A FRAMEWORK FOR THE QUANTITATIVE EVALUATION OF DISENTANGLED REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent AI research has emphasised the importance of learning *disentangled* representations of the explanatory factors behind data. Despite the recent focus on models which can learn such representations, visual inspection remains the primary method for evaluating the degree of disentanglement achieved. 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 quantitatively evaluating the quality of disentangled representations learned by different models. Three criteria are explicitly defined and quantified to elucidate the quality of learnt representations and compare models on an equal basis. Experiments with the recent InfoGAN model (Chen et al., 2016) for learning disentangled representations illustrate the appropriateness of the framework and provide a baseline for future work.

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., 2016). Such an understanding is crucial to reasoning about data with unseen factor combinations—a task on which humans are currently far superior to state-of-the-art AI models. As argued in several notable works (Desjardins et al., 2012; Bengio et al., 2013; Higgins et al., 2016; Chen et al., 2016), 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, 2014a; Kulkarni et al., 2015; Higgins et al., 2016; Chen et al., 2016). 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., 2016; Denton & Birodkar, 2017), visual inspection remains the standard evaluation metric. While the work of Higgins et al. (2016) partially addresses this issue (as discussed in section 2) and various definitions have implied additional desiderata like interpretability and invariance (Desjardins et al., 2012; Bengio et al., 2013; Cohen & Welling, 2014b; Kulkarni et al., 2015; Chen et al., 2016), current research generally lacks a clear metric for quantitatively evaluating and comparing disentangled representations. In this work we propose a framework to quantitatively evaluate disentangled representations. To elucidate the quality of learnt representations and compare models on an equal basis, desiderata of disentangled representations are *explicitly* defined and *quantified*. These 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 use it to quantitatively evaluate the representations learned by information maximizing generative adversarial networks (InfoGAN) (Chen et al., 2016).

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 work. Finally, we describe the synthetic dataset and InfoGAN model specifics before presenting the experimental results.

2 THEORETICAL FRAMEWORK

Models for disentangled factor learning seek a compact data representation or *code* which consists of disentangled and interpretable latent variables. For graphics-generated data, the K-dimensional generative factors $\boldsymbol{z}$ are designed to be an ideal such representation. Thus, the ideal disentangled code $\boldsymbol{c}^*$ should be some permutation of $\boldsymbol{z}$. That is, $\boldsymbol{c}^* = f^*(\boldsymbol{z})$, where f^* is a generalised permutation matrix (monomial matrix[1]). As $f^{*^{-1}} = f^{*T}$, we can instead write $\boldsymbol{z} = f^{*T}(\boldsymbol{c}^*)$ to interpret the monomial matrix f^{*T} as a regressor which predicts $\boldsymbol{z}$ from $\boldsymbol{c}^*$. For notational simplicity, we now use f^* (rather than f^{*T}) to denote the ideal regressor, i.e., a monomial matrix. 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 f^* 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 $f_j(\boldsymbol{c}) : \mathbb{R}^D \rightarrow \mathbb{R}^1$, where D is the dimensionality of $\boldsymbol{c}$. 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 contain information about a given generative factor. 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; Cohen & Welling, 2014a; Kulkarni et al., 2015; Higgins et al., 2016; Chen et al., 2016), namely *disentanglement*, *informativeness* and *completeness*.

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. Disentanglement implies invariance to all but one generative factor and distinguishes genuinely disentangled representations from those that are just statistically independent[2]. The disentanglement score D_i of code variable c_i is quantified by $D_i = 1 - H_K(P_{i*})$, where $H_K(P_{i*}) = -\sum_{j=0}^{K-1} P_{ij} \log_K P_{ij}$ denotes the entropy, $P_{ij} = |W_{ij}| / \sum_{k=0}^{K-1} |W_{ik}|$ denotes the 'probability' of a weight between c_i and z_j and $|W_{ij}|$ denotes the magnitude of the weight used to scale c_i when 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.

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 a representation or *code* $\boldsymbol{c}$ about a given 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 $\hat{z}_j = f_j(\boldsymbol{c})$.

