A FRAMEWORK FOR THE QUANTITATIVE EVALUATION OF DISENTANGLED REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent AI research has emphasized the importance of learning *disentangled* representations of the explanatory factors behind data. Despite the growing interest in models which can learn such representations, visual inspection remains the standard evaluation metric. While various desiderata have been implied in recent definitions, it is currently unclear what exactly makes one disentangled representation better than another. In this work we propose a framework for the quantitative evaluation of disentangled representations when the ground-truth latent structure is available. Three criteria are *explicitly* defined and *quantified* to elucidate the quality of learnt representations and thus compare models on an equal basis. To illustrate the appropriateness of the framework, we employ it to compare quantitatively the representations learned by a state-of-the-art model (InfoGAN) and those learned by a baseline model (PCA).

1 INTRODUCTION

To gain a conceptual understanding of our world, models must first learn to understand the factorial structure of low-level sensory input without supervision (Bengio et al., 2013; Lake et al., 2016; Higgins et al., 2017). As argued in several notable works (Desjardins et al., 2012; Bengio et al., 2013; Chen et al., 2016; Higgins et al., 2017), this understanding can only be gained if the model learns to *disentangle* the underlying explanatory factors hidden in unlabelled input.

A disentangled representation is generally described as one which separates the factors of variation, explicitly representing the important attributes of the data (Desjardins et al., 2012; Bengio et al., 2013; Cohen & Welling, 2014b; Kulkarni et al., 2015; Chen et al., 2016; Higgins et al., 2017). For example, given an image dataset of human faces, a disentangled representation may consist of separate dimensions (or features) for the face size, hairstyle, eye colour, facial expression, etc. Ultimately, we would like to learn representations that are invariant to irrelevant changes in the data. However, the relevant downstream tasks are generally unknown at training time and hence it is difficult to deduce a priori which features will be useful. Thus, the most robust method is to disentangle as many factors of variation as possible, discarding as little information as possible (Desjardins et al., 2012; Bengio et al., 2013).

Despite the expanding literature on models which seek to learn disentangled representations (Desjardins et al., 2012; Reed et al., 2014; Zhu et al., 2014; Cheung et al., 2014; Larsen et al., 2015; Makhzani et al., 2015; Yang et al., 2015; Kulkarni et al., 2015; Whitney et al., 2016; Chen et al., 2016; Higgins et al., 2017; Denton & Birodkar, 2017), visual inspection remains the standard evaluation metric. While the work of Higgins et al. (2017) partially addresses this issue (as discussed in section 3) and various definitions have implied additional desiderata like interpretability (Bengio et al., 2013; Kulkarni et al., 2015; Chen et al., 2016), invariance (Goodfellow et al., 2009; Cohen & Welling, 2014a;b; Lenc & Vedaldi, 2015) and equivariance (Kivinen & Williams, 2011; Lenc & Vedaldi, 2015; Jayaraman & Grauman, 2015), current research generally lacks a clear metric for quantitatively evaluating and comparing disentangled representations.

In this work we propose a framework for the quantitative evaluation of disentangled representations when the ground-truth latent structure is available. To elucidate the quality of learnt representations and thus compare models on an equal basis, desiderata of disentangled representations are *explicitly* defined and *quantified*. These unified desiderata help define the disentangled representations which we seek and remove the need for a subjective visual evaluation by a human arbiter. To illustrate

the appropriateness of this framework, we employ it to compare quantitatively the representations learned by information maximizing generative adversarial networks (InfoGAN, Chen et al. 2016) and those learned by principal components analysis (PCA).

In the remainder of this paper, we begin by detailing the theoretical framework and how it facilitates the quantitative evaluation of disentangled representations. Next we review related desiderata and metrics for evaluating disentangled representations. Finally, we describe the dataset and model specifics before presenting the experimental results.

2 THEORETICAL FRAMEWORK

Models for disentangled factor learning seek a compact data representation or *code* $\boldsymbol{c}$ of dimension D, which consists of disentangled and interpretable latent variables. For synthetic data, the K-dimensional generative factors $\boldsymbol{z}$ are designed to be an ideal such representation. Thus if $D = K$ the ideal disentangled code $\boldsymbol{c}^*$ should be some (scaled) permutation of $\boldsymbol{z}$, i.e. they should be related by a generalized permutation matrix (or monomial matrix [1]). If $D > K$, one would expect to obtain this monomial structure along with a number of 'dead' or irrelevant units in $\boldsymbol{c}$ which are not predictive of / informative about $\boldsymbol{z}$. Thus, we can quantitatively evaluate the codes learned by a given model M using the following steps:

1. Train M on a synthetic dataset with generative factors $\boldsymbol{z}$
2. Retrieve $\boldsymbol{c}$ for each sample $\boldsymbol{x}$ in the dataset ($\boldsymbol{c} = M(\boldsymbol{x})$)
3. Train regressor f to predict $\boldsymbol{z}$ given $\boldsymbol{c}$ ($\hat{\boldsymbol{z}} = f(\boldsymbol{c})$)
4. Quantify f's deviation from the ideal mapping and the prediction error

