Reproducability Challenge: β-VAE Group Report

Dominik Koller, Anonymous Member

April 2022

Abstract

The paper we have chosen has two main contributions. The first introduces a hyperparameter β to the VAE architecture to encourage disentangled latent representations. The second, is a quantification of disentanglement between the dimensions of the learned latent space. We successfully reproduce both these contributions. We also extend the work with a new investigation, into the authors' hypothesis that β-VAEs may improve transfer learning performance. Our results do not show a significant improvement in transfer learning performance. We discuss the significance of our results and limitations in our method, which may stem from our dataset choice and limited computational resources. Finally, we suggest methods for investigating the transfer learning hypothesis with more computational resources.

Contents

1 Motivation

We have chosen this paper for three reasons. Firstly, we believe variational autoencoders (VAEs) capture some of the most exciting ideas in current ML research. They provide highly non-linear dimensionality reduction of high-dimensional data, which can be seen as a method for learning abstract concepts in an unsupervised manner. They also provide an unsupervised way of approximating a distribution over any set of high-dimensional data, only requiring a differentiable learning architecture to and from a latent space; this seemed to us extremely powerful and worthy of exploration. Secondly, we believe that human interpretability of learned models is one of the most important problems to work on in machine learning today. Modern "black-box" learners like most deep neural networks are frequently applied in influential positions. The creation of interpretable models that can replace black-box models without sacrificing performance will be essential to ensure reduce bias, ensure robustness and provide accountability. There has therefore recently been dramatically increasing interest in interpretable models [LPK20].

The goal of beta-VAE is not only to provide a latent space that can be sampled from as VAEs already do, but also to provide disentangled dimensions in the latent space. This is an important step towards human interpretability: it enables interpreting a particular dimension in latent space as a particular generating variable in the (usually unknown) generating distribution. Thirdly, we saw in this paper an opportunity not only to reproduce the given results, but also for testing the hypothesis given at the very end of the paper suggesting that a disentangled latent space might allow for easier transfer learning (which we suggest can be seen as machine interpretability of the latent space).

2 Theoretical framework β-VAE

This section is intended to clarify the corresponding section in the original paper [Hig+17]. We largely follow their reasoning and notation, with small modifications which we point out and justify.

Given a dataset, we assume that all datapoints $\mathbf{x} \in \mathbb{R}^N$ are *i.i.d.* according to an underlying distribution D. A generative model gives a *p.d.f.* $p_\theta(\mathbf{x}) \mapsto [0, 1]$ that predicts $\mathbb{P}(\mathbf{x})$, where θ are the model parameters. We also want to be able to sample from p_θ. Following a maximum likelihood approach, we maximise the expected likelihood as a function of the model parameters:

$$\underset{\theta}{\operatorname{argmax}} \quad \mathbb{E}_{x\sim D}[p_\theta(\mathbf{x})] \tag{1}$$

Now the β-VAE framework assumes that $\mathbf{x}$ is generated from a set of ground truth generative factors $\mathbf{v} \in \mathbb{R}^K$ and $\mathbf{w} \in \mathbb{R}^H$, where $\mathbf{v}$ are conditionally independent, that is $p(\mathbf{v}|\mathbf{x}) = \prod_k p(v_k|\mathbf{x})$. It aims at learning the joint distribution

of $\mathbf{x} \sim D$ and latent variables $\mathbf{z} \in \mathbb{R}^M$ for $M \geq K$, which should capture the conditionally independent generative factors $\mathbf{v}$ in a *disentangled* manner, and capture $\mathbf{w}$ in the remaining dimensions of $\mathbf{z}$. From (1) and the law of total expectation we get

$$\underset{\theta}{\operatorname{argmax}} \quad \mathbb{E}_{x \sim D}\big[\mathbb{E}_{z \sim p_\theta|x}[p_\theta(\mathbf{x}|\mathbf{z})]\big] \tag{2}$$