[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.

[2] E.g. in a Gaussian factor analysis model we may well obtain a rotated version of the true generative factors due to the spherical symmetry of the prior (the rotation of factors problem).

Completeness. The degree to which the underlying factors of variation are captured with a representation of equal dimensionality. The completeness score C_j for generative factor z_j is quantified by $C_j = 1 - H_D(\tilde{P}_{*j})$, where $H_D(\tilde{P}_{*j}) = -\sum_{i=0}^{D-1} \tilde{P}_{ij} \log_D \tilde{P}_{ij}$ denotes the entropy and $\tilde{P}_{ij} = |W_{ij}| / \sum_{d=0}^{D-1} |W_{dj}|$ denotes the 'probability' of a weight between c_i and z_j. 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.

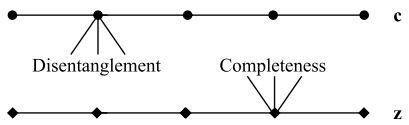


Figure 1: **Visualising disentanglement and completeness.** A one-to-one mapping between $\boldsymbol{z}$ and $\boldsymbol{c}$ is ideal. We can quantify the deviation from this ideal mapping using the disentanglement and completeness scores.

Together, the degree of disentanglement and completeness quantify the deviation from the ideal regressor f^*, a monomial matrix. While disentanglement quantifies the number of generative factors captured by a given code variable, completeness quantifies the number of code variables which capture a given generative factor. Figure 1 illustrates this idea. While interpretability is ultimately a subjective concept defined by a human arbiter, codes that are both disentangled and complete are likely to be interpretable for natural tasks like vision. The same cannot be said for tasks in a non-human domain like stock prediction, where codes that make sense to us diverge from those that best represent the underlying factors of variation (Whitney, 2016).

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 codes (representations). 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 the next section and in (Higgins et al., 2016) support this idea, with several code variables resembling non-linear functions (like the sine and cosine) of the object azimuth. 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. More specifically, we replace the lasso regression weight matrix W in previous formulae with a matrix of relative importances R, where R_{ij} denotes the relative importance of c_i in predicting z_j. 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). In addition to quantifying the deviation from f^*, the disentanglement and completeness scores clearly expose any disentangling that is done by the (non-linear) regressor itself. As performance generally improves with the number of estimators n in the ensemble, we fix $n = 10$ for simplicity and computational efficiency. All other parameters and hyperparameters are fit to a validation set.

Related Work. Higgins et al. (2016) also propose a metric to quantify the degree of disentanglement achieved by different models. In this work, an additional dataset of 'factor changes' is generated by taking the absolute difference between two latent representations corresponding to images which differ only by a change in a single generative factor. Given these changes in latent space, a linear classifier is trained to predict which generating factor caused the change between the images, with the classification accuracy quantifying the degree of disentanglement. While this metric quantifies the degree of disentanglement (latent variables must be primarily perturbed by changes in a single generative factor to achieve high classification accuracy), it does not quantify the amount of information captured about the generative factors or the completeness of the representation. By quantifying these additional criteria, our simple metrics provide a more thorough evaluation of the

disentangled representations learned by a given model, without the need to generate an additional dataset.

Several metrics have been proposed to measure various criteria often associated with disentangled representations, including equivariance, invariance and equivalence (Goodfellow et al., 2009; Lenc & Vedaldi, 2015; Cohen & Welling, 2014b;a; Jayaraman & Grauman, 2015). While these metrics focus on how the learned representations are affected by specific transformations of the input data (such as rotations and translations), our framework and constituent metrics directly evaluate the quality of disentangled representations learned on any given synthetic dataset.

3 EXPERIMENTS

To demonstrate the appropriateness of the proposed framework, we train InfoGAN (Chen et al., 2016) on a synthetic dataset and quantitatively compare the learned representations or *codes* to those learned by PCA.

3.1 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., 2016). 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.

