Unsupervised Disentangled Representation Learning: Survey on Methods of Learning Disentangled Representation with VAEs

Omran Kaddah
Technical University of Munich
kaddah.omran@gmail.com

Abstract: This work compares different state-of-the-art methods for learning disentangled representation in unsupervised settings using the framework of Variational Autoencoders(VAEs). The comparison is based on quality of the outputs of the studied models, as well as measuring disentanglement of the latent representations.

Keywords: Representation Learning, Disentanglement, Evaluation Metrics

1 Introduction

Obtaining disentangled and interpretable representation of data can be beneficial in many areas in the field of Artificial Intelligence(AI) such as semi-supervised classification [1, 2], reinforcement learning [3], and transfer learning and zero-shot learning [4]. Most of the works on learning disentangled representation have been based on the class of frameworks VAE [5], and generative adversarial network(GAN) [6]. On one hand, works on VAEs have been more focused on simple datasets, and achieved humble results. None of these works have been able to achieve the desired disentanglement where latent units are totally uncorrelated for simple datasets, and also suffered from a low quality and blurry generated outputs. On the other hands, works on GANs that explicitly added terms to the objective to encourage mutual information between latent representation and outputs such as InfoGAN [7] achieved similar comparable results to those of VAEs, other than that, works such as [8, 9] excelled in having high quality outputs, and the ability to control and manipulate the output. However, for these models, multiple latent units mostly correspond to one or more generative factor, and require extra steps to figure out for how traverse latent unit in order to make aimed changes in the output. This bring us to what definition of disentanglement we adopt in this work. It is the definition in [10] where each dimension of a representation corresponds to one factor of variation that does not depend on any other dimension. In this paper, only works of the VAE framework on disentangled representation learning will be considered, evaluated, and compared under one neural network architecture, in order to make the comparison as objective and fair as possible.

In this paper the following is going to be discussed:

1. The VAE frameworks.
2. Disentanglement Metrics.
3. Quantitative and qualitative comparisons between the studied models.
4. Observation, other related experiments that were carried along with the survey.
5. Suggestions and future works.

2 Variational Autoencoder (VAE)

VAE [5] is a generative model that aims to learn the marginal likelihood of a given data. It achieves that through variational distributions that approximate the intractable marginal likelihood of data by optimizing the evidence lower bound(ELBO). The variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$, where ϕ is the parameters of the encoder, is introduced. It aims to approximate the underlying latent distribution of data. This results in the objective function of VAE, which is to be maximized the ELBO:

$$\log p_\theta(\mathbf{x}) \geq \mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) \tag{1}$$

Where θ is the parameters of the decoder network. The latent prior is assumed to be a unit isotropic Gaussian distribution, and the latent posterior is assumed to be a Gaussian distribution with diagonal covariance matrix. The derivation of the ELBO can found in [5]. The rest of this section will discuss different variations on the objective function, which had the aim to improve disentanglement in the latent representation.

2.1 β-VAE

The simplest variation on the objective function, it was introduced in [3], and it stems from idea that the objective function of VAE is closely related the information bottleneck method [11](derivation for that can be found in [12]).

β-VAE objective:

$$1/N \sum_{i=1}^{N} [\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(\mathbf{i})})}[\log p_\theta(\mathbf{x}^{(\mathbf{i})}|\mathbf{z})] - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(\mathbf{i})}) \parallel p(\mathbf{z}))] \tag{2}$$

Information Bottleneck Principle:

$$max(I(\mathbf{z};\mathbf{y}) - \beta I(\mathbf{x};\mathbf{z}))$$

One can clearly see that the original objective of VAE can be retrieved by setting $\beta = 1$. It was further elaborated in [13] that placing extra penalty on the kullback leibler(KL) divergence between variational posterior and prior of latent distribution, would push the model to pass compact and important information through the bottleneck, as the KL divergence term implies the mutual information $I(\mathbf{X};\mathbf{Z})$ between the input and latent embedding, and this term is also correlated with reconstruction term. Thus, the model would have to pass less information, and information that contributes better to the reconstruction term. Disentangled representation does the best for such case, and that explains the source of disentanglement with β-VAE model. One can also see that putting extra penalty on KL divergence term will negatively affect the quality of the reconstruction, as there will be less information coming from the input through bottleneck to the output. To reduce this effect, It was proposed [13] to set a Lagrange multiplier for capacity of the bottleneck, pushing the bottleneck to have certain capacity that increases after some iterations during training phase of the network. In this paper this model is referred to as controlled capacity(CC)-VAE