We now detail the proposed evaluation metrics, i.e., steps 3 and 4. We train K regressors to predict the value of K generative factors. The regressor f_j predicts z_j given $\boldsymbol{c}$, that is, it learns a mapping $f_j(\boldsymbol{c}) : \mathbb{R}^D \rightarrow \mathbb{R}^1$. We begin with linear regressors and encourage a *sparse* mapping between $\boldsymbol{c}$ and $\boldsymbol{z}$ with an ℓ_1 regularisation penalty (lasso regressors). With the inputs and targets normalised to have zero mean and unit variance, the magnitude of the resulting regression weights rank the learnt code variables $c_0, \ldots, c_{D-1}$ in order of *relative importance* to the prediction. That is, they reveal which code variables capture information about a given generative factor. Thus, we define the matrix of relative importances R as $R_{ij} = |W_{ij}|$ for linear regression, where R_{ij} denotes the relative importance of c_i in predicting z_j and $|W_{ij}|$ denotes the magnitude of the weight used to scale c_i in predicting z_j. This allows us to explicitly define and quantify three criteria of disentangled representations or *codes* which are implicit in recent definitions (Desjardins et al., 2012; Bengio et al., 2013; Kulkarni et al., 2015; Chen et al., 2016; Higgins et al., 2017), namely *disentanglement*, *completeness* and *informativeness*.

Disentanglement. The degree to which a representation factorises or *disentangles* the underlying factors of variation, with each variable (or dimension) capturing at most one generative factor. The disentanglement score D_i of code variable c_i is quantified by $D_i = (1 - H_K(P_{i.}))$, 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 'probability' of c_i being important for predicting z_j. If c_i is important for predicting a single generative factor, the score will be 1. If c_i is equally important for predicting all generative factors, the score will be 0. D_i can be visualised by examining row i of the Hinton diagrams as in Figure 3.

In order to account for dead or irrelevant units in $\boldsymbol{c}$, relative code variable importance $\rho_i = \sum_j R_{ij} / \sum_{ij} R_{ij}$ is used to construct a *weighted* average $\sum_i \rho_i D_i$ expressing overall disentanglement. If a code variable c_i is irrelevant for predicting $\boldsymbol{z}$, then its ρ_i (and thus contribution to the overall disentanglement) will be near zero.

Completeness. The degree to which each underlying factor is captured by a single code variable. The completeness score C_j in capturing generative factor z_j is quantified by $C_j = (1 - H_D(\tilde{P}_{.j}))$,

[1] A matrix is monomial if there is exactly one non-zero element in each row and column. If the non-zero elements have value 1 the matrix is a permutation matrix.

where $H_D(\tilde{P}_{\cdot j}) = -\sum_{d=0}^{D-1} \tilde{P}_{dj} \log_D \tilde{P}_{ij}$ denotes the entropy of the $\tilde{P}_{\cdot j}$ distribution. If a single code variable contributes to z_j's prediction, the score will be 1 (complete). If all code variables equally contribute to z_j's prediction, the score will be 0 (maximally overcomplete). C_j can be visualised by examining column j of the Hinton diagrams as in Figure 3.

Informativeness. The amount of information that a representation captures about the underlying factors of variation. To be useful for natural tasks which require knowledge of the important attributes of the data (e.g. object recognition), representations must ultimately capture information about the underlying factors of variation (Bengio et al., 2013; Chen et al., 2016). The informativeness of code $\boldsymbol{c}$ about generative factor z_j is quantified by the prediction error $E(z_j, \hat{z}_j)$ (averaged over the dataset), where E is an appropriate error function and $\hat{z}_j = f_j(\boldsymbol{c})$. It is important to note that the prediction error $E(z_j, \hat{z}_j)$, and thus this informativeness metric, is dependent on the capacity of f, with linear regressors only capable of extracting information about $\boldsymbol{z}$ in $\boldsymbol{c}$ that is explicitly represented (linearly encoded). Hence this informativeness metric is also dependent on a model's ability to explicitly represent information about $\boldsymbol{z}$ in $\boldsymbol{c}$, which in turn is dependent on the model's ability to disentangle the underlying factors of variation ($\boldsymbol{z}$). Thus the informativeness metric has some overlap with the disentanglement metric, with the size of the overlap determined by the capacity of f (no overlap with infinite capacity).

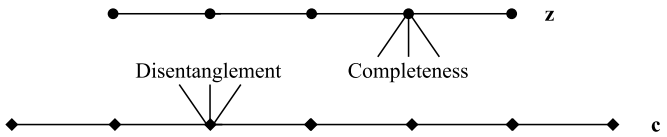


Figure 1: **Visualising disentanglement and completeness.**

While the disentanglement score quantifies the number of generative factors captured by a given code variable, the completeness score quantifies the number of code variables which capture a given generative factor. Together, these scores quantify the deviation from the ideal one-to-one mapping between $\boldsymbol{z}$ and K of the dimensions in $\boldsymbol{c}$. Figure 1 illustrates this idea.

Despite the overlap between the disentanglement and informativeness metrics with low-capacity linear regressors, these are ultimately *distinct* criteria. While disentanglement requires each code variable in $\boldsymbol{c}$ to be only perturbed by changes in a single z, informativeness requires these perturbations to be systematic and thus *informative*. This motivates the use of non-linear regressors. We use random forest regressors due to their inbuilt ability to determine the relative importance of each feature to a given prediction, thus allowing us to quantify the degree of disentanglement and completeness as before by directly specifying the matrix of relative importances R. Random forests average the predictions and feature importances from each decision tree in the ensemble. The number of times a tree chooses to split on a particular input variable determines its importance to the prediction. Thus, the relative importance of each input variable c_i is given by the number of cases split on c_i over the total number of splits (Breiman et al., 1984).