3.2 MODEL

Driven by the idea that some form of understanding is required in order to be able to synthesise the observed examples, deep generative modelling has become one of the leading approaches to unsupervised representation learning (Bengio et al., 2013; Chen et al., 2016). It is hoped that interpretable

(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.

disentangled representations will be learned automatically by a sensible generative model. However, if the generator uses the latent representations in a highly-entangled way, individual dimensions may not correspond to semantic features of the data (Chen et al., 2016). As a result, several recent works have imposed additional learning constraints to encourage generative models to learn disentangled representations without supervision (Desjardins et al., 2012; Reed et al., 2014; Zhu et al., 2014; Cohen & Welling, 2014b; Cheung et al., 2014; Larsen et al., 2015; Makhzani et al., 2015; Chen et al., 2016; Higgins et al., 2016). Of these models, it can be argued that InfoGAN (Chen et al., 2016) and the variational autoencoder (VAE) extension of Higgins et al. (2016) are the most promising due to their scalability and lack of assumptions about the underling factors of variation[3].

Extending the GAN of Goodfellow et al. (2014), InfoGAN (Chen et al., 2016) splits the input noise vector 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., 2017a) and the 64×64 ResNet (He et al., 2016) architecture described in the open source implementation of Gulrajani et al. (2017b). We modify this implementation to add InfoGAN's variational regularisation of mutual information to the IWGAN training objective, splitting the input noise vector 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 continuous (normal) latent codes. Thus, to retrieve the (most likely) representation or code $\boldsymbol{c}$ for a given image $\boldsymbol{x}$, we simply take the mean 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 cost 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. For illustrative purposes, we show the best of 10 random runs as we found InfoGAN to be quite sensitive to random initialisation. 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.

3.3 RESULTS

Tables 1 and 2 present the results for the lasso and random forest regressors respectively. 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 (0). Hence, we report the normalised root-mean-square error (NRMSE) in these tables. Table 1a presents the test set NRMSE for images containing factor combinations similar to those on which the models were trained. The representation or *code* learned by InfoGAN ($\boldsymbol{c}-\mathrm{InfoGAN}$) clearly outperforms the code learned by PCA ($\boldsymbol{c}-\mathrm{PCA}$) in predicting each generative factor. 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, as indicated by the relatively large drop in prediction error when using the non-linear regressor (see Table 2a); (ii) InfoGAN struggles to capture enough information about the azimuth in $\boldsymbol{c}-\mathrm{InfoGAN}$, as indicated by the fact that the error in predicting the azimuth remains much higher than the rest (even with the non-linear regressor). By comparing the results of the linear regressor in Table 1a with those of the non-linear regressor in Table 2a, we see that both codes better predict the generative factors with increased capacity—especially $\boldsymbol{c}-\mathrm{PCA}$.

Tables 1b and 2b present the disentanglement scores for the lasso and random forest regressors respectively. With both regressors, the variables in $\boldsymbol{c}-\mathrm{InfoGAN}$ achieve a much higher disentanglement score than those in $\boldsymbol{c}-\mathrm{PCA}$, with each variable in $\boldsymbol{c}-\mathrm{InfoGAN}$ closer to capturing a single generative factor. That is, $\boldsymbol{c}-\mathrm{InfoGAN}$ is more *disentangled*. The high disentanglement scores of $\boldsymbol{c}-\mathrm{InfoGAN}$ in Table 2b confirm that the low NRMSEs in Table 2a were not due to any substantial disentangling done by the (non-linear) random forest regressor itself. The same cannot be said for

[3]Despite their separate inspirations (information theory and neuroscience), the learning constraints imposed by these models are closely-related. See Appendix A.

$\boldsymbol{c}-\mathrm{PCA}$, with its low disentanglement scores in Table 2b revealing that this 'tangled' code is in fact disentangled by the regressor itself to achieve lower NRMSEs. Figure 3 helps to visualise the disentanglement and identify the generative factors captured by each code variable. For example, comparing $c_0-\mathrm{PCA}$ and $c_0-\mathrm{InfoGAN}$ in figures 3a and 3b (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 (illustrated by the much higher zero-shot NRMSE in tables 1c and 2c). In particular, tables 1c and 2c show that $\boldsymbol{c}-\mathrm{PCA}$ is even outperformed by the constant regressor in predicting z_1, while $\boldsymbol{c}-\mathrm{InfoGAN}$ predicts the value of unseen factor combinations reasonably well.