Since we do not have access to $\mathbf{z} \sim p_\theta|\mathbf{x}$, we use another approximation, $\mathbf{z} \sim q_\phi|\mathbf{x}$, which we approximate again using a maximum likelihood approach:

$$\begin{aligned} & \underset{\theta,\phi}{\operatorname{argmax}} \quad \mathbb{E}_{x \sim D}\big[\mathbb{E}_{z \sim q_\phi|x}[p_\theta(\mathbf{x}|\mathbf{z})]\big] \\ = \quad & \underset{\theta,\phi}{\operatorname{argmax}} \quad \mathbb{E}_{x \sim D}\big[\mathbb{E}_{z \sim q_\phi|x}[log(p_\theta(\mathbf{x}|\mathbf{z}))]\big] \end{aligned} \tag{3}$$

Now we must choose a prior distribution $p(\mathbf{z})$. We choose $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as sampling from this is simple and it encourages *independence of dimensions* in order for $\mathbf{z}$ to capture the conditionally independent factors $\mathbf{v}$ in a disentangled manner. We want the approximation $q_\phi(\mathbf{z})$ to be close to this prior, which we may express by requiring that the KL divergence between $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$ be less than some error bound ϵ, resulting in a constrained optimization problem:

$$\begin{aligned} \underset{\theta,\phi}{\operatorname{argmax}} \quad & \mathbb{E}_{x \sim D}\big[\mathbb{E}_{z \sim q_\phi|x}[logp_\theta(\mathbf{x}|\mathbf{z})]\big] \\ \text{s.t.} \quad & D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z})\big) \leq \epsilon \end{aligned} \tag{4}$$

We convert this constrained optimization problem to a strict one by adding a slack term β to the inequality constraint to get a loss function we can optimize:

$$\underset{\theta,\phi}{\operatorname{argmax}} \quad \mathbb{E}_{x \sim D}\big[\mathbb{E}_{z \sim q_\phi|x}[logp_\theta(\mathbf{x}|\mathbf{z})]\big] - \beta \cdot D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z})\big) \tag{5}$$

The authors arrive at this formulation using a KKT Lagrangian instead [Hig+17]. We think this might be misleading, since they provide no argument for manually choosing the KKT multiplier (which corresponds to β) instead of performing the corresponding dual optimization (which would include β in the variables to optimize over). While the main contribution of the paper *is* to vary β and observe the results, this hyperparameter tuning does *not* correspond to the formal constrained optimization problem (4), but in optimizing an external measure such as a qualitative or quantitative measure of disentanglement. Thus the KKT formalization does not actually do anything towards justifying β as a hyperparameter. The approach we present here arrives at the same result, but makes the manual choice for β more explicit by adding β as a slack term and thus as a hyperparameter.

The optimization problem in (5) is identical to the loss function found in the original VAE, except for the additional β term [KW13]. We may thus follow the derivation in [KW13] to arrive at the final loss function:

$$\mathcal{L}(\theta, \phi; \mathbf{x}) \simeq \beta \cdot \frac{1}{2} \sum_{j=1}^{M} \left(1 + log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\right) \quad + \quad logp_\theta(\mathbf{x}|\mathbf{z}) \tag{6}$$

where $\boldsymbol{\sigma}^2, \boldsymbol{\mu}$ are outputs of the probabilistic encoder corresponding to $q_\phi|\mathbf{x}$ and $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ according to the reparametrization trick also outlined in [KW13]. The reconstruction loss term $logp_\theta(\mathbf{x}|\mathbf{z})$ is a Monte Carlo estimate using a single sample of $\boldsymbol{\epsilon}$.

3 Model Architecture

We used convolutional neural networks as probabilistic encoders and decoders for the distributions $\mathbf{z} \sim q_\phi|\mathbf{x}$ and $\mathbf{x} \sim p_\theta|\mathbf{z}$, respectively.

3.1 Encoder Architecture

The encoder follows a standard convolutional neural network architecture as outlined e.g. in [DV16]. Its output are the parameters $(\boldsymbol{\mu}, log(\boldsymbol{\sigma}^2))$ for the distribution $\mathbf{z} \sim q_\phi|\mathbf{x}$, thus the size of the output layer must be $2 \cdot M$. We output log-variance for numerical stability for small variance values.