While the ideal code would be able to explicitly represent each generative factor with a single variable, models with generic priors cannot be expected to learn such complete and explicit codes. For example, generative factors which are drawn from a distribution on a circle cannot be accurately captured by single code variables on which unwrapped prior distributions are imposed. Thus, with generic priors like the standard normal, information about such generative factors may be non-linearly encoded across multiple code variables. Empirical results in Appendix B, Appendix C and (Higgins et al., 2017, fig. 7) support this idea, with several code variables resembling non-linear functions (like the sine and cosine) of the object azimuth. This further motivates the use of non-linear regressors.

Our criteria assume that it is possible to recover the latent factors $\boldsymbol{z}$ from the data. If the data $\boldsymbol{x}$ depends on a *linear* combination of (some of) the underlying z's with a spherically symmetric distribution, then it will only be possible to recover these components up to a rotation matrix. This is the well-known issue of the rotation of factors in the linear factor analysis model (see e.g., Mardia, Kent, and Bibby 1979, sec. 9.6), and also leads to the condition in independent components analysis (ICA) that at most one of the z's can be Gaussian (Hyvärinen et al., 2001). In this case, the infor-

mativeness metric remains valid but the disentanglement and completeness metrics do not as they are dependent on the arbitrary rotation which determines $\boldsymbol{c}$'s alignment with $\boldsymbol{z}$. Although $\boldsymbol{z}$ may be used to compute the rotation matrix which best aligns $\boldsymbol{c}$ and $\boldsymbol{z}$, we ultimately wish to evaluate models which will not have access to $\boldsymbol{z}$ at test time.

3 Related Work

The question of how well a learned representation $\boldsymbol{c}$ matches the true generative factors $\boldsymbol{z}$ has been considered in the 'square' case of independent components analysis (ICA), where $D = K$. In the ICA case, the data is generated as $\boldsymbol{x} = A\boldsymbol{z}$ and the learned representation is obtained as $\boldsymbol{c} = W\boldsymbol{x}$, where A is the mixing matrix and W is the learned 'un-mixing' matrix. Ideally $P = WA$ will be equal to a permutation matrix. Yang & Amari (1997, sec. 6.1) propose an error metric to assess how close P is to a permutation matrix[2]. This metric takes the form

$$E = \sum_i \left(\sum_j \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1 \right) + \sum_j \left(\sum_i \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1 \right), \tag{1}$$

summing two terms which have similar goals to our disentanglement and completeness scores respectively, although expressed by comparing with the maximum value in the row or column, rather than via an entropic measure. Note that, due to the linear structure of ICA, there is no explicit mapping between $\boldsymbol{c}$ and $\boldsymbol{z}$. We report separate scores as they capture distinct criteria and go beyond this metric by handling the non-square case when $D > K$.

Predicting $\boldsymbol{z}$ from $\boldsymbol{c}$ has also been considered previously. Higgins et al. (2017, sec. A.5) use a linear classifier to predict discrete settings of $\boldsymbol{z}$ and thus quantify the amount of explicit information about $\boldsymbol{z}$ in $\boldsymbol{c}$, albeit with an unnecessary discretisation. Higgins et al. (2017, sec. 3) also propose a disentanglement metric. With this method, one of the generative factors say z_k is held fixed, and pairs of $\boldsymbol{x}$'s are drawn, generated with different random $\boldsymbol{z}$'s except for the fixed z_k. Pairwise absolute differences of the resulting codes $|\boldsymbol{c}_1 - \boldsymbol{c}_2|$ are then computed and averaged over repetitions before being used to train a linear classifier to predict which generative factor was held fixed. In our view this is unnecessarily cumbersome—by setting up a regression problem to predict $\boldsymbol{z}$ from $\boldsymbol{c}$ as we have done, the structure of the R_{ij} matrix can be interrogated to quantify the degree of disentanglement. In addition, this facilitates the quantification of additional criteria, namely completeness and informativeness, without needing to generate any additional datasets.

Glorot et al. (2011, fig. 3) predict $\boldsymbol{z}$ from $\boldsymbol{c}$ using a lasso regressor but only to qualitatively assess disentanglement, visually assessing the overlap of important features for the separate tasks of domain recognition and sentiment classification. Karaletsos et al. (2015) do so with an unspecified regressor, thus quantifying informativeness. In addition, they devise a quantitative metric to determine a model's ability to disentangle the underlying factors of variation in images. In particular, they predict the order of query-specific oracle triplets of images, where the order indicates image similarity with respect to a query (e.g., 'Where is the light condition most similar in terms of azimuth?'). However, the proposed metric is specifically designed to evaluate their 'oracle-prioritized belief network' and thus overly cumbersome to be used as a generic disentanglement metric.

Properties such as invariance and equivariance have been proposed as desiderata for representations or codes (Goodfellow et al., 2009; Kivinen & Williams, 2011; Cohen & Welling, 2014b; Lenc & Vedaldi, 2015; Jayaraman & Grauman, 2015). In our view these qualities arise naturally from a properly disentangled and informative code. Consider, for example, the code of an object which consists of separate variables for its class (e.g., cup, bottle, banana etc.), position, pose, texture etc. If the object is translated, its position code variable(s) will transform accordingly (equivariance), but other code variables will remain invariant.

4 Experiments

We employ the framework to compare quantitatively the codes learned by InfoGAN (Chen et al., 2016) to those learned by PCA. The scores for each criteria quantify how superior InfoGAN's codes are to PCA's and also explain *why* this is the case—illustrating the appropriateness of the framework.

[2]We thank Andriy Mnih for pointing out to us the work of Yang and Amari.

