Overlooked Implications of the Reconstruction Loss for VAE Disentanglement Supplementary Material

Nathan Michlo , Richard Klein , Steven James
University of the Witwatersrand, Johannesburg, South Africa
nathan.michlo1@students.wits.ac.za, {richard.klein, steven.james}@wits.ac.za

A Identifying Factor Importance

A factor in a dataset is considered more important if a VAE prefers to learn it before another factor. Burgess *et al.* [2017] identify this order of importance through a slow increase of the information capacity of VAEs during training. We note that simply by looking at the average perceived distance between observations along factor traversals, this ordering can be determined. Factors with a greater average distance will minimise the error in the reconstruction loss due to random sampling the most when learnt first. These factors (or components thereof) will thus generally be preferred.

To compute the average perceived distance along a factor f, we sample a ground-truth coordinate vector $\boldsymbol{y}^{(a)} \in \mathcal{Y}$ and then another random different coordinate vector $\boldsymbol{y}^{(b)} \in \mathcal{Y}^{(a,i)}$ over the traversal for factor f passing through $\boldsymbol{y}^{(a)}$. Note that $\boldsymbol{y}^{(a)} \neq \boldsymbol{y}^{(b)}$. Then, we compute the perceived distance between the corresponding observations $\mathrm{d}_{\mathrm{pcv}}(\boldsymbol{x}^{(a)}, \boldsymbol{x}^{(b)})$. We repeat this process to compute the expected perceived distance along factor traversals, given by Equation (1).

$$d_i = \mathbb{E}_{a \in \mathcal{Y},\ b \in \mathcal{Y}^{(a,i)},\ a \neq b} \left[\mathrm{d}_{\mathrm{pcv}}(\boldsymbol{x}^{(a)}, \boldsymbol{x}^{(b)}) \right] \tag{1}$$

We determine the factor importance for dSprites as: $d_{\mathrm{x}} \approx 0.058$ and $d_{\mathrm{y}} \approx 0.057$ position, then $d_{\mathrm{scale}} \approx 0.025$, then $d_{\mathrm{shape}} \approx 0.022$, and finally $d_{\mathrm{orientation}} \approx 0.017$. This aligns with the order determined by Burgess *et al.* [2017]. Computing estimates over an entire dataset can be intractable—for our estimates, we sample at least 50000 pairs per factor.

Additionally, we compute the average perceived distance between any random pairs in the datasets (see Equation (2)) and find that the average distance is higher. For dSprites specifically, we have $d_{\mathrm{ran}} \approx 0.075$. This suggests that the ground-truth factors correspond to axes in the data that minimise the reconstruction loss and is further evidence as to why VAEs appear to learn disentangled results.

$$d_{\mathrm{ran}} = \mathbb{E}_{a \in \mathcal{Y},\ b \in \mathcal{Y},\ a \neq b} \left[\mathrm{d}_{\mathrm{pcv}}(\boldsymbol{x}^{(a)}, \boldsymbol{x}^{(b)}) \right] \tag{2}$$

A.1 Factor Importance Results

In Appendix A, we relate our work to Burgess *et al.* [2017] by estimating the importance of different factors over the dSprites [Matthey *et al.*, 2017] dataset using the reconstruction loss (MSE) as the perceived distance function between observation pairs.

We compute and list the order of importance of factors from the remaining datasets in Table 1. These importance values are computed as the average perceived distances between 50000 randomly sampled observation pairs taken along random factor traversals. Factors with higher average perceived distances will be prioritised by the model. For comparison, the average distance between any random pair in the dataset is also given. The average distances between pairs along factor traversals are usually less than the random distance, indicating that the ground-truth factors usually correspond to axes in the data that minimise errors.

Dataset	Factor	Mean Dist.	Dist. Std.
Cars3D	**random**	0.0519	0.0188
	azimuth	0.0355	0.0185
	object type	0.0349	0.0176
	elevation	0.0174	0.0100
3D Shapes	**random**	0.2432	0.0918
	wall hue	0.1122	0.0661
	floor hue	0.1086	0.0623
	object hue	0.0416	0.0292
	shape	0.0207	0.0161
	scale	0.0182	0.0153
	orientation	0.0116	0.0079
Small NORB	**random**	0.0535	0.0529
	lighting	0.0531	0.0563
	category	0.0113	0.0066
	rotation	0.0090	0.0071
	instance	0.0068	0.0048
	elevation	0.0034	0.0030
dSprites	**random**	0.0754	0.0289
	position y	0.0584	0.0378
	position x	0.0559	0.0363
	scale	0.0250	0.0148
	shape	0.0214	0.0095
	orientation	0.0172	0.0106
XYSquares	**random**	0.0308	0.0022
	y (R, G, B)	0.0104	0.0000
	x (R, G, B)	0.0104	0.0000

Table 1: Average perceived distances sampled along random factor traversals for different datasets. Components of factors with higher average distances will usually be prioritised by the model.

We visualise the distribution of distances along factor traversals using cumulative frequency plots as in Figure 1. It is

interesting to note the distinct shift in structure for the adversarial XYSquares dataset, since distance values are constant depending on the number of differing factors.

