconstruction capacity of the autoencoder by a form of annealing of

the Kullbeck-Leibler (KL) term in the VAE loss. Chen et al. (2018) seeks to minimize the total correlation of the latent variables. The total correlation is a generalization of mutual information to higher dimensions, equal to the Kullbeck-Leibler divergence of the joint distribution from the product of the marginals for each variable. Kim & Mnih (2018) propose FactorVAE, which approximates the total correlation using a discriminator network. Chen et al. (2018) propose $\beta$-TCVAE, which also aims to minimize the total correlation. Total correlation is approximated using Monte-Carlo on a sampling method inspired by importance sampling.

Locatello et al. (2019) challenged the earlier DE methods both empirically and theoretically. They proved that it is always possible for a model to learn an entangled representation that appears disentangled only on the available data. They also showed that disentangled representations do not necessarily perform better on downstream tasks such as classification, as had been claimed before. Later, Locatello et al. (2020) claimed that including a small amount of supervision was sufficient to learn disentangled representations. Despite this challenge, DE continues to be an active research area, with many models being proposed in the last two years. Some are unsupervised, such as (Klindt et al., 2020), which proposed a VAE-based model to learn disentangled representations from videos of natural scenes. Some are semi-supervised, such as (Träuble et al., 2021) which explores the post-hoc use of training labels to train a linear regressor to transform the latent representation to predict the disentangled generative factors from the (possibly entangled) representation produced by the model. These predictions are used as a new representation, which the authors claim is disentangled. Some are fully supervised, such as (Sha & Lukasiewicz, 2021), which partitions the neurons into a subset for each factor, and defines several additional DE loss terms using the ground-truth labels. Some can operate either supervised or unsupervised, such as Parted-VAE (Hajimiri et al., 2021), which encourages DE by minimizing the Bhattacharyya distance (1946) from a multivariate normal.

**Disentanglement Metrics.** Higgins et al. (2016) made an early attempt to quantify disentanglement, and proposed fixing one of the generative factors and varying all others, then using a linear model to predict which factor was fixed from the variance in each of the latent neurons. They claim that high classification accuracy implies better DE. Kim & Mnih (2018) build on this approach, and also propose a linear classifier, this time replacing the input to the classifier with the index of the neuron with the lowest empirical variance so the resulting classifier is the majority vote classifier. Ridgeway & Mozer (2018) decompose DE into two different concepts, modularity and explicitness, each with its own metric. They measure modularity as the deviation from an "ideally modular" representation, where every latent neuron has nonzero mutual information with exactly one generative factor. Explicitness is measured similarly to the metric of Kim & Mnih (2018), except using the mean of one-vs-rest classification and ROC area-under-curve (AUC) instead of accuracy.

Different to these classifier-based metrics, there has also been a series of classifier-free metrics proposed. Kumar et al. (2017) proposed the SAP score, which calculates, separately for each generative factor, the $R^2$ coefficient with each of the latent dimensions, and then takes the difference between the largest and second largest. A representation will score highly on SAP if, for each generative factor, one neuron is very informative and no other neurons are. A similar idea is employed by the mutual information gap (MIG) metric (Chen et al., 2018), except using mutual information instead of $R^2$. Eastwood & Williams (2018) decompose the task into disentanglement, completeness and informativeness (DCI). Then, they train a linear classifier to predict each generative factor from each latent factor, and estimate DCI from the weights and accuracy of the trained classifier. Do & Tran (2019) propose several metrics, most prominently RMIG, which is a different way to compute MIG, and JEMMIG which subtracts the joint entropy from mutual information gap to penalize extra information in the latent variables. These metrics are based on a decomposition into modularity, compactness and explicitness, which is also advocated by a recent survey of DE metrics (Zaidi et al., 2020). Suter et al. (2019) propose to measure the maximum amount that a given neuron can be changed by changing a generative factor other than the one it corresponds to (should be low).

**Definitions of Disentanglement.** There is some ambiguity in the literature as to whether DE requires that each factor is represented by a single neuron, or allows representation by multiple neurons. The original definition by Bengio (2009) (echoed by Higgins et al. (2016), Kim & Mnih (2018) and others) just stipulates that each neuron represents a single factor, and seems to allow that each factor is represented by multiple neurons. However, a stronger concept of DE is normally implied, namely, an injective function from factors to the unique neurons that represent them. This is implicit

| | small | large |
|---|---|---|
| Square | 0 | 0 or 1 with equal probability |
| Circle | 0 or 1 with equal probability | 1 |

Table 1: Example of how one neuron ($z_2$) can partially encode two factors.

in the common metrics, which map each factor to a single neuron, and in the descriptions of DE in other works, e.g., "learning one exclusive factor per dimension" (Pineau & Lelarge, 2018). The weaker notion of DE, which allows the representation of a factor by multiple neurons, loses some of the claimed benefits of DE: it would be hard to tell which neurons represent a given factor if we have to check all *subsets* of neurons, of which there are exponentially many, and we may still be unable to interpret what a single neuron represents, if all we know is what sets of neurons represent. For these reasons, we assume the strong notion, and it is mostly towards this notion that our challenges are directed. However, one method we test on in Section 6 uses a subset of neurons to represent shape, which allows us to see whether our arguments extend to such cases.