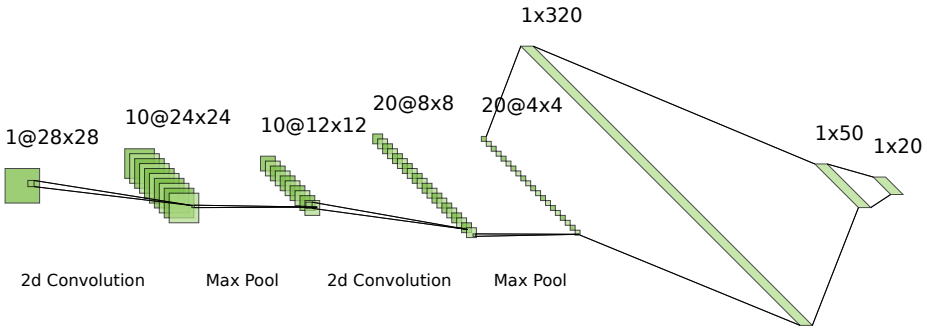

Figure 1: Encoder Architecture

No non-linearity is used on the output layer in order for the last fully connected layer to be able to output any range for $\boldsymbol{\mu}$ and $log(\boldsymbol{\sigma}^2)$. For details such as the parameters on the convolutional layers and non-linearities, see the class *Encoder* in *models.py* in [Ano22a].

3.2 Decoder Architecture

The decoder mirrors the architecture of the encoder. Encoder layers are mirrored by the following decoder layers:

Encoder Layer	Decoder Layer
Fully Connected(a, b)	Fully Connected(b, a)
2D Max Pool(kernel size=k)	Interpolate(scale factor=k, mode='nearest')
2D Convolution(in channels=a, out channels=b, kernel size=k)	2D Transpose Convolution(in channels=b, out channels=a, kernel size=k)

These pairs are chosen to enable the decoder to closely model the inverse of the encoder, as suggested in [DV16]. The last layer of the decoder uses a *sigmoid* so that the outputs are in $[0, 1]$ as in the input images. For details including the parameters on the convolutional layers and non-linearities used, see the class *Decoder* in *models.py* in [Ano22a].

3.3 Loss Function

We use *MSE* to model the reconstruction loss $-logp_\theta(\mathbf{x}|\mathbf{z})$, which is often used as it is proportional to the log likelihood, including in the practical for this course. It has been suggested that this is a mistake, as the variance of the distribution is part of the proportionality constant, which shrinks during the training process, and *LMSE* should be used instead [Yu20]. However, we did not test the empirical consequences of this claim.

4 Dataset Choice

[Hig+17] uses a range of higher resolution datsets including **CelebA**. However, we were only able to train using the **MNIST** dataset as well as our own **Shapes** dataset due to very limited access to computational resources. **MNIST** was sufficient for qualitative inspection of the reconstruction loss across varying values of β and to test the authors conjecture about transfer learning (section 7).

Prior to considerations about computational resources, we were planning on using the **CelebA** dataset. This dataset might provide more interesting independent generative factors, such as face rotations, lighting conditions, and continuous facial characteristics. These independent generative factors could have lead to better disentanglement results. Also, some of these factors might be highly predictive of characteristics for which labels exist in **CelebA**, such as glasses, hair colour and smiles, which could have improved our transfer learning results.

5 Experiments and Results

We ran a variety of experiments inline with those of the original paper. This included training beta-VAEs across various values of beta and latent-space sizes. We explored the variation of $KL\ Divergence\ Loss + Reconstruction\ Loss$, varying latent space size in the range $[5, 125]$ and β in the range $[0.002, 20]$. As expected, larger values of beta and lower latent space size both consistently lead to larger test loss. Models showed good performance for low values of β (Figure 2). We also inspected reconstruction quality manually, with β in $[0, 16]$ in latent space sizes 2 and 10. This qualitative comparison shows the reconstruction quality goes down significantly for higher values of β. Quality already goes down significantly for $\beta = 2$ in both cases, which might indicate that this value or higher are not suitable for this particular dataset (Figure 3).