Tables 1d and 2d 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. In addition, Figure 3 helps to explain some exceptions or outliers, such as the low completeness score (overcompleteness) of $\boldsymbol{c}-\mathrm{InfoGAN}$ in predicting z_0. Inspecting the figure justifies this relatively-low score, with z_0 clearly captured by a combination of $c_1-\mathrm{InfoGAN}$ and $c_3-\mathrm{InfoGAN}$. However, this is still much less overcomplete than $\boldsymbol{c}-\mathrm{PCA}$, with each of its constituent variables capturing some information about z_0 (see Figure 3b).

(a) **Test set NRMSE**

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

(b) **Disentanglement**

Code	c_0	c_1	c_2	c_3	c_4	c_5	**Avg.**
PCA	0.16	0.38	0.42	0.14	0.63	0.16	0.31
InfoGAN	0.83	0.64	0.65	0.93	0.80	0.80	0.77

(c) **Zero-shot NRMSE**

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

(d) **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

Table 1: **Lasso regression results**. **(a)** Test set NRMSE on images containing factor combinations similar to those on which the models were trained **(b)** Disentanglement scores for each dimension of $\boldsymbol{c}-\mathrm{InfoGAN}$ and $\boldsymbol{c}-\mathrm{PCA}$. **(c)** NRMSE on images containing unseen factor combinations. Low NRMSE indicates good zero-shot inference performance. **(d)** Completeness scores for each generative factor.

(a) **Test set NRMSE**

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

(b) **Disentanglement**

Code	c_0	c_1	c_2	c_3	c_4	c_5	**Avg.**
PCA	0.19	0.38	0.47	0.53	0.87	0.15	0.39
InfoGAN	0.96	0.90	0.91	0.93	0.89	0.96	0.93

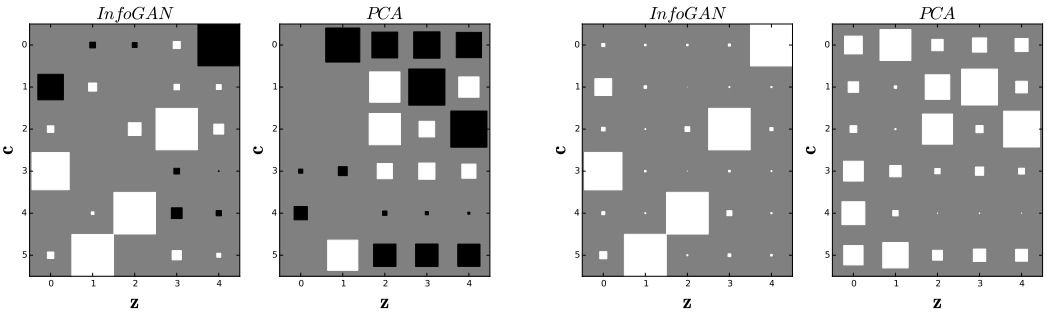
(c) **Zero-shot NRMSE**

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

(d) **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

Table 2: **Random forest regression results**. **(a)** Test set NRMSE on images containing factor combinations similar to those on which the models were trained. **(b)** Disentanglement scores for each dimension of $\boldsymbol{c}-\mathrm{InfoGAN}$ and $\boldsymbol{c}-\mathrm{PCA}$. **(c)** NRMSE on images containing unseen factor combinations. Low NRMSE indicates good zero-shot inference performance. **(d)** Completeness scores for each generative factor.



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

Figure 3: **Visualising the degree of disentanglement and completeness within learnt representations.** Positive weights are represented by a white square and negative by a black square, while the magnitude is indicated by the size. Row i illustrates the importance of c_i to each prediction. That is, the amount of information c_i captures about each generative factor and thus its disentanglement. Column j illustrates the relative importance of each code variable for predicting z_j and thus the completeness. Ideally, each row and column would contain a single (large) weight, indicating a one-to-one mapping or monomial matrix.