## 3 Problems with Existing Disentanglement Metrics

### 3.1 Incorrect Alignment of Latent Variables

The majority of existing metrics are based on aligning the set of factors $G$ with the set of neurons $Z$; that is, for each factor, finding the neuron that it is represented by. Each neuron is only supposed to represent a single factor, however all existing metrics simply relate each factor to the maximally informative variable (normally this means the variable with maximum mutual information, but can be another measure of informativeness, e.g., weight from a linear classifier (Kumar et al., 2017)). This does not enforce the constraint of having distinct neurons for distinct features, it means that the same neuron could be selected as representing multiple different factors. For example, consider again the model trained on a dataset of small squares, small circles, large squares and large circles. Suppose such a model has two latent variables, $z_1$ and $z_2$, the first is random noise, unrelated to the inputs, the second is related as per Table 1. The second variable encodes both colour and shape, each to an accuracy of 75%, whereas the other variable encodes both only to 50% (random guess). Thus, the second will be chosen as the representative of both colour and shape. We observe that cases like this often occur in practice; see the appendix for an example.

The reader may feel that the example from Table 1 is not a problem, as that neuron only represents each of the two factors to 75% accuracy, and we want a model that represents at close to 100% accuracy. However, a metric should not only work properly for models that are close to perfect, it should also be able to distinguish between two models, both of which are far from perfect but one of which is better than the other. Incorrect alignment can interfere with distinguishing between such models. To take MIG as an example, the model from Table 1, would get a higher score than another in which $z_2$ is as above and $z_1$ represents shape to an accuracy of $70\%$ (0.189 vs 0.129, calculation in supplementary material). This is a failure of the metric, as the second model is clearly closer to the desired disentangled representation than the first.

Rather than aligning each factor independently, we should align all factors simultaneously and enforce that they are aligned to distinct neurons:

$$\underset{\{f:G\to Z|\ \text{f is injective}\ \}}{\arg\max} \sum_{g\in G} I(g; f(g))\,, \tag{1}$$

where $I$ denotes mutual information (this could be replaced with $R^2$ or any other measure of informativeness). A solution to equation 1 can be computed using the Kuhn-Munkres algorithm (Munkres, 1957). The result is a mapping from factors in $G$ to neurons in $Z$ where no two factors are mapped to the same neuron. This better fits the notion of DE than previous approaches.

### 3.2 Distributed Entanglements

The second problem with existing metrics is that they miss information that may be distributed over multiple neurons. This problem manifests when measuring whether the information for one factor

is restricted to just one variable, and is a shortcoming both of classifier-based and classifier-free metrics. In what follows, let $g_0, \ldots, g_n$ be the ground-truth generative factors, and let $z_0, \ldots, z_m$, $m \geq n$ be the corresponding neurons, ordered so that $z_i$ has been selected to correspond to $g_i$ for each $i$. Further, let $z_{\neq i}$ refer to the set of all neurons other than $z_i$. Now, consider XOR: $g_0 = z_1 \oplus z_2$. (We again use discrete variables for clarity, a continuous approximation could be $|z_1 - z_2|, z_1, z_2 \in [0, 1]$.) Almost all existing metrics, both classifier-based and classifier-free, would conclude that $g_0$ is not represented by any other latent variable, and so may assign a high score.

For classifier-free metrics, the approach of existing methods is to compare individual neurons pairwise. To again take MIG as an example, the requirement is that, for each $i$, $g_i$ has low mutual information with each variable in $z_{\neq i}$. However, with XOR, this condition is met but $z_{\neq i}$ collectively still (near) perfectly represent the $g_i$. In this case, $I(g_0; z_1) = I(g_0; z_2) = 0$, but $I(g_0; z_1, z_2) \neq 0$, in fact $I(g_0; z_1, z_2)$ could be near maximal: $I(g_0; z_1, z_2) \approx H(g_0)$. The same failure mode can be found in most related metrics such as JEMMIG, SAP and DCI. Classifier-based metrics would also fail to penalize $g_0 = z_1 \oplus z_2$, because they use linear classifiers, and the strength of the linear relationship between $z_1$ and $z_2$ is zero.