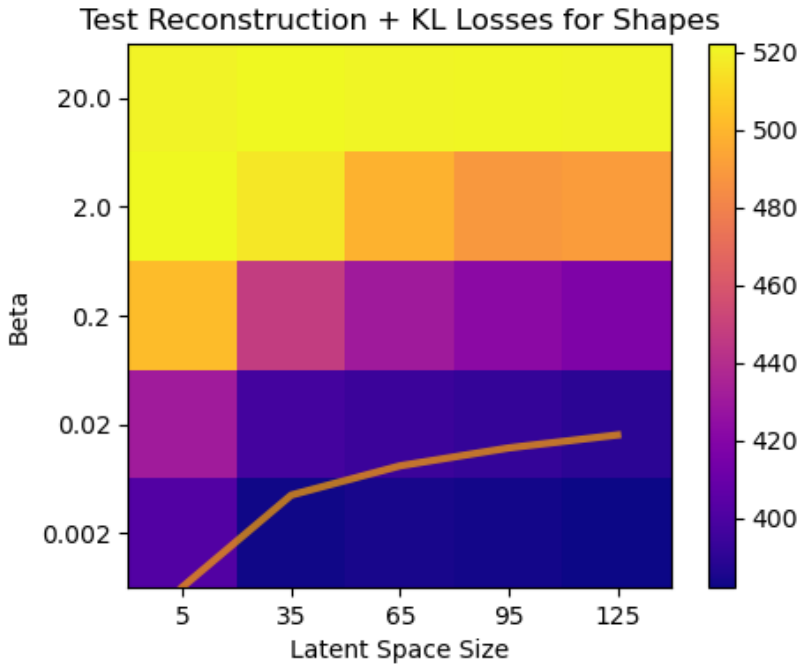

Figure 2: Consistently, models with higher normalised values of beta or lower dimensional latent spaces have higher test loss. This is to be expected, since each of these factors reduces the latent channel capacity. The orange line represents the normalised value of beta that corresponds to a standard VAE ($\beta = 1$) for any given latent space size.

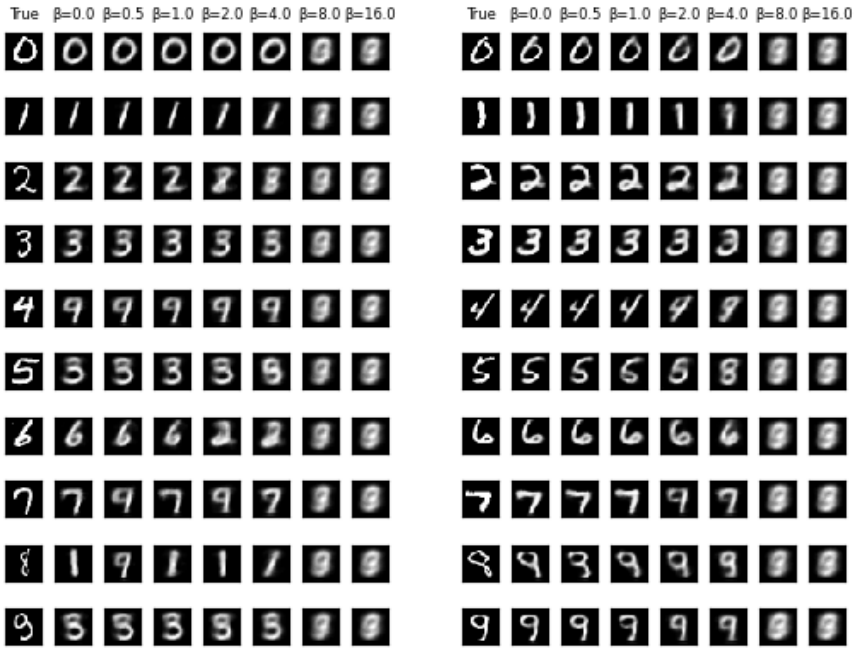

Figure 3: First columns show the input images, consecutive columns show the reconstruction using models trained with the corresponding beta values. Models have latent space size 2 on the left, 10 on the right.