B Implementation Details

In this section, we describe our various implementation details of the β-VAE [Higgins *et al.*, 2016] and Ada-GVAE [Locatello *et al.*, 2020] frameworks, as well as the handling and standardisation of the different ground-truth datasets.

B.1 Beta Normalisation

For general consistency across datasets with different numbers of channels and models with different numbers of latent units, we implement beta normalisation as described by Higgins *et al.* [2016].

Instead of taking the sum over the KL divergence in the regularisation term and the sum over elements in the reconstruction term of the VAE loss, we instead compute the means over elements in both terms and adjust the β value accordingly.

B.2 Symmetric KL

The original Ada-GVAE implementation uses the asymmetric KL divergence $D_{\mathrm{KL}}\left(p \mid\mid q\right)$ as the distance function between the corresponding latent units of observation pairs. The Ada-GVAE uses this distance measure to estimate which of these latent distributions should be averaged together.

We instead follow the approach of Dittadi *et al.* [2021] and use the symmetric KL divergence to compute these distances between latent units, improving the averaging procedure and computation of the threshold. The symmetric KL divergence is defined in Equation (3).

$$\tilde{D}_{\mathrm{KL}}\left(p,\ q\right) = \frac{1}{2} D_{\mathrm{KL}}\left(p \mid\mid q\right) + \frac{1}{2} D_{\mathrm{KL}}\left(q \mid\mid p\right) \qquad (3)$$

B.3 Sampling Ada-GVAE Pairs

The Ada-GVAE [Locatello *et al.*, 2020] framework introduces weak supervision by sampling pairs of observations such that there are always $k \in [1, \mathrm{F}]$ differing factors between them, where F is the total number of factors generating the dataset. We use the weaker but more realistic case for sampling each pair, where k is sampled uniform randomly from the range $[1, \mathrm{F}]$ as described in the original paper.

B.4 Dataset Standardisation

For improved consistency and training performance, dataset observations are standardised. We first resize the observations to a width and height of 64×64 pixels using bilinear filtering if needed. Then the observations are normalised such that on average each channel of the image has a mean of 0 and a standard deviation of 1. Normalisation constants for each channel are precomputed across the entire dataset and are given in Table 2.

Dataset	Mean	Std
Cars3D	**R**: 0.897667614997663 **G**: 0.889165802006751 **B**: 0.885147515814868	0.225031955315030 0.239946127898126 0.247921063196844
3D Shapes	**R**: 0.502584966788819 **G**: 0.578759756608967 **B**: 0.603449973185958	0.294081404355556 0.344397908751721 0.366168598152475
Small NORB	0.752091840108860	0.095638790168273
dSprites	0.042494423521890	0.195166458806261
XYSquares	**R**: 0.015625 **G**: 0.015625 **B**: 0.015625	0.124034734589209 0.124034734589209 0.124034734589209

Table 2: Precomputed channel-wise normalisation constants for datasets, assuming values of the input data are in the range $[0, 1]$.

C Experiment Hyper-Parameters

In this section, we give further details on the experiments conducted throughout the paper and their chosen hyper-parameters. For easier comparison with prior work, we use similar hyper-parameters, optimiser and model choices to Higgins *et al.* [2016]; Kim and Mnih [2018]; Locatello *et al.* [2019].

C.1 Model Architecture

We use similar convolutional encoder and decoder models as Higgins *et al.* [2016]. A full description of the basic VAE architecture is given in Table 3. The Gaussian encoder parameterises the mean and log variance of each latent distribution. The decoder uses the Gaussian derived Mean Squared Error (MSE) as the loss function. The number of input channels the encoder receives and the number of output channels the decoder produces depends on the dataset the model is trained on, this is either 1 or 3 channels.

Encoder		
Input	{1 **or** 3}x64x64	
Conv.	32x4x4	(*stride 2*, ReLU)
Conv.	32x4x4	(*stride 2*, ReLU)
Conv.	64x4x4	(*stride 2*, ReLU)
Conv.	64x4x4	(*stride 2*, ReLU)
Linear	256	(ReLU)
2x Linear	{9 **or** 25}	
Decoder		
Input	{9 **or** 25}	
Linear	256	(ReLU)
Linear	1024	(*reshape 64x4x4*, ReLU)
Upconv.	64x4x4	(*stride 2*, ReLU)
Upconv.	32x4x4	(*stride 2*, ReLU)
Upconv.	32x4x4	(*stride 2*, ReLU)
Upconv.	{1 **or** 3}x4x4	(*stride 2*)

Table 3: VAE encoder and decoder architectures. The model's inputs and outputs change based on the number of channels in the dataset, while the number of latent units the model has depends on the experiment hyper-parameters.

C.2 Optimiser And Batch Size

Models are trained using the Adam [Kingma and Ba, 2015] optimiser with a learning rate of 10^{-3}. A batch size of 256 is used in the case of the β-VAE [Higgins *et al.*, 2016]. Similarly, in the case of the weakly-supervised Ada-GVAE [Locatello *et al.*, 2020], 256 observation pairs are sampled per batch using the strategy from Appendix B.3.