To our knowledge, the only existing metric that would correctly handle this case is that of Suter et al. (2019). However, this metric still incorrectly aligns factors (Section 3.1), and may be too harsh: if there are junk neurons that vary randomly for the input but do not encode any information, this would be penalized, which may not be desirable if $m > n$. Also, taking the maximum change across the dataset means that even one input point that causes significant variation in the wrong neurons could give a very bad score even if all other examples were perfect.

# 4 Proposed Metrics

## 4.1 Single-neuron Classification (SNC)

Our first proposed metric tests whether each factor is well-represented by a single neuron, by first computing the alignment function $f$ by equation 1, then measuring how accurately each $z_i$ can classify the values of feature $f(z_i)$. In order to use $z_i$ as a classifier, we divide its values across the dataset into $K$ bins containing equal numbers of points, where $K$ is the number of classes in $f(z_i)$. That is, if $N$ is the size of the dataset, the $N/K$ smallest values are put into the first bin, etc. We then align these $K$ bins with the $K$ ground-truth classes. Similar to equation 1, this can be done with the Kuhn-Munkres algorithm. The final accuracy of the single-neuron classifier is the percentage agreement between the aligned bin labels, and the ground-truth labels. If the representation is disentangled, then this accuracy should be high, ideally as high as that of a classifier that uses all neurons. This is essentially a quantification of the property that latent traversals aim to show qualitatively. In the terminology of Ridgeway & Mozer (2018), it is a measure of explicitness, except that they fit a linear classifier on all neurons, not just $z_i$, which therefore allows for non-axis-alignment. That is, if for each $g_i$, there is some line in representation space such that the representation vector encodes $g_i$ as the distance of the projection along that line, then this is regarded as disentangled. Using a single-neuron classifier, on the other hand, requires that line to be an axis. We also report results for such a linear classifier in Section 6.2.

## 4.2 Neuron Knockout (NK)

Our second proposed metric tests whether $z_{\neq i}$ contains any information about $g_i$, by training an MLP to predict $g_i$ from $z_{\neq i}$. If the representation is disentangled, then this accuracy should be low, ideally at random guessing. By comparing to the accuracy of an MLP that uses all neurons, we can also get an indication of how relevant $z_i$ is to representing $g_i$. This is similar to the technique of gene knockout in genetics, which tests how relevant a given gene is to a given function, by removing the gene and measuring the effect on function. This is crucially different from existing methods, which only test whether each neuron individually contains information about $g_i$, and which consequently miss distributed entanglements (Section 3.2). Existing methods, e.g., fail to detect when $z_{\neq i}$ encode $g_i$ via XOR. Our method, however, because it uses an MLP, can detect non-linear encodings such as XOR. The results from Section 6 show that such encodings occur in practice.

Table 2: Accuracy predicting each factor using an MLP on all neurons (full), a single-neuron classifier (SNC), a linear classifier on all neurons (linear), and an MLP with the corresponding neuron and top two corresponding neurons knocked out (NK1 and NK1, respectively). We report the average for simple factors (size, location, colour) and complex factors (shape and orientation), and all factors. For a truly disentangled representation, all the relevant information about each factor should be given by its corresponding neuron. Therefore, SNC or, by a different interpretation of disentanglement, linear, should be as high as full, and NK1 and NK2 should be very low. However, SNC and linear are significantly below full, and NK1 and NK2 are high, suggesting entanglement. Best for each block is in bold.