CC-VAE objective:

$$\frac{1}{N} \sum_{i=1}^{N} [\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(\mathbf{i})})}[\log p_\theta(\mathbf{x}^{(\mathbf{i})}|\mathbf{z})] - \gamma|C - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(\mathbf{i})}) \parallel p(\mathbf{z}))|] \tag{3}$$

2.2 Joint-VAE

This objective [14] follows the same principle of the β-VAE objective. However, it considers modeling latent discrete dimensions with Gumble-softmax distribution [15] that approximates the non-differentiable Categorical distribution, and thus, allowing for gradient flow to the encoder. In this case the output of the encoder for the discrete latent dimension are fed to softmax function, and the KL divergence term is computed between the posterior of Gumble-softmax distribution and uniform categorical distribution

2.3 Factor-VAE

Proposed in [16], it was shown based on [17] that KL divergence term can reformulated as follows:

$$D_{KL}(q(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) = I(\mathbf{x};\mathbf{z}) + D_{KL}(q(\mathbf{z}) \parallel p(\mathbf{z})) \tag{4}$$

Where the first term is the mutual information of the input and latent embedding, and the second term is the KL divergence between the marginal latent distribution and the factorial prior, that its marginal can be expressed as product of each dimension. That brings the objective function of Factor-VAE:

$$\frac{1}{N} \sum_{i=1}^{N} [\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(\mathbf{i})})}[\log p_\theta(\mathbf{x}^{(\mathbf{i})}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(\mathbf{i})}) \parallel p(\mathbf{z}))] - \gamma D_{KL}(q(\mathbf{z}) \parallel \bar{\mathbf{q}}(\mathbf{z})) \tag{5}$$

Where $\bar{\mathrm{q}}(\mathbf{z}) = \prod_{j=1}^{d} q(\mathbf{z}^{(i)})$. The Total Correlation(TC) [18] appears as the new term to VAE objective, and weighted by γ. This term pushes the network to learn more factorized dimensions for the latent embedding. This objective therefore, does not explicitly put extra penalty for the mutual information that is implied in KL divergence term of the VAE objective. A sample from $q(\mathbf{z}|\mathbf{x}^{(i)})$ for uniformly chosen data point is considered a sample from $q(\mathbf{z})$. TC is minimized with density ratio trick [19, 20]. Thus, for this objective, a discriminator network is required, which is denoted with D.

$$TC(\mathbf{z}) = D_{KL}(q(\mathbf{z}) \parallel \bar{\mathrm{q}}(\mathbf{z})) \approx \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{D(\mathbf{z})}{1 - D(\mathbf{z})}\right] \tag{6}$$

2.4 β-TCVAE

KL divergence term of the original objective of VAE can be also also reformulated as follows [21]:

$$\mathbb{E}_{p(\mathbf{x})}\Big[D_{KL}\big(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\big)\Big] = \underbrace{D_{KL}\left(q(\mathbf{z},\mathbf{x})||q(\mathbf{z})p(\mathbf{x})\right)}_{\text{(i) Index-Code MI}} + \underbrace{D_{KL}\big(q(z)||\prod_j q(z_j)\big)}_{\text{(ii) Total Correlation}} + \underbrace{\sum_j D_{KL}\left(q(z_j)||p(z_j)\right)}_{\text{(iii) Dimension-wise KL}} \tag{7}$$

Where (i) and (ii) have been already discussed in the previous sections, and (iii) is the total of how much each individual latent dimension deviate from their corresponding prior. Further detailed decomposition analysis can be found in the original paper [21]. The final objective will be:

$$\frac{1}{N}\sum_{i=1}^{N}[\mathbb{E}_{q_\phi(\mathbf{z}|x^{(i)})}[\log p_\theta(x^{(i)}|\mathbf{z})] - \Big(\alpha\big(D_{KL}\big(q(\mathbf{z},\mathbf{x})||q(\mathbf{z})p(\mathbf{x})\big)\big) + \beta\big(D_{KL}\big(q(z)||\prod_j q(z_j)\big)\big) + \gamma\big(\sum_j D_{KL}(q(z_j)||p(z_j))\big)\Big) \tag{8}$$

The authors [21] also proposed two methods for evaluating the density of $q(\mathbf{z})$,these are minibatch-weighted sampling and minibatch stratified sampling. The first will only be considered and applied in the experiments on β-TCVAE.

$$\mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] \approx \frac{1}{M}\sum_{i=1}^{M}\left[\log \frac{1}{NM}\sum_{j=1}^{M} q(\mathbf{z}(x^{(i)})|\mathbf{z}_j)\right] \tag{9}$$

Where M is the size of the batch and $\mathbf{z}(\mathbf{x}^{(i)})$ is the latent embedding from datapoint $\mathbf{x}^{(i)}$. For clarity, $q(\mathbf{z}(\mathbf{x}^{(i)})|\mathbf{z}_j)$ means the density for $\mathbf{z}(\mathbf{x}^{(i)})$ evaluated with parameters from $\mathbf{x}^{(i)}$.