(a) **Seen factor combinations**

(b) **Unseen factor combinations**

Figure 2: **Data samples.** (a) Examples of images (and corresponding generative factor combinations) on which the models are trained. (b) Examples of images in the 'gap' that was created containing unseen factor combinations.

4.1 DATA

The graphics renderer described in (Moreno et al., 2016) was employed to generate 200,000 images of an object (teapot) with varying pose and colour (see Figure 2a). For simplicity, the camera is centred on the object, the scene background is removed and additional generative factors (shape and lighting) are held constant. Each generative factor is independently sampled from its respective uniform distribution: azimuth(z_0) $\sim U[0, 2\pi]$, elevation(z_1) $\sim U[0, \pi/2]$, red(z_2) $\sim U[0, 1]$, green(z_3) $\sim U[0, 1]$, blue(z_4) $\sim U[0, 1]$. In line with recent architectures, we set the image dimensions to be $64 \times 64 \times 3$.

Disentangled representations should enable a model to perform zero-shot inference, that is, generalise its knowledge beyond the training distribution by recombining previously-learnt factors (Bengio et al., 2013; Higgins et al., 2017). Thus, we can further evaluate the disentangled representations learned by a given model by quantifying its ability to perform zero-shot inference. We use the ground-truth values of the generative factors to create two different data distributions. More specifically, we isolate all images whose generative factor values lie in a particular range to create a 'gap' in the original dataset. This gap then serves as our zero-shot data containing unseen factor combinations. Informally, the images in this gap can be described as 'red' teapots from 'above'. Formally, the generative factors of these images satisfy the following condition: $z_2 > (z_3 + 0.15)$ and $z_2 > (z_4 + 0.15)$ and $z_1 > \frac{\pi}{4}$. This dataset contained approximately 20,000 images, with (extreme) samples given in Figure 2b. Note that zero-shot inference is facilitated by disentangled and informative representations, thus is not a core component of our framework, but rather a 'bonus'.

4.2 MODELS

We train both InfoGAN (Chen et al., 2016) and PCA with 6 code variables. For illustrative purposes, we choose the best of 10 random runs by visual inspection for InfoGAN as we found the model to be quite sensitive to random initialisation. Note that InfoGAN's code variables are termed 'latent codes' in (Chen et al., 2016) while PCA's are termed principal components. Further details on the InfoGAN model setup are provided in Appendix A.1.

4.3 RESULTS

As each target is normalised to have a standard deviation of 1, the root-mean-square error (RMSE) in predicting each target is naturally normalised relative to the constant regressor which guesses the expected value of the targets. Hence, we report the normalised root-mean-square error (NRMSE) in predicting each target.

Tables 1a and 2a present the disentanglement scores for the lasso and random forest regressors respectively. With both regressors, the variables in InfoGAN's code ($\boldsymbol{c}-\text{InfoGAN}$) achieve much higher disentanglement scores than those in PCA's code ($\boldsymbol{c}-\text{PCA}$), with each variable in $\boldsymbol{c}-\text{InfoGAN}$ closer to capturing a single generative factor. That is, $\boldsymbol{c}-\text{InfoGAN}$ is more *disentangled*. Figure 3 helps to visualise the disentanglement and identify the generative factors captured by each code variable. For example, comparing $c_0-\text{PCA}$ and $c_0-\text{InfoGAN}$ in figure 3 (the first rows),

it is clear that $c_0-\mathrm{PCA}$ captures information about each generative factor while $c_0-\mathrm{InfoGAN}$ (almost) solely captures information about z_4. The lack of disentanglement within $\boldsymbol{c}-\mathrm{PCA}$ indicates that PCA is unable to separate the factors of variation ($\boldsymbol{z}$) in the data, ultimately preventing it from generalising to data with unseen factor combinations. This is illustrated by the high zero-shot NRMSE of $\boldsymbol{c}-\mathrm{PCA}$ in tables 1d and 2d, with $\boldsymbol{c}-\mathrm{PCA}$ even outperformed by the constant regressor in predicting z_1. In contrast, $\boldsymbol{c}-\mathrm{InfoGAN}$ predicts the value of unseen factor combinations reasonably well.

Tables 1c and 2c present the completeness scores for the lasso and random forest regressors respectively. The high completeness scores of $\boldsymbol{c}-\mathrm{InfoGAN}$ reveal that it captures each generative factor with approximately one code variable. That is, they show that $\boldsymbol{c}-\mathrm{InfoGAN}$ is almost complete. In contrast, the low scores of $\boldsymbol{c}-\mathrm{PCA}$ reveal that is it severely overcomplete, using several code variables to capture each generative factor. Again, Figure 3 helps to identify the generative factors captured by a given code variable and visualise the completeness. With an ideal representation, Figure 3 would show a single large weight in K rows and each column, indicating a one-to-one mapping between $\boldsymbol{z}$ and K of the dimensions in $\boldsymbol{c}$. Figure 3 also helps to explain the low completeness score (overcompleteness) of $\boldsymbol{c}-\mathrm{InfoGAN}$ in predicting z_0, with z_0 clearly captured by a combination of $c_1-\mathrm{InfoGAN}$ and $c_3-\mathrm{InfoGAN}$. However, this is still less overcomplete than $\boldsymbol{c}-\mathrm{PCA}$, with each of its constituent variables capturing distinct information about z_0 (see Figure 3b).