| | | dsprites | | | 3dshapes | | | mpi3d | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | simple | complex | avg | simple | complex | avg | simple | complex | avg |
| $\beta$-VAE | full | 94.0 (1.1) | 82.5 (14.4) | 89.4 (1.1) | 99.8 (8.8) | 99.8 (0.2) | 99.8 (0.2) | 99.7 (0.1) | 80.1 (8.1) | 91.5 (7.5) |
| | SNC | 16.0 (3.1) | 24.6 (2.1) | 19.4 (1.9) | 17.7 (9.5) | 30.7 (11.3) | 22.0 (4.0) | 48.7 (8.2) | 8.2 (0.6) | 32.6 (5.3) |
| | linear | 58.6 (3.6) | 27.7 (1.3) | 46.2 (2.5) | 85.7 (0.0) | 65.6 (10.0) | 79.0 (1.3) | 74.1 (3.4) | 20.9 (1.4) | 52.8 (2.8) |
| | NK1 | 37.9 (5.9) | 76.8 (14.1) | 53.5 (3.7) | 98.7 (0.0) | 91.1 (8.3) | 96.1 (2.8) | 85.4 (4.9) | 62.9 (6.4) | 76.0 (2.8) |
| | NK2 | 26.8 (1.9) | 70.9 (18.4) | 44.4 (1.3) | 92.5 (8.7) | 74.2 (6.1) | 86.4 (1.8) | 76.2 (4.0) | 47.7 (4.7) | 64.1 (0.8) |
| $\beta$-TC-VAE | full | 81.7 (1.8) | 73.9 (7.8) | 78.5 (4.1) | 99.9 (0.1) | 99.9 (0.2) | 99.9 (0.1) | 89.2 (13.5) | 75.8 (8.1) | 83.9 (7.5) |
| | SNC | 55.2 (13.7) | 21.5 (6.0) | 41.7 (9.8) | 16.2 (2.1) | 30.5 (6.1) | 21.0 (2.9) | 54.6 (9.0) | 7.9 (0.6) | 35.7 (5.3) |
| | linear | 61.0 (16.0) | 51.0 (20.2) | 57.0 (17.4) | 82.1 (4.3) | 64.6 (10.8) | 76.3 (2.4) | 76.5 (5.2) | 19.5 (1.4) | 53.6 (2.8) |
| | NK1 | 53.0 (2.9) | 72.9 (7.6) | 60.9 (3.8) | 98.4 (0.7) | 92.4 (6.2) | 96.4 (2.0) | 76.3 (2.6) | 58.8 (6.4) | 69.3 (2.8) |
| | NK2 | 47.6 (2.5) | 65.1 (4.9) | 54.6 (1.7) | 92.5 (3.4) | 72.5 (4.4) | 85.8 (3.1) | 63.1 (3.1) | 43.6 (4.7) | 54.8 (0.8) |
| fVAE | full | 83.4 (1.2) | 67.5 (2.1) | 77.1 (1.5) | 97.9 (3.7) | 98.4 (1.6) | 98.1 (2.4) | 99.2 (0.9) | 74.5 (18.0) | 88.9 (7.5) |
| | SNC | 20.0 (1.5) | 23.9 (0.8) | 21.6 (0.7) | 14.6 (1.3) | 21.1 (3.0) | 16.8 (1.7) | 44.7 (2.2) | 8.0 (0.2) | 30.1 (1.2) |
| | linear | 33.9 (4.6) | 26.2 (2.2) | 30.8 (2.4) | 75.7 (4.1) | 44.2 (9.5) | 65.2 (4.1) | 67.3 (2.9) | 19.7 (2.0) | 48.4 (1.9) |
| | NK1 | 39.3 (7.3) | 60.7 (1.3) | 47.9 (4.8) | 86.9 (5.6) | 81.7 (12.0) | 85.2 (7.0) | 79.6 (1.6) | 55.3 (2.9) | 69.4 (0.6) |
| | NK2 | 26.7 (3.0) | 56.3 (1.5) | 38.5 (2.3) | 63.7 (7.7) | 59.8 (10.3) | 62.4 (7.7) | 69.9 (1.9) | 42.0 (4.2) | 57.7 (1.9) |
| Parted-VAE | full | 73.4 (2.5) | 69.1 (8.1) | 71.7 (4.2) | 98.4 (0.9) | 91.9 (15.2) | 96.2 (5.2) | 76.1 (0.6) | 51.3 (1.0) | 65.2 (0.7) |
| | SNC | 20.6 (9.9) | 25.6 (1.5) | 22.6 (6.1) | 22.1 (5.2) | 40.4 (30.8) | 28.2 (12.3) | 39.1 (3.1) | 31.6 (0.6) | 34.3 (0.9) |
| | linear | 40.6 (5.4) | 29.5 (3.9) | 36.2 (2.6) | 71.8 (5.0) | 68.7 (19.0) | 70.8 (5.8) | 53.7 (2.9) | 37.1 (0.4) | 45.8 (1.6) |
| | NK1 | 42.8 (5.0) | 52.0 (4.1) | 46.5 (4.1) | 67.2 (5.0) | 28.4 (6.8) | 54.3 (5.5) | 75.7 (3.2) | 41.6 (1.0) | 60.9 (2.0) |
| | NK2 | 36.8 (5.2) | 6.6 (1.2) | 29.2 (3.9) | 42.3 (5.2) | 20.7 (8.0) | 33.7 (6.3) | 86.2 (3.3) | 35.4 (1.5) | 61.8 (2.3) |