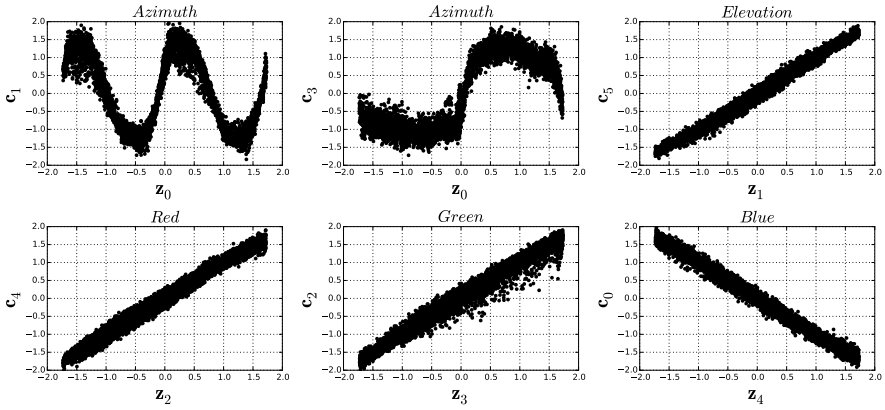


Figure 4: **Learnt codes vs. generative factors.** Depicted is the relationship between each generative factor and the corresponding 'most important' InfoGAN code variable.

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 $(z_1, \ldots, z_4)$ 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 5 in the Appendix). As depicted in Figure 5b, these variables learned similar non-linear functions, slightly resembling (scaled) 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 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.

4 CONCLUSION

In this work we have presented a framework for quantitatively evaluating the disentangled representations learned by different models. The quality of learnt representations is elucidated through the explicit definition and quantification of three criteria: disentanglement, informativeness and completeness. In addition, the promising quantitative results of InfoGAN illustrate the appropriateness of the framework and provide a baseline for future models which seek to learn disentangled representations without supervision. While we have focused on image data in this work, we hope that future work will explore the applicability of the framework to other types of synthetic data.

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

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

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*, 2017a.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein GANs implementation. `https://github.com/igul222/improved_wgan_training`, 2017b. Accessed: 2017-01-07.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Irina Higgins, Loic Matthey, Xavier Glorot, Arka Pal, Benigno Uria, Charles Blundell, Shakir Mohamed, and Alexander Lerchner. Early visual concept learning with unsupervised deep learning. *arXiv preprint arXiv:1606.05579*, 2016.

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.

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.

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 Whitney. Disentangled representations in neural models. *arXiv preprint arXiv:1602.02383*, 2016.

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

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 A RELATIONSHIP BETWEEN TWO CONSTRAINTS THAT ENCOURAGE DISENTANGLING

To make the prior over 'incompressible latent noise' $P(\boldsymbol{z})$ more explicit, the lower bound on mutual information defined by Chen et al. (2016) can be written as:

$$L_I(G,Q) = \mathbb{E}_{\boldsymbol{c}\sim P(\boldsymbol{c}),\boldsymbol{z}\sim P(\boldsymbol{z})}[\log Q(\boldsymbol{c}|G(\boldsymbol{z},\boldsymbol{c}))] + H(\boldsymbol{c}), \tag{1}$$