Tables 1b and 2b present the test set NRMSEs in predicting $\boldsymbol{z}$ for factor combinations similar to those on which the models were trained. $\boldsymbol{c}-\mathrm{InfoGAN}$ clearly outperforms $\boldsymbol{c}-\mathrm{PCA}$ in predicting each generative factor with both regressors. That is, it is far more *informative*. It is worth noting the significantly higher error in predicting the azimuth (z_0) from $\boldsymbol{c}-\mathrm{InfoGAN}$. This can be attributed to: (i) the low-capacity linear regressor being unable to extract the information encoded in $\boldsymbol{c}-\mathrm{InfoGAN}$ about the azimuth (indicated by much lower azimuth prediction error of the non-linear regressor in Table 2b); (ii) InfoGAN struggles to capture enough information about the azimuth in $\boldsymbol{c}-\mathrm{InfoGAN}$ (indicated by the relatively high error in predicting the azimuth with both regressors). Comparing the results of the linear regressor in Table 1b with those of the non-linear regressor in Table 2b, we see that both codes better predict the generative factors with increased capacity—especially $\boldsymbol{c}-\mathrm{PCA}$. As discussed in section 2, the prediction error with the non-linear random forest regressor is likely a better quantification of informativeness as it is less dependent on the ability of the model to explicitly represent information about $\boldsymbol{z}$ in $\boldsymbol{c}$.

(a) **Disentanglement**

Code	c_0	c_1	c_2	c_3	c_4	c_5	**W. Avg.**
PCA	0.16	0.38	0.42	0.14	0.63	0.16	0.20
InfoGAN	0.83	0.64	0.65	0.93	0.80	0.80	0.77

(b) **Informativeness**

Code	z_0	z_1	z_2	z_3	z_4	**Avg.**
PCA	0.99	0.72	0.42	0.47	0.46	0.59
InfoGAN	0.50	0.09	0.09	0.13	0.09	0.18

(c) **Completeness**

Code	z_0	z_1	z_2	z_3	z_4	**Avg.**
PCA	0.83	0.54	0.14	0.20	0.19	0.38
InfoGAN	0.56	0.85	0.80	0.63	0.78	0.72

(d) **Zero-shot**

Code	z_0	z_1	z_2	z_3	z_4	**Avg.**
PCA	1.01	1.78	0.77	0.91	0.91	1.05
InfoGAN	0.49	0.20	0.35	0.21	0.16	0.28

Table 1: **Lasso regression results**. **(a)** Disentanglement scores for each code variable in $\boldsymbol{c}-\mathrm{InfoGAN}$ and $\boldsymbol{c}-\mathrm{PCA}$. 'W. Avg.' abbreviates weighted average. **(b)** Test set NRMSE on images containing factor combinations similar to those on which the models were trained. **(c)** Completeness scores for each generative factor. **(d)** NRMSE on images containing unseen factor combinations. Low error indicates good zero-shot performance. $z_0, \ldots, z_4$ represent azimuth, elevation, red, green and blue generative factors respectively.

(a) **Disentanglement**

Code	c_0	c_1	c_2	c_3	c_4	c_5	**W. Avg.**
PCA	0.19	0.38	0.47	0.53	0.87	0.15	0.35
InfoGAN	0.96	0.90	0.91	0.93	0.89	0.96	0.93

(b) **Informativeness**

Code	z_0	z_1	z_2	z_3	z_4	**Avg.**
PCA	0.65	0.41	0.27	0.34	0.34	0.40
InfoGAN	0.29	0.06	0.06	0.08	0.07	0.11

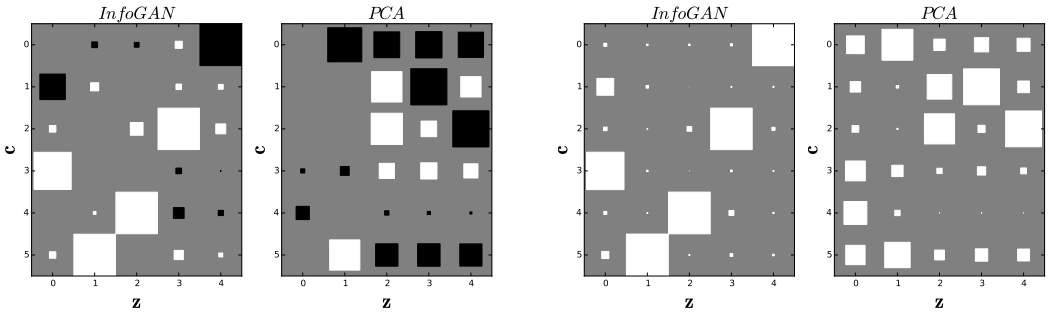
(c) **Completeness**

Code	z_0	z_1	z_2	z_3	z_4	**Avg.**
PCA	0.11	0.45	0.37	0.49	0.49	0.38
InfoGAN	0.63	0.97	0.95	0.92	0.96	0.89

(d) **Zero-shot**

Code	z_0	z_1	z_2	z_3	z_4	**Avg.**
PCA	0.71	1.12	0.86	0.87	0.81	0.87
InfoGAN	0.30	0.17	0.30	0.19	0.14	0.22

Table 2: **Random forest regression results**. Caption of Table 1 applies.



(a) **Lasso weight magnitudes** (b) **Random forest feature importances**

Figure 3: **Visualising disentanglement and completeness within learnt representations.** Positive weights/importances are represented by a white square and negative by a black square, while size indicates the magnitude. Row i illustrates the importance of c_i to each prediction and thus the disentanglement. Column j illustrates the relative importance of each code variable for predicting z_j and thus the completeness.