## 5 Novel Combinations of Familiar Features (NCFF)

In this section, we present a new task, in addition to the above metrics, for measuring disentanglement: recognizing novel combinations of familiar features. These are data points that exhibit features that have each been seen individually during training, but have not been seen together. If the representation learnt by a model is truly disentangled, then such novel combinations should not present any difficulty, because the model would be able to extract representation elements for each feature separately, and when considered separately, each is familiar to the model. However, if there is entanglement between the different features, then the novel combination is out of distribution. Formally, let $p(X, Y)$ be the empirical distribution from the data for the two features, $q(X, Y)$ the distribution learnt by the model, and $x$, $y$ the two feature values that do not appear together in the data. Then,

$$p(X = x), p(Y = y) > 0 \tag{2}$$

$$p(X = x, Y = y) = 0\,. \tag{3}$$

So, if $q$ approximates the joint distribution $p$, we would expect $q(X = x, Y = y) \approx 0$. However, if $q$ has been learned as a factored distribution, where $q(X) \approx p(X)$ and $q(Y) \approx p(Y)$, then $q(X = x, Y = y) = q(X)q(Y) > 0$. Our proposed task is as follows: randomly sample values for two chosen features, form a testset of points with those two values for those two features, train the VAE (or supervised model) on the trainset, encode the entire dataset with the encoder from the VAE (this step is not needed for supervised models), and finally train and test the classifier on the train/test sets respectively using the encodings as input.

This task is somewhat similar to the investigation into correlated features, such as those by Träuble et al. (2021). However, it differs in a number of important ways. Firstly, the dependence that we introduce between $X$ and $Y$ is restricted entirely to one combination of values, with probability zero of this combination and otherwise the features being totally independent. This is in contrast to (Träuble et al., 2021), who have some degree of correlation between all values of the two correlated features. Secondly, they do not have a real-world task with which to evaluate the effect of entanglements, instead they examine feature importance in a gradient boosting tree to predict each $g_i$, and the pairwise mutual information between $z_i$ and $z_j$ which, as we argued in Section 3.2, is a poor

method for investigating entanglement. Our proposed task of correctly classifying NCFF is a more natural, realistic way to measure if the models are learning to encode features independently.

## 6 EXPERIMENTAL EVALUATION

**Datasets.** We test on three datasets. **Dsprites** contains 737,280 black-and-white images with features $(x, y)$-coordinates, size, orientation and shape. **3dshapes** contains 480,000 images with features object/ground/wall colour, size, camera azimuth and shape. **MPI3D** contains 103,680 images of objects at the end of a robot arm with features object colour, size and shape, camera height and azimuth and altitude of the robot arm.

**Implementation Details.** For unsupervised models, parted-VAE is trained using the author's code at https://github.com/sinahmr/parted-vae, and other models are trained using the library at https://github.com/YannDubs/disentangling-vae, all use default parameters. The number of neurons is fixed to $m+2$, where $m$ is the number of factors. The MLPs and linear models trained to predict the factors are trained using Adam, learning rate $.001$, for 75 epochs. The MLP has one hidden layer of size $256$. The supervised model, MTD, is trained using the author's code (obtained privately) with all default parameters, for 10 epochs.

### 6.1 SINGLE-NEURON CLASSIFICATION AND NEURON KNOCKOUT

Table 2 shows the results of our two proposed metrics, SNC and NK on the three datasets described above. Each dataset includes a slightly different set of features, and displaying all features impairs readability, so we group the features into simple (colour, size and position) and complex (shape and orientation), for each dataset, and also report the average across all features. Full results are given in the appendix. We test several popular DE models, $\beta$-VAE, FactorVAE, $\beta$-TCVAE, and PartedVAE. The details of these models are given in Section 2. For each combination of dataset, model and factor, we report the accuracy from five different classifiers. "Full" and "linear" train an MLP and linear model, respectively, on all neurons. "SNC" is the single-neuron classifier described in Section 4.1, "NK1" and "NK2" are the two different implementations of the single-neuron classifiers described in Section 4.1, knocking out, respectively, the single neuron aligned to the factor being predicted, and the two highest neurons aligned to the factor being predicted. To find the second-highest neuron, we realign all neurons and factors using the same method, after removing the highest factor. "NK2" is inapplicable to PartedVAE, because they specify a limited number of neurons to encode class.

The single-neuron classifier (second row in each block) is very inaccurate at predicting the factor that it is supposed to be representing, compared to using an MLP on all neurons (first row in each block). With the partial exception of some simple factors on some datasets, the SNC accuracy is far lower than that of the full MLP, for both simple and complex features. The linear model performs slightly better, but is still significantly below the accuracy of the MLP. This suggests that each $g_i$ is represented much more accurately by a distributed, non-linear entangled encoding across all neurons, rather than just by $z_i$. This would mean, for example, that the decoder network makes use of this latter representation. Although an MLP is a more powerful model, this should not help test set accuracy unless there is relevant information in the input that it can leverage, i.e., distributed entanglements. The fact that MLP accuracy is much higher than linear accuracy and that of SNC, suggests that such entanglements exist. As discussed in Section 2, we mostly assume that DE requires each factor represented by a single neuron. However, PartedVAE defines a subset of neurons to represent shape, and another subset to represent other features. Here, (bottom block) "SNC" means training an MLP on those to predict class from those neurons dedicated to class, and using the above-described method to predict other factors. Unsurprisingly, this generally performs better than the other models, especially on the complex factors (wherein shape is included). However, it still performs markedly lower than the full MLP, which suggests that even this weaker notion of DE is failing to be achieved.