6 Disentanglement Quantification

One contribution of [Hig+17] is a novel disentanglement quantification score. The measure scores each encoder, by quantifying how disentangled each of its latent dimensions are. Given a dataset in which images $\mathbf{x}$ are generated by $x \sim Sim(\mathbf{v})$ where $\mathbf{v}$ is a chosen and known vector of independent generative variables, we measure the disentanglement of an encoder *enc* as follows:

- Fix a chosen generative factor y, and generate a pair of generative vectors $\mathbf{v}_1$ and $\mathbf{v_2}$ that match in dimension y.
- Create two images $\mathbf{x}_1 \sim Sim(\mathbf{v_1})$ and $\mathbf{x}_2 \sim Sim(\mathbf{v_2})$ and encode them to get $\mathbf{z_1} = enc(\mathbf{x_1})$ and $\mathbf{z_2} = enc(\mathbf{x_2})$.
- Calculate the difference between these two encodings: $\mathbf{z_{diff}} = \mathbf{z_1} - \mathbf{z_2}$
- Repeat this process L times for fixed y, and take the average $\bar{\mathbf{z}}_{diff}$.
- Choose a new y, and repeat B times to create a dataset of $\bar{\mathbf{z}}_{diff}$'s.

- Train a classifier to predict y from the vector $\bar{\mathbf{z}}_{diff}$. The accuracy of the classifier is the disentanglement score

Further details of implementation of the score can be found in our GitHub repository [Ano22a] or in the original paper [Hig+17]. In order to implement the papers disentanglement measure, it is necessary to have a dataset with well-defined generative factors. To this aim, we replicate the **Shapes** dataset. Using four generative variables ($\mathbf{v} = [x, y, scale, rotation]$), we create images containing a white shape on a black background.

We ran a series of experiments, varying the normalised $\beta_{norm} = \beta * \frac{M}{N}$, where M is the size of the latent space and N is the size of the image ($28x28$)). We varied the normalised beta from $2e-3$ to $2e1$, and the latent space size from 5 to 125.

6.1 Disentanglement Quantification Results

Like the original experiments, our disentanglement measure demonstrates that, excessively high values of beta reduce disentanglement score across the range of latent space sizes. Similarly, they demonstrate that larger latent-space continue to have high disentanglement at larger values of beta than lower dimension latent spaces (Figure 4).

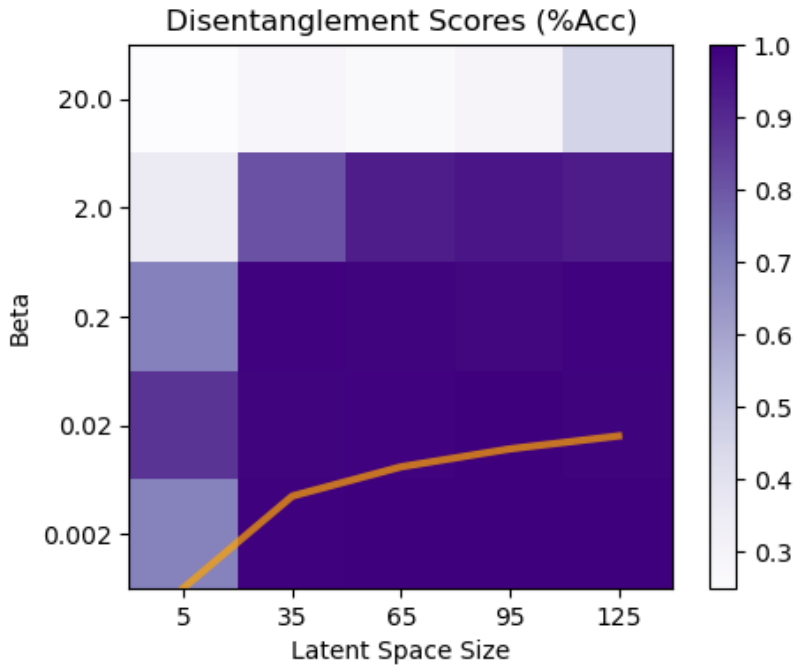

Figure 4: Color axis shows disentanglement score test accuracy

Unlike the original paper, we were unable to demonstrate that, for any particular latent-space size, raising the value of β increases the measured degree of disentanglement. We have three hypotheses for why this might be the case:

Lower Resolution images To save on computational cost, our *Shapes* dataset used 28x28 images, rather than the larger 64x64. Although normalising beta should counteract some of this effect, it is possible that the difference in results is a product of the simpler dataset. One of the impacts is that we could not vary the generating factors as much, such as the position and scale of the shape; another is that some variations like rotation are not as easily discernible. With access to more computational resources, we would experiment with higher resolution images, in which the effects of changing generative variables is more distinct.

Insufficient experimentation Due to limited computational resources, we were limited in the number of experiments we could run, and therefore combinations of hyperparameter we could investigate. We explored 5 values of β_{norm} and 5 latent space sizes, requiring 25 experiments. [Hig+17] were able to explore at least 400 hyperparameter combinations. With greater computational resources, we would explore a wider range of values ($\beta_{norm} \in [0.00002, 20.0]$) at a higher resolution. It is possible that a finer resolution of this hyperparameter search would have revealed an optimal value for β_{norm} as in [Hig+17].

Overly expressive classifier We can see that almost all of the models got a near-perfect disentanglement score, successfully predicting nearly 100 percent of the y values. We can also see that the models that perform poorly on the disentanglement metric are closely related to those that perform poorly on the test loss. It is possible that all models which learn to encode with reasonable success provide sufficient disentanglement for the classifier to be able to detect. One improvement to our experiments might be to reduce the power of the classifier. Since the classifier currently consists of only a single layer, to reduce its complexity, it is not obviously possible to reduce the number of parameters but harsh regularisation or fewer epochs of training might reduce the models performance. It is not clear that this reduction in performance would make the disentanglement score more informative.

6.2 Qualitative Disentanglement Results

We ran qualitative experiments exploring disentanglement of five-dimensional latent space with varying values of β on **MNIST** and our own **Shapes** dataset.

The experiments on **MNIST** showed that all dimensions are used to encode information correlated with the categorical distribution of digits - that is, there was no dimension which could be varied without changing which digit was represented. This effectively means that no two dimensions can be independent, as the distribution over digits is categorical. If there were, for example,

a dimension encoding *stroke thickness* but nothing else, this could be independent of other dimensions. We did not find any such dimensions in the latent spaces for any value of β, and thus have to conclude that there is no significant disentanglement using β-VAE on **MNIST**. Figure 5 shows variation of one latent dimension. For every value of β, information concerning the categorical distribution is encoded: we see variation from the digit 0 to the digit 1. The only relevant difference is the lower reconstruction quality for $\beta = 2$, specifically blurrier constructed images.

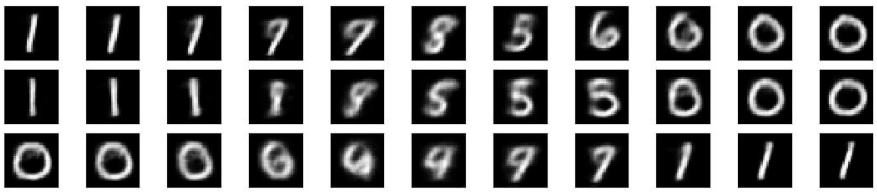

Figure 5: Rows show $\beta = 0$, 1, 2 from top down. Columns show varying one dimension of a 5-dimensional latent space from -2.5 to 2.5

Qualitative inspection of disentanglement on our own **Shapes** dataset showed better, though still not conclusive results. The setup remains the same as before, with β values 0, 1, 2 and a latent space size of 5. We can see notable differences: for $\beta = 0$, we see at least two generative variables encoded in one dimension, size and position. This is not the case for $\beta = 1$, which encodes only the rotation of the heart shape (except for the fact that towards the origin, the latent space always shows circles - this might be due to the fact that rotation is hard to learn on such low-resolution images and thus the high-density area around the origin matches position only but not rotation). Thus $\beta = 1$ seems to show significantly better disentanglement than $\beta = 0$. With $\beta = 2$, however, reconstruction of the heart fails and rotation is not captured at all; also, the dimension we varied does not seem to encode any significant information (Figure 6). Note that $\beta = 1$ corresponds to the standard VAE, thus these results fail to show improvement by adding the hyperparameter β.