5 Conclusion

In this work we have presented a framework for the quantitative evaluation of disentangled representations when the ground-truth latent structure is available. The quality of learnt representations is elucidated through the explicit definition and quantification of three criteria: disentanglement, completeness and informativeness. We employed our framework to compare quantitatively the codes learned by InfoGAN to those learned by PCA. This contrastive comparison demonstrates the appropriateness of our framework, with the three criteria explaining why InfoGAN's learnt codes are superior to PCA's and the metric scores quantifying this level of superiority.

While our framework is limited to synthetic datasets where it is possible to recover $\boldsymbol{z}$, reliable disentanglement is far from solved even in this restricted setting. Hence, we believe our framework and its constituent metrics take a substantial and important step forward in understanding learned representations. We will make the code and dataset publicly available on acceptance of this paper, and hope this facilitates further model comparisons and eventually the establishment of quantitative benchmarks for disentangled factor learning. With many models and datasets, reporting detailed results for both regressors may be quite cluttered. In such cases, the average criteria scores with the random forest regressor may be an adequate summary to report, with its higher capacity making informativeness and disentanglement more distinct criteria. While we have focused on image data in this work, future work may explore the applicability of our framework to other types of synthetic data.

6 Acknowledgements

Removed to preserve anonymity.

REFERENCES

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

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems 29*, pp. 2172–2180, 2016.

Brian Cheung, Jesse A Livezey, Arjun K Bansal, and Bruno A Olshausen. Discovering hidden factors of variation in deep networks. *arXiv preprint arXiv:1412.6583*, 2014.

Taco S Cohen and Max Welling. Transformation properties of learned visual representations. *arXiv preprint arXiv:1412.7659*, 2014a.

Taco S Cohen and Max Welling. Learning the irreducible representations of commutative lie groups. In *International Conference on Machine Learning*, pp. 1755–1763, 2014b.

Emily Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. *arXiv preprint arXiv:1705.10915*, 2017.

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

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 513–520, 2011.

Ian Goodfellow, Honglak Lee, Quoc V Le, Andrew Saxe, and Andrew Y Ng. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems 22*, pp. 646–654, 2009.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pp. 2672–2680, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic Visual Concepts with a Constrained Variational Framework. In *5th International Conference on Learning Representations*, 2017.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, 2001.

Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1413–1421, 2015.

Theofanis Karaletsos, Serge Belongie, and Gunnar Rätsch. Bayesian representation learning with oracle constraints. *arXiv preprint arXiv:1506.05011*, 2015.

Jyri J Kivinen and Christopher K. I. Williams. Transformation equivariant Boltzmann machines. In *International Conference on Artificial Neural Networks*, pp. 1–9. Springer, 2011.

Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems 28*, pp. 2539–2547, 2015.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, pp. 1–101, 2016.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 991–999, 2015.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, London, 1979.

Pol Moreno, Christopher K. I. Williams, Charlie Nash, and Pushmeet Kohli. Overcoming occlusion with inverse graphics. In *ECCV Geometry Meets Deep Learning Workshop 2016*, pp. 170–185. Springer, 2016.

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

William F Whitney, Michael Chang, Tejas Kulkarni, and Joshua B Tenenbaum. Understanding visual concepts with continuation learning. *arXiv preprint arXiv:1602.06822*, 2016.

H. H. Yang and S-i. Amari. Adaptive On-Line Learning Algorithms for Blind Separation—Maximum Entropy and Minimum Mutual Information. *Neural Computation*, 9(7):1457–1482, 1997.

Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems 28*, pp. 1099–1107, 2015.

Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *Advances in Neural Information Processing Systems 27*, pp. 217–225, 2014.

A EXPERIMENTAL SETUP

A.1 INFOGAN

Extending the GAN of Goodfellow et al. (2014), InfoGAN (Chen et al., 2016) splits the generator input into two parts; 'incompressible noise' and 'latent codes' which target salient semantic features of the data. The manner in which the generator may use these latent codes is constrained by adding a regularisation term to the GAN objective, representing the mutual information between the latent codes and generated images. For stability, we use the training objective of the improved Wasserstein GAN (IWGAN) (Gulrajani et al., 2017) and the 64×64 ResNet architecture described in the open source implementation of Gulrajani et al. (2017)[3]. We modify this implementation to add InfoGAN's variational regularisation of mutual information to the IWGAN training objective, splitting the generator input into noise and latent code components before implementing the auxiliary network Q as in (Chen et al., 2016). The network Q parametrises the approximate posterior over latent codes $Q(\boldsymbol{c}|\boldsymbol{x})$, with $Q(\boldsymbol{x})$ returning a mean and standard deviation for each continuous (normal) latent code in the factorised posterior $Q(\boldsymbol{c}|\boldsymbol{x})$. Thus, to retrieve the (most likely) representation or code $\boldsymbol{c}$ for a given image $\boldsymbol{x}$, we simply take the means returned by $Q(\boldsymbol{x})$. Q shares all convolutional layers with the discriminator or 'critic' D, each adding their own final output layer. All hyperparameters of the IWGAN implementation remain unchanged while we found setting the mutual information coefficient $\lambda = 8$ to be sufficient, ensuring that the mutual information penalty was on the same scale as the unbounded WGAN objectives. For all experiments, we use 6 continuous latent codes and 128 noise variables resulting in a generator input with dimension 134. Further details on the architecture, hyperparameters and combined training objective are provided in our open-source implementation, to be made publicly available on acceptance of this paper.