Distributed entanglements are also suggested by the results of the neuron knockout setting (lines four and five in each block). Here, in general, the accuracy is still quite high even after removing the neuron that was supposed to contain all the relevant information. As with SNC, there are some features that show reasonable results (see appendix for full results), but mostly the accuracy drop is only slight, suggesting that much of the representation of $g_i$ is distributed over $z_{\neq i}$. Even the NK2 setting can often classify shape nearly perfectly. This is also true of parted-VAE. As with the SNC

experiments, it performs slightly better than the other models but is still suggestive of entanglement. Note that the NK2 setting for parted-VAE (bottom row, bottom block), has four neurons out of 7-9 neurons removed (the three dedicated to class plus one extra), so deterioration in performance is expected. As well as being evidence for entanglement, this highlights that fault tolerance is at odds with DE. In one sense, it is a good thing that after removing any one or two neurons, even those with the highest mutual information with the target feature, the representation still contains enough information to classify with high accuracy. Fault-tolerance was one of the original arguments for the advantages of distributed representations (Fahlman & Hinton, 1987; Hinton et al., 1986), and the brain has been shown to be highly fault-tolerant (Yu, 2016). However, if, as disentangled models aim for, the information for each feature is contained in only one neuron, then the representation is very fragile, losing just one neuron means all the information is lost. This conceptual conflict with fault-tolerance suggests that DE may not always be desirable, and we should be clear what benefit DE offers if we are to pursue it. The results from Table 2 show that current supposedly disentangled models largely retain the benefit of fault-tolerance, at the cost of entanglement.

### 6.2 Novel Combinations of Familiar Features Results

In this section, we present the results for the task of classifying novel combinations of familiar features. As well as the models from Section 6.1, we also report results on a recent fully supervised method (Sha & Lukasiewicz, 2021), which we denote MTD. Because of the architecture of MTD, it is not suitable to evaluate it on the SNC and NK metrics, but it is suitable to see whether it can correctly identify novel combinations of familiar features. Here, the train set consists of all data points other than those with a specific combination of values for two specific features, e.g., large circles, and the test set consists of all other data points. Because there is a very large number of combinations of feature values, it is not feasible to test all of them in enough detail to obtain reliable results. We restrict our attention to just shape and size combinations. We randomly sample a number of shape-size value combinations to exclude, five for dsprites, six for mpi3d and 8 for 3dshapes (see appendix for details), and report the average. For parted-VAE and MTD, the "normal test set" predicts factors from the corresponding specified subsets of neurons, for other models, all neurons are used.

Points from the test set are excluded both from the training of the VAE and of the classifier (same in the "normal test set" setting). There is perhaps a danger here that the MLP itself entangles two features that are not entangled in the representation itself. For example, if large circles are excluded then, when classifying shape, the MLP could learn that whenever the "colour" neuron indicates "large", it should place low probability mass on "circle". We feel this is unlikely to affect results significantly, as the relationship between the "size" neuron and the value of shape would be highly non-linear and present only for a small subset of data points. However, to ensure that this is not the reason for such a failure to classify NCFF, we also test under three additional settings: "NCFF linear" trains a linear classifier instead of an MLP, so the non-linear relationship could not be learnt, and "NCFF excl1/2" exclude from the classifier, the top one and two neurons (respectively) for the other features. That is, when classifying shape, the neuron(s) for size are removed and vice versa.

As Table 3 shows, the accuracy for all three of these settings, as well as in the original "NCFF" setting, is very low, essentially at the level of random guessing. This is even true for MTD, the supervised model. On the normal test set, where the data points are split into train-test randomly every model is capable of classifying the unseen data accurately. They are all $75 + \%$, with many at $90 + \%$. For the novel combinations, however, the model tends to capture one factor accurately, but the other factor very inaccurately. This happens even in the "NCFF linear" and "NCFF excl 1/2" settings which, as discussed above, shows that it is not the classifier that is introducing entanglement. Rather, the entanglement must be in the representation itself. The failure to classify NCFF suggests that, however the model has learned to encode the data, it is not able to represent the range of values independently for each feature type. For example, it is completely unable to represent "square" without also representing "not large". This would mean that the representation is highly entangled.