7 Transfer Learning

In the conclusion of the paper, the authors suggest that varying the value of beta to more successfully disentangle the latent factors, might provide greater success for supervised or transfer learning: *"We believe that using our approach as an unsupervised pretraining stage for supervised or reinforcement learning will produce significant improvements for scenarios such as transfer or fast learning."* [Hig+17]. A major benefit of transfer learning from an unsupervised problem to a supervised problem is that it reduces the number of labels needed at the

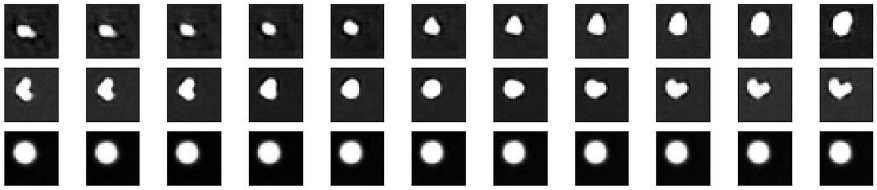

Figure 6: Rows show $\beta = 0$, 1, 2 from top down. Columns show varying one dimension of a 5-dimensional latent space from -2.5 to 2.5

supervised stage to achieve high performance. We empirically investigate the possibility that training classifiers using higher values of beta improves sample efficiency. We trained β-VAE with latent dimension sizes 2 and 10 on **MNIST**, then fixed the weights of the encoder and trained a single layer linear regression with softmax non-linearity to a one-hot encoded classification of digits. We did this for different values of β and plotted test loss against supervised samples used. 7

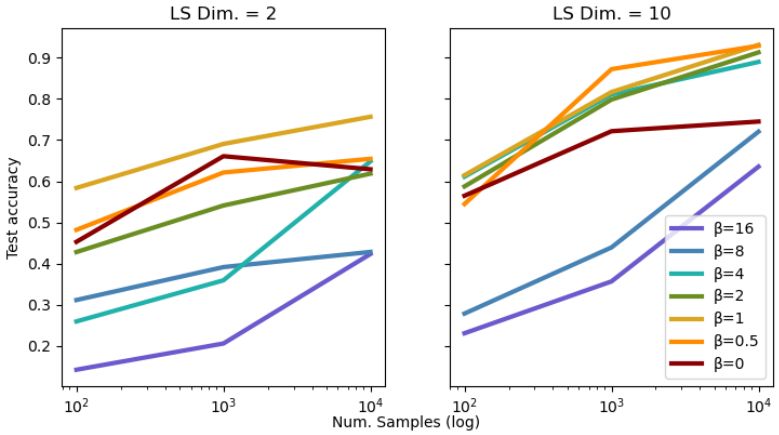

Figure 7: Color axis shows disentanglement score test accuracy

Our results show that more samples generally lead to better performance. They also show that increasing latent space dimension leads to improved performance and that increasing beta too high reduces performance. This is to be expected. They do show that increasing the value of β from 0 to 0.5 to 1 improves performance with latent space dimension 2 but do not robustly show that sample efficiency in particular improves for any particular value of $\beta > 1$. In fact, our results show that although the $\beta \in \{8, 16\}$ models performed okay with a large number of samples, they performed poorly with few samples: im-

plying they are *less* sample efficient than the $\beta = 1$ model. We could therefore *not* verify the claim that β-VAE could improve transfer learning results.