C.3 Experiment Sweeps

Experiment plots and results are all produced from models trained over grid searches of hyper-parameters. Grid search values are given in Table 4. If values are not specified in the hyper-parameter sweep, then default values from the corresponding section of the experiment or supplementary material are used.

C.4 Total Compute

We estimate that approximately $\sim$ 1040 hours of compute across a computing cluster have been used to train the models needed to generate the plots and results presented throughout this paper.

Due to the inherent high variance of unsupervised VAE results, multiple runs using the same hyper-parameters but different random seeds are needed for comparing frameworks [Locatello *et al.*, 2019]. This susceptibility of unsupervised methods to the starting random seed makes extended comparisons between frameworks prohibitive due to the computational cost.

References

Christopher Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. In *Workshop on Learning Disentangled Representations at the 31st Conference on Neural Information Processing Systems,*, 2017.

Andrea Dittadi, Frederik Träuble, Francesco Locatello, Manuel Wuthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer, and Bernhard Schölkopf. On the transfer of disentangled representations in realistic settings. In *International Conference on Learning Representations*, 2021.

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.

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

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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 *International Conference on Machine Learning*, pages 4114–4124. PMLR, 2019.

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

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dSprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

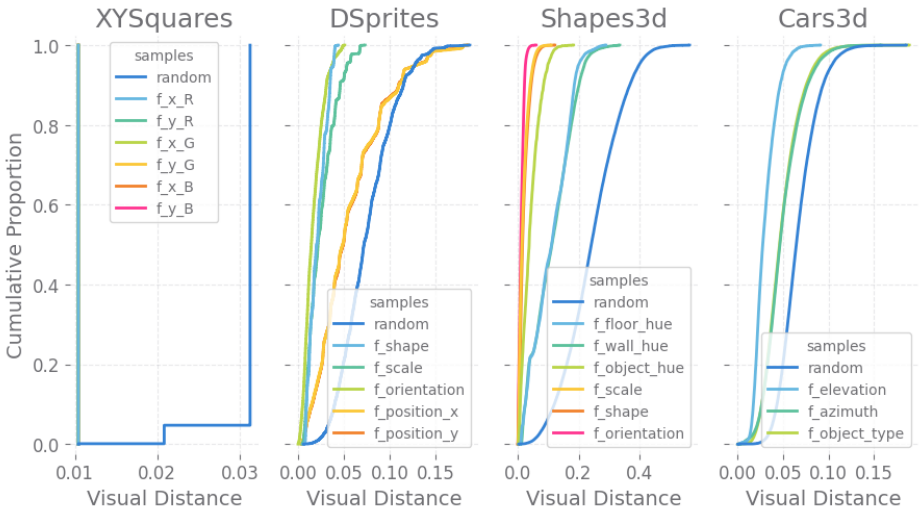

Figure 1: Cumulative proportion of perceived distance values between pairs sampled along factor traversals, compared to perceived distances between random pairs. Factors which are more important for the VAE to learn first to minimise the reconstruction loss have higher average perceived distances (lines shifted further to the right). This corresponds to the experimental results from Burgess *et al.* [2017] which show that as the information capacity of a VAE is increased, it learns factors in order. For dSprites, this is x and y position, followed by scale, then shape, and finally orientation.

Experiment	Total	Hyper-Parameters
5.3. Adversarial Experiments (Figure 6)	$8 \times 2 \times 2 \times 5 = 160$ $\times 1$ repeats $= 160$ $\times \sim$ 4h $\approx$ 640h	train steps $= 115200$ beta $(\beta) \in \{0.000316, 0.001, 0.00316, 0.01, 0.0316, 0.1, 0.316, 1.0\}$ framework $\in \{\beta$-VAE, Ada-GVAE$\}$ latents $(\mathrm{D}) \in \{9, 25\}$ dataset $\in \{$dSprites, 3D Shapes, Cars3D, Small NORB, XYSquares$\}$
5.4. Varying Levels of Overlap (Figure 9)	$2 \times 2 \times 8 = 32$ $\times 5$ repeats $= 160$ $\times \sim$ 2h $\approx$ 320h	train steps $= 57600$ beta $(\beta) \in \{0.001, 0.00316\}$ framework $\in \{\beta$-VAE, Ada-GVAE$\}$ latents $(\mathrm{D}) = 9$ dataset $=$ XYSquares grid spacing $\in \{8, 7, 6, 5, 4, 3, 2, 1\}$
6.1. Augmented Loss Experiments (Figure 11)	$2 \times 2 \times 2 = 8$ $\times 5$ repeats $= 40$ $\times \sim$ 2h $\approx$ 80h	train steps $= 57600$ beta $(\beta) \in \{0.0001, 0.0316\}$ framework $\in \{\beta$-VAE, Ada-GVAE$\}$ latents $(\mathrm{D}) = 25$ dataset $=$ XYSquares recon. loss $\in \{$MSE, BoxBlurMSE$\}$ box blur radius $= 31$ (63x63 in size) box blur weight $= 63^2 = 3969$

Table 4: Grid search hyper-parameters used for the different experiments throughout this paper.