Some prior works have investigated novel combinations of familiar features before, and mostly they have claimed their model can meaningfully represent such combinations, however, these prior works have only examined the reconstructions, i.e., the output of the decoder: Higgins et al. (2016) display figures of reconstructed chairs with a round bottom for certain latent traversals and Esmaeili et al. (2019) present reconstructions for MNIST digits with certain combinations of digits and features

Table 3: Classification accuracy for novel combinations of familiar features, best for each block in bold. For a truly disentangled representation, this task should not present additional difficulty, because each factor can be recognized separately. However, NCFF is consistently much lower than for a normal test set, mostly at the level of random guessing. The same is true when using a linear classifier (NCFF linear) and when excluding opposite neurons (NCFF excl 1/2), showing the entanglement is not merely introduced by the classifier. The failure at this task suggests entanglement.

| | | dsprites | | | 3dshapes | | | mpi3d | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | size | shape | both | size | shape | both | size | shape | both |
| $\beta$-VAE | normal testset | 99.0 | 99.6 | **98.7** | 99.4 | 100.0 | **99.9** | 98.9 | 85.8 | **85.1** |
| | NCFF | 5.3 | 4.2 | 0.0 | 16.9 | 27.7 | 0.0 | 72.8 | 5.2 | 4.3 |
| | NCFF linear | 2.5 | 91.6 | 2.0 | 36.0 | 70.6 | 17.2 | 95.1 | 7.6 | 5.0 |
| | NCFF excl 1 | 3.2 | 90.4 | 2.0 | 32.7 | 71.6 | 14.0 | 88.5 | 9.1 | 7.0 |
| | NCFF excl 2 | 4.2 | 94.9 | 3.8 | 36.1 | 70.8 | 17.4 | 97.8 | 11.3 | 10.0 |
| $\beta$-TCVAE | normal testset | 88.6 | 94.3 | **85.7** | 99.6 | 100.0 | **99.6** | 97.6 | 81.2 | **79.9** |
| | NCFF | 0.2 | 38.0 | 0.0 | 7.1 | 61.4 | 3.9 | 83.0 | 3.6 | 3.4 |
| | NCFF linear | 0.0 | 99.8 | 0.0 | 41.2 | 81.1 | 28.9 | 98.0 | 11.3 | 10.6 |
| | NCFF excl 1 | 0.0 | 92.5 | 0.0 | 28.7 | 79.8 | 19.9 | 92.1 | 13.0 | 11.6 |
| | NCFF excl 2 | 0.0 | 82.9 | 0.0 | 32.6 | 80.6 | 20.3 | 96.8 | 6.4 | 5.3 |
| FactorVAE | normal test set | 97.5 | 98.4 | **96.4** | 92.6 | 98.8 | **91.5** | 97.9 | 82.3 | **80.9** |
| | NCFF | 3.1 | 28.3 | 0.3 | 13.5 | 36.9 | 1.2 | 67.3 | 2.6 | 2.1 |
| | NCFF linear | 14.4 | 92.6 | 12.2 | 40.8 | 62.1 | 18.1 | 91.9 | 9.6 | 8.8 |
| | NCFF excl 1 | 11.2 | 95.6 | 11.0 | 36.1 | 60.3 | 19.4 | 87.7 | 9.6 | 8.6 |
| | NCFF excl 2 | 8.5 | 70.6 | 6.4 | 28.8 | 67.8 | 9.3 | 97.8 | 9.0 | 8.3 |
| PartedVAE | normal testset | 77.0 | 97.4 | **75.9** | 96.7 | 99.0 | **95.8** | 80.7 | 100.0 | **29.1** |
| | NCFF | 3.6 | 25.6 | 0.0 | 10.0 | 87.4 | 9.7 | 79.4 | 0.0 | 0.0 |
| | NCFF linear | 2.2 | 12.7 | 0.3 | 14.2 | 95.3 | 4.5 | 87.8 | 0.0 | 0.0 |
| | NCFF excl 1 | 3.1 | 28.3 | 1.3 | 25.7 | 21.1 | 0.5 | 92.6 | 0.0 | 0.0 |
| | NCFF excl 2 | 0.6 | 62.9 | 0.4 | 22.2 | 95.2 | 21.6 | 90.1 | 0.2 | 0.2 |
| MTD | normal testset | 79.9 | 80.2 | **99.7** | 100.0 | 100.0 | **100.0** | 99.9 | 95.8 | **95.8** |
| | NCFF | 11.3 | 33.4 | 2.9 | 12.1 | 23.3 | 1.3 | 59.9 | 16.5 | 5.0 |
| - | random | 16.7 | 33.3 | 5.6 | 12.5 | 25.0 | 3.1 | 50.0 | 16.7 | 8.3 |

(e.g., line thickness) excluded. However, we argue that these investigations do not provide sufficient evidence to make claims about the internal representation. Being able to reconstruct an image accurately does not imply anything about the internal representation, it could just be the result of learning the identity function. The key question is not what the decoder reconstructs from the encoding or from a latent traversal, it is what representation the encoder produces. Prior investigations into NCFF only examined the former, and so wrongly concluded that their models could understand NCFF. Our proposed task of classifying NCFF, on the other hand, directly investigates what information is present in the encoder's representations of the data, and finds, contra the claims of prior works but in line with the results from Section 3.2, evidence of strong entanglement.