$\boldsymbol{D}/\boldsymbol{Q}$	$\boldsymbol{G}$
Input 64×64×3	Input 133
3×3 64 conv.	FC 4·4·8·64
LN, ReLU, 3×3 64 conv	BN, ReLU, 3×3 512 conv ↑
LN, ReLU, 3×3 128 conv, pool	BN, ReLU, 3×3 512 conv
LN, ReLU, 3×3 128 conv	BN, ReLU, 3×3 256 conv ↑
LN, ReLU, 3×3 256 conv, pool	BN, ReLU, 3×3 256 conv
LN, ReLU, 3×3 256 conv	BN, ReLU, 3×3 128 conv ↑
LN, ReLU, 3×3 512 conv, pool	BN, ReLU, 3×3 128 conv
LN, ReLU, 3×3 512 conv	BN, ReLU, 3×3 64 conv ↑
LN, ReLU, 3×3 512 conv, pool	BN, ReLU, 3×3 64 conv
$\boldsymbol{D}$: FC 1 / $\boldsymbol{Q}$: FC 133·2	BN, ReLU, 3×3 3 conv, tanh

Table 3: **InfoGAN architecture.**

A.2 REGRESSORS

As performance generally improves with the number of trees n in the ensemble, we fix $n = 10$ for the random forest regressor. All other parameters and hyperparameters are fit to a validation set.

B GENERATIVE FACTORS VS. LATENT CODES WITH GENERIC PRIORS

Figure 4 plots each generative factor against the corresponding 'most important' InfoGAN code variable, as indicated by the lasso regression weight magnitudes and random forest feature importances. Information about each unwrapped generative factor (z_1, z_2, z_3, z_4) is linearly-encoded in single code variables $(c_5, c_4, c_2, c_0$ respectively). In contrast, distinct information about the azimuth (z_0) is non-linearly encoded across c_1 and c_3, reinforcing the argument that models with generic priors cannot be expected to learn the most complete and explicit representation of topologically distinct factors of variation. Furthermore, when InfoGAN was retrained instead with 10 code variables, it used three of these to capture the azimuth (see Figure 5a), each resembling (scaled) sine and cosine functions (see Figure 5b).

[3] `https://github.com/igul222/improved_wgan_training`

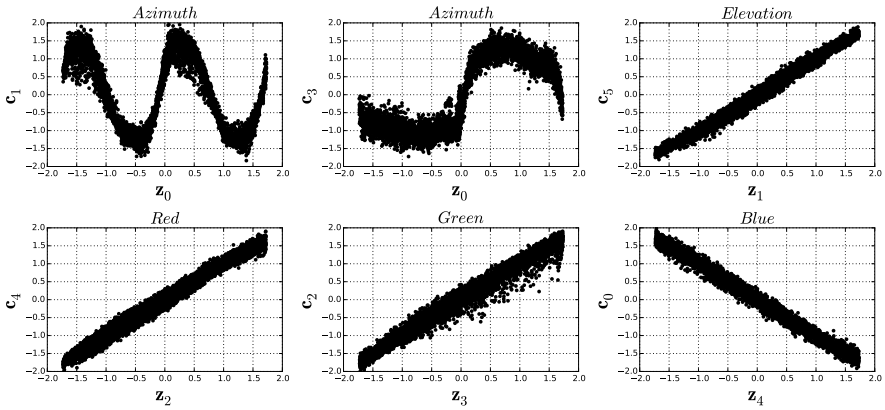
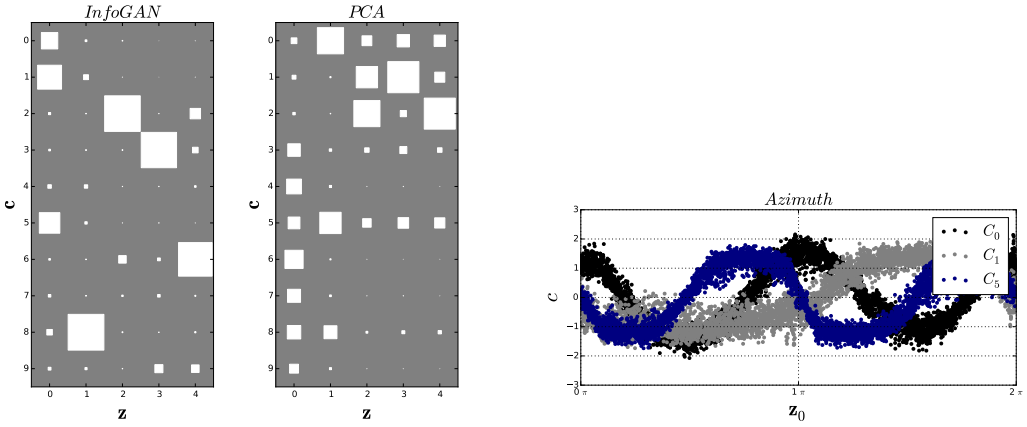


Figure 4: **Learnt codes vs. generative factors.** Depicted is the relationship between each generative factor and the corresponding 'most important' InfoGAN code variable.

C $\mathbf{D = 10, K = 5}$

As shown in Figure 5a, several 'dead' or redundant code variables (c_4, c_7, c_9) enable a high degree of completeness in $\boldsymbol{c}-$InfoGAN. Although $\boldsymbol{c}-$InfoGAN captures z_0 (azimuth) with 3 code variables, $\boldsymbol{c}-$PCA uses a least 7. Figure 5b shows that the 3 InfoGAN code variables which capture azimuth resemble scaled versions of sine and cosine functions. While the regression results for InfoGAN with 10 latent codes were inferior, Figure 5a bodes well for InfoGAN's ability to learn disentangled and interpretable codes without any prior knowledge about the number of underlying factors of variation (i.e. when trained with excessive latent codes). We note that further hyperparameter searches and random runs may have yielded better results.