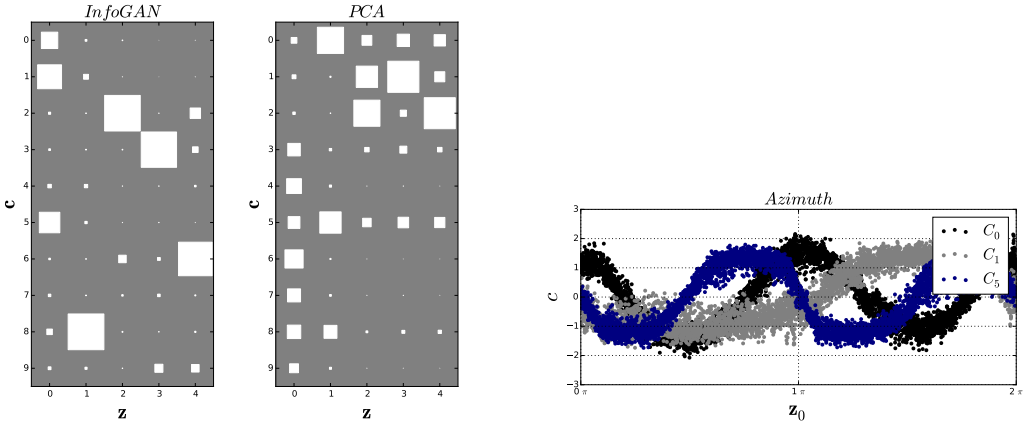
where $\boldsymbol{c}$ is the latent codes, $\boldsymbol{z}$ is the incompressible noise and $Q(\boldsymbol{c}|G(\boldsymbol{z},\boldsymbol{c}))$ is the approximate posterior over latent codes. Thus, it can be shown that maximising this lower bound on mutual information is equivalent to minimising a redundancy term similar to that of Higgins et al. (2016), which is defined as the KL divergence from the posterior over latent codes to the prior.

Proof

$$\begin{aligned}
L_I(G,Q) &= \mathbb{E}_{\boldsymbol{c}\sim P(\boldsymbol{c}),\boldsymbol{z}\sim P(\boldsymbol{z})}[\log Q(\boldsymbol{c}|G(\boldsymbol{z},\boldsymbol{c}))] + H(\boldsymbol{c}) && (2)\\
&= \int_c\int_z p(\boldsymbol{c})p(\boldsymbol{z})\log q(\boldsymbol{c}|G(\boldsymbol{z},\boldsymbol{c}))dcdz - \int_c p(\boldsymbol{c})\log p(\boldsymbol{c})dc && (3)\\
&= \int_c\int_z p(\boldsymbol{c})p(\boldsymbol{z})\log q(\boldsymbol{c}|G(\boldsymbol{z},\boldsymbol{c}))dcdz - \int_c\int_z p(\boldsymbol{c})p(\boldsymbol{z})\log p(\boldsymbol{c})dcdz && (4)\\
&= \int_c\int_z p(\boldsymbol{c})p(\boldsymbol{z})\log \frac{q(\boldsymbol{c}|G(\boldsymbol{z},\boldsymbol{c}))}{p(\boldsymbol{c})}dcdz && (5)\\
&= -\int_z p(\boldsymbol{z})\int_c p(\boldsymbol{c})\log \frac{p(\boldsymbol{c})}{q(\boldsymbol{c}|G(\boldsymbol{z},\boldsymbol{c}))}dcdz && (6)\\
&= -\mathbb{E}_{\boldsymbol{z}\sim P(\boldsymbol{z})}[D_{KL}(P(\boldsymbol{c})||Q(\boldsymbol{c}|G(\boldsymbol{z},\boldsymbol{c}))]. && (7)
\end{aligned}$$