## 7 Conclusion

This paper has cast doubt on whether current, supposedly disentangled models are really producing disentangled representations. First, we examined the metrics that are used to assess performance of disentangled models and identified that they have two significant flaws: incorrect alignment of generative factors with neurons and a failure to detect information spread over multiple neurons, which we call distributed entanglement. We described two new metrics that avoid these problems: single-neuron classification and neuron knockout. We then tested existing DE models on these new metrics and showed that, for most features, they perform poorly. Next, we propose the classification of novel combinations of familiar features as a real-world task with which disentanglement can be measured. Disentangled models should be able to perform this task, but when we test it empirically, it turns out they perform little better than random guessing, which suggests entanglement.

## REFERENCES


Yoshua Bengio. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

Yoshua Bengio. Deep learning of representations: looking forward. In *Proceedings of the International Conference on Statistical Language and Speech Processing*, pp. 1–37. Springer, 2013.

Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In *Proceedings of the International Conference on Machine Learning*, pp. 552–560. PMLR, 2013.

A. Bhattacharyya. On a measure of divergence between two multinomial populations. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 7(4):401–406, 1946.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in $\beta$-VAE. *arXiv preprint ArXiv:1804.03599*, 2018.

Ricky TQ Chen, Xuechen Li, Roger B. Grosse, and David K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in Neural Information Processing Systems*, 31, 2018.

Nicholas J. Conard. Palaeolithic ivory sculptures from southwestern germany and the origins of figurative art. *Nature*, 426(6968):830–832, 2003.

Kien Do and Truyen Tran. Theory and evaluation metrics for learning disentangled representations. *arXiv preprint ArXiv:1908.09961*, 2019.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *Proceedings of the International Conference on Learning Representations*, 2018.

Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H. Brooks, Jennifer Dy, and Jan-Willem Meent. Structured disentangled representations. In *Proceedings of the The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2525–2534. PMLR, 2019.

Scott E. Fahlman and Geoffrey E. Hinton. Connectionist architectures for artificial intelligence. *Computer*, 20(01):100–109, 1987.

Sina Hajimiri, Aryo Lotfi, and Mahdieh Soleymani Baghshah. Semi-supervised disentanglement of class-related and class-independent factors in VAE. *arXiv preprint ArXiv:2102.00892*, 2021.

William F. Hansen and William Hansen. *Classical Mythology: a Guide to the Mythical World of the Greeks and Romans*. Oxford University Press, USA, 2005.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. $\beta$-VAE: Learning basic visual concepts with a constrained variational framework. 2016.

Geoffrey E. Hinton et al. Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, volume 1, pp. 12. Amherst, MA, 1986.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *Proceedings of the International Conference on Machine Learning*, pp. 2649–2658. PMLR, 2018.

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

David Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint ArXiv:2007.10930*, 2020.

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

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the International Conference on Machine Learning*, pp. 4114–4124. PMLR, 2019.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *Proceedings of the International Conference on Machine Learning*, pp. 6348–6359. PMLR, 2020.

DI Mendeleev. A natural system of the elements and its use in predicting the properties of undiscovered elements. *Tomado En Http://www. Rod. Beavon. Clara. Net/neweleme. Htm*, 1871.

James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

Edouard Pineau and Marc Lelarge. InfoCatVAE: representation learning with categorical variational autoencoders. *arXiv preprint ArXiv:1806.08240*, 2018.

Karl Ridgeway and Michael C. Mozer. Learning deep disentangled embeddings with the f-statistic loss. *Advances in Neural Information Processing Systems*, 31, 2018.

Lei Sha and Thomas Lukasiewicz. Multi-type disentanglement without adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9515–9523, 2021.

Carol Stratton. *Buddhist Sculpture of Northern Thailand*. Serindia Publications, Inc., 2004.

Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: validating deep representations for interventional robustness. In *Proceedings of the International Conference on Machine Learning*, pp. 6056–6065. PMLR, 2019.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. In *Proceedings of the International Conference on Machine Learning*, pp. 10401–10412. PMLR, 2021.

Byron M. Yu. Fault tolerance in the brain. *Nature*, 532(7600):449–450, 2016.

Julian Zaidi, Jonathan Boilard, Ghyslain Gagnon, and Marc-André Carbonneau. Measuring disentanglement: A review of metrics. *arXiv preprint ArXiv:2012.09276*, 2020.