We discuss three hypotheses to explain our results:

MNIST does not have continuous generative factors The **MNIST** dataset may have independent generating variables such as writing speed, tilt, or pen thickness. However, the most important generative factor for this transfer learning task is which digit has been written: taken from a discrete categorical distribution. Therefore, tuning beta might make the latent variables more individually representative of the continuous generative factors, but this might not aid in a classification task.

$\beta = 1$ is appropriate for MNIST Our results do show that lower and higher values of β lead to worse performance. It is possible that contingent factors about MNIST (its resolution, colour depth) make $\beta \approx 1$ appropriate for learning disentangled representations. On more complex datasets, different values of β may be appropriate, necessitating the hyperparameter.

β-VAE does not significantly improve transfer learning It may also be possible that learning disentangled representations is useful for interpretability but not for transfer learning. This could be because the loss of information from reduction in latent channel capacity dominates the gain from simplicity of disentangled representations: especially if the classifier is highly expressive.

With more computation time, we suggest performing transfer learning experiments with our setup on the **CelebA** dataset: the dataset has many highly independent generating variables as stated before, and has binary annotations which can be used in the same unsupervised to supervised transfer learning setup.

8 Interactive Demonstration

For better communication of the results, we aimed at building interactive demonstrations of the conducted experiments (online at [Ano22b], source at [Ano22c]). We think interactive demonstrations can help communicate significant results in machine learning to a wider audience, as well as allow experts to explore models and data more efficiently. However, we found that the best current pipelines for creating interactive demonstrations require a prohibitive time effort. We used the ONNX format to export our models to javascript in order to run in a browser; however, we found javascript implementations of ONNX are incomplete and poorly maintained. There is no interactive drawing library that works well together with ONNX models, so we wrote our own minimal interactive drawing library. The current interactive poster should be seen as a proof-of-concept. Showing all results of this paper interactively would have required building an interaction library for ONNX and would have taken a prohibitive

amount of time. We suggest such an effort could improve machine learning research and communication, including human interpretability of models through easier exploration.

9 Conclusion

We have repeated the two important results of the given paper, training a VAE with added hyperparameter β and disentanglement quantification using an artificial dataset. The quantitative results suggest that for the MNIST dataset, a beta value of around 1.0 is optimal, which does not confirm better disentanglement of β-VAE compared to regular VAE architecture. We have suggested this may be due to the simplicity of the dataset, along with the categorical distribution of digits which does not lend itself readily to a disentangled non-categorical latent distribution. We have also shown how to set up transfer learning using a pre-trained encoder, and quantified results. We did not find a significant improvement of transfer learning using the MNIST dataset; again, we speculated the reason for this may be the categorical distribution of MNIST, which does not lend itself to be represented in a disentangled latent space, and listed other possible reasons. We have suggested a procedure to test transfer learning using a β-VAE encoder, which requires larger datasets and more compute time than is available to us.

References

[KW13] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. DOI: 10.48550/ARXIV.1312.6114. URL: https://arxiv.org/abs/1312.6114.

[DV16] Vincent Dumoulin and Francesco Visin. *A guide to convolution arithmetic for deep learning*. 2016. DOI: 10.48550/ARXIV.1603.07285. URL: https://arxiv.org/abs/1603.07285.

[Hig+17] Irina Higgins et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *ICLR*. 2017.

[LPK20] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. "Explainable AI: A Review of Machine Learning Interpretability Methods". In: (2020). DOI: 10.3390/e23010018. URL: https://dx.doi.org/10.3390/e23010018.

[Yu20] Ronald Yu. "A Tutorial on VAEs: From Bayes' Rule to Lossless Compression". In: *CoRR* abs/2006.10273 (2020). arXiv: 2006.10273. URL: https://arxiv.org/abs/2006.10273.

[Ano22a] Anonymized. *beta-vae*. hyperurl.co/jruj1z. 2022.

[Ano22b] Anonymized. *Interactive Poster*. https://tinyurl.com/interactivePosterBVAE. 2022.

[Ano22c] Anonymized. *Interactive Poster Source*. `https://tinyurl.com/interactivePosterSource`. 2022.