B 2× OVERCOMPLETE LATENT CODES

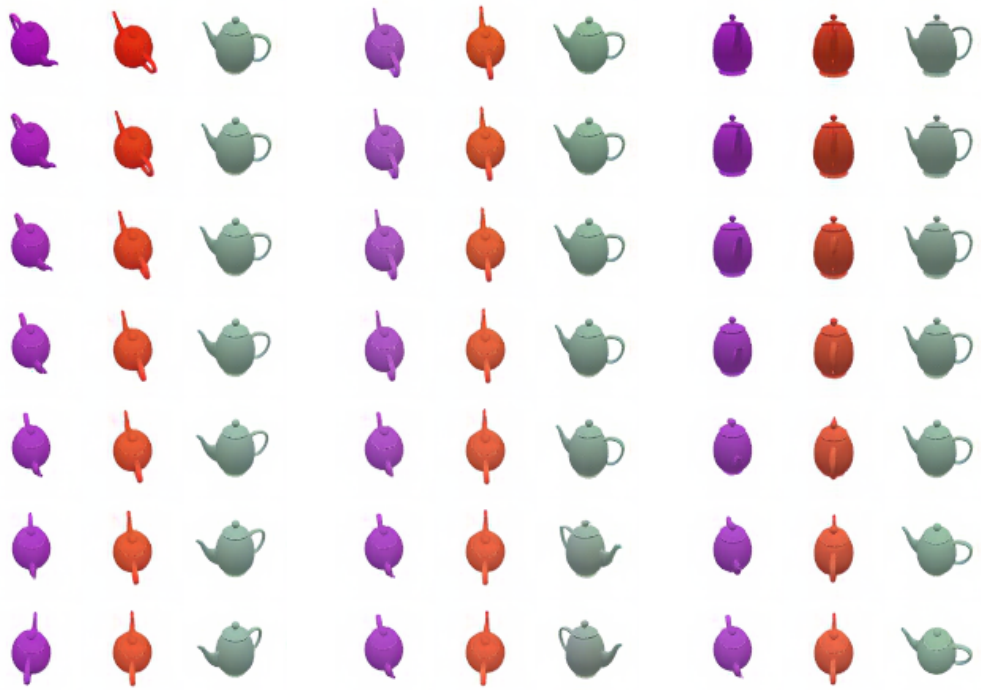
As shown in Figure 5a, several redundant code variables (c_4, c_7, c_9) enable a high degree of completeness in $\boldsymbol{c}-$InfoGAN. Although $\boldsymbol{c}-$InfoGAN captures z_0 with 3 code variables, $\boldsymbol{c}-$PCA uses a least 7. Figure 5b shows that the 3 InfoGAN code variables which capture z_0 (azimuth) resemble scaled versions of sine and cosine functions.



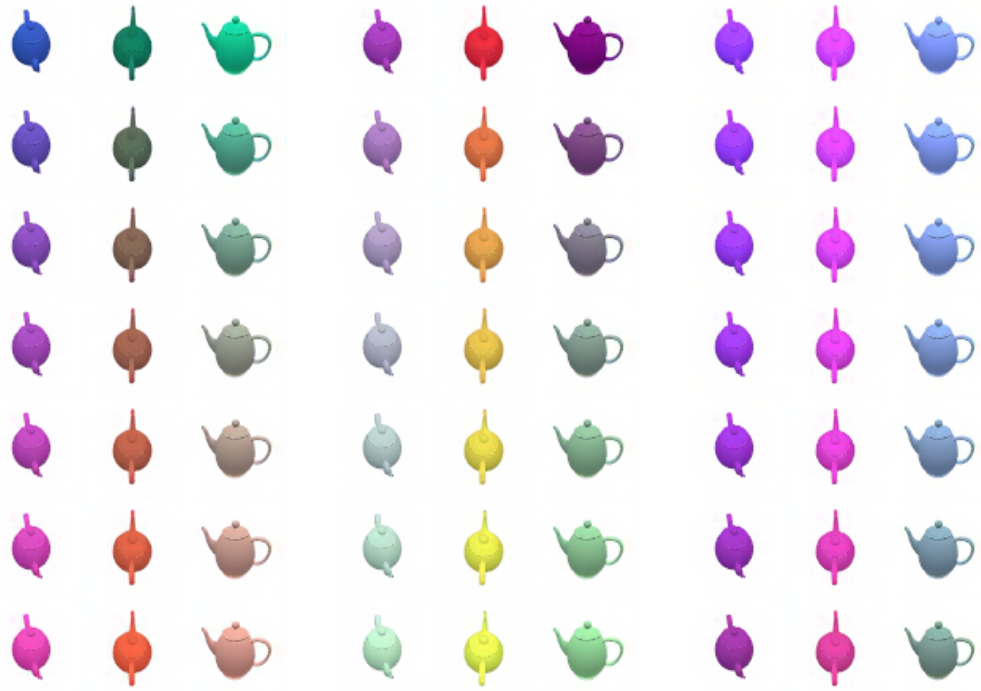
(a) **Lasso weight magnitudes** (b) **Learnt codes vs. generative factors**

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

C 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.