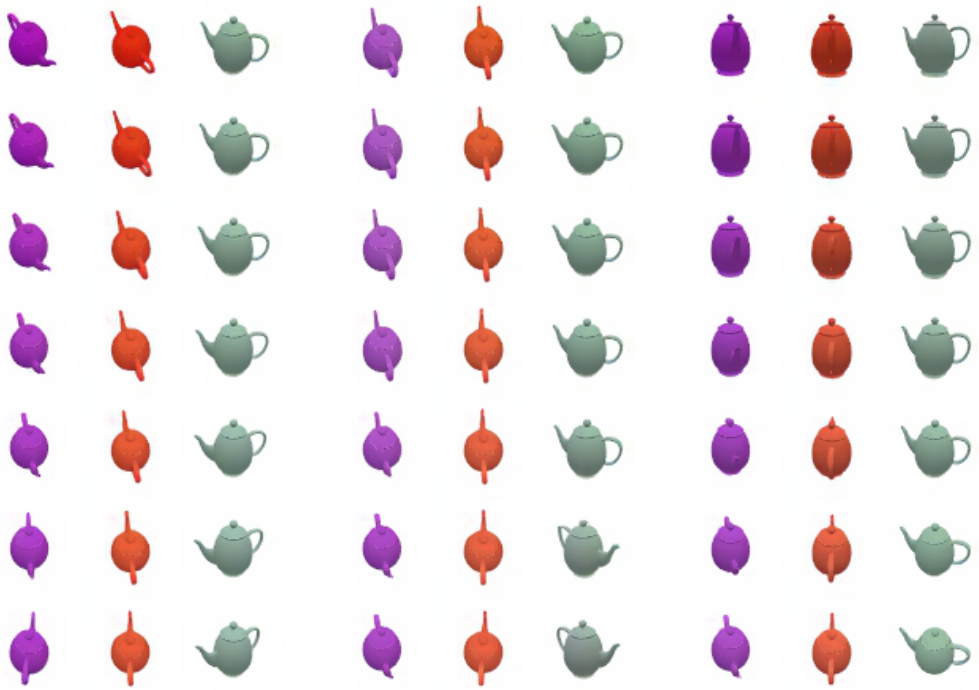


(a) **Random forest feature importances**

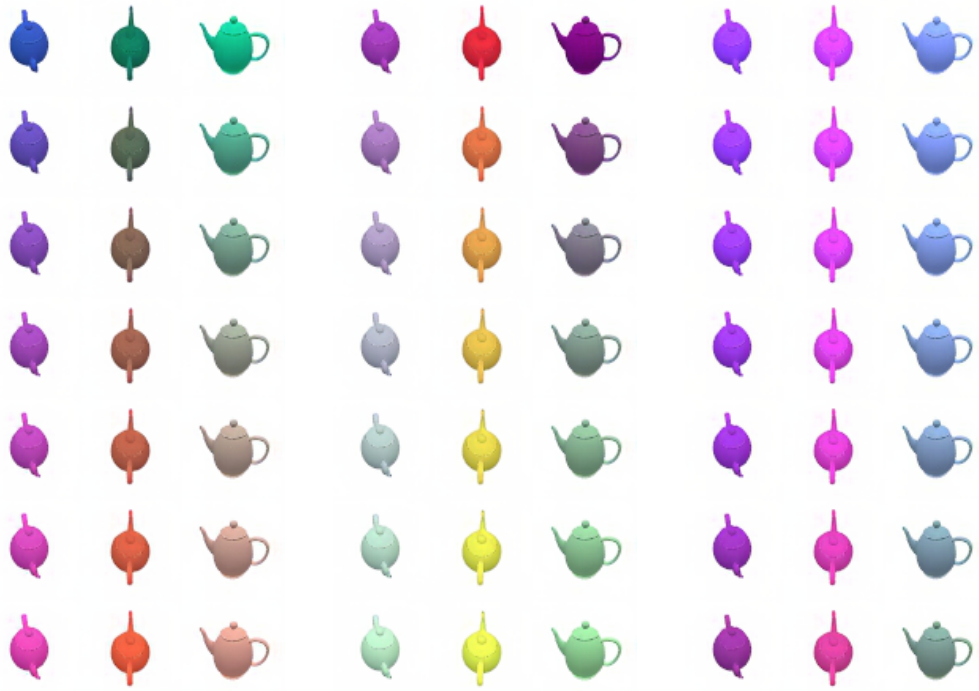
(b) **Azmiuth vs. InfoGAN code variables**

Figure 5: **Visualising the degree of disentanglement and completeness within learnt representations.**

D VISUALLY ASSESSING DISENTANGLEMENT



(a) **Azimuth(c_1)** (b) **Azimuth(c_3)** (c) **Elevation(c_5)**



(d) **Red(c_4)** (e) **Green(c_2)** (f) **Blue(c_0)**

Figure 6: **Manipulating the latent codes.** Each subfigure column represents a different random sample (or initialisation) of $\boldsymbol{c}$. For each random sample, c_i is varied from -1 (bottom) to 1 (top) to show the effect on generated images. There appears to be a high degree of disentanglement as each subfigure contains a single type of semantic variation.