3 Disentanglement Metrics

There has been a number of metrics for measuring disentanglement, three of which came along the works mentioned earlier. Metric mentioned in [22] suffers from sensitivity to hyperparameter tuning and choice of classifier, has failure mode, and cannot detect axis-alignment (a latent dimension being aligned to a generative factor). The later work in [16] addressed the issues of the previous metric, and it is going to be considered in comparison and evaluation section. The Factor-VAE metric is described in **Algorithm 1**.

Mutual-Information-Gap(MIG) [21] was also initially considered. However the implementation provided by the authors did not adhere to the results provided in the paper, and resulted small values compared to what the paper had shown. Therefore, making the metric unreliable, leaving only the

Algorithm 1 Factor-VAE Metric

$dataset \leftarrow$ **get_latent_samples() List** of(**int**, **List**) ▷ get list of list of samples that have one fixed generative factor, each list is of length L. Note that this dataset is of latent embedding, where the latent units with low KL Divergence are discarded. The truncated embeddings are also divided by the empirical standard deviation of full dataset latent embedding.
$votes \leftarrow$ **matrix**$(num\_factors, num\_latent\_dim)$
for each $factor, \mathbf{z} \in dataset$ **do**
    $variances \leftarrow$ **get_var_per_dims**$(\mathbf{z})$
    $i \leftarrow$ **argmin**$(\mathbf{z})$
    record_vote$(votes, i, factor)$
end for
$classifier \leftarrow$ **argmax_colums**$(votes)$ ▷ Here we get a classifier, which is an array, where each index corresponds to a latent dimension, each cell contains the generative factor that is assigned to that index of the cell
for each $index, factor \in classifer$ **do**
    $correct\_votes \leftarrow correct\_votes + votes(factor, index)$
end for
$num\_votes \leftarrow$ **sum_elements**$(votes)$
return $correct\_votes \div num\_votes$

option of discarding it. and No comment was provided from author's side, despite that the issue has been reported from different people in the community [23]. Two common drawback of all previous disentanglement metrics, is that they all require a model with an encoder that outputs a latent embedding, which means that these metrics cannot be applied for GANs. And they require a dataset with known generative factors as well.

4 Evaluations and Comparisons

The implementation was carried with PyTorch library[24] on Python, and the weights of the network were initialized by default methods of the library in version1.14.0. The experiments were done on two datasets. which are sorted according to known generative factors, Dsprites(2D shapes) [25] and 3D Shapes [26]. The models have been evaluated with same encoder and decoder architecture mentioned in[13] for Dsprites, and for 3D shapes the architecture shown in Table 1 was used, where the each layer is followed by batch normalization then RelU activation. In this work the number of experiments carried for Dsprites dataset was view for an ablation study, especially with respect variations on the hyperparameters. Therefore to pick a better hyperparameters for the model, the numerous experiments already done in [23] were taken into consideration. Best results for each of models trained for one of the presented objective function are displayed in both quantitative and qualitative comparison. Few experiments were done for the 3Dshapes, therefore, comparison will not be discussed in this work. However, tools are for making experiments are already available and can be found in the repository of this work. It is important to note this comparison is not perfect, and definitely has limitations, such as, comparison is made on one dataset, different united architectures for the models were not considered, and of course not all search space of hyperparameters is covered because it is huge.

4.1 Quantitative Comparison

Two criteria are considered for the evaluation and comparison, quality of reconstruction measured as the average reconstruction error over the entire dataset, and disentanglement measured with Factor-VAE disentanglement metric. The results of the disentanglement metric were recorded after the average of 5 trials, since the samples for a fixed generative factor are random.

4.2 Qualitative Comparison

For visual comparisons, samples for each model from the same input can be found in the appendix. Observe that almost all models were able to find and encode 4 generative factors out of 5 of Dsprites dataset. These are the position in the x-axis, position the y-axis, orientation, and scale. However,

Table 1: Architecture of model used for 3D shapes dataset.

Encoder					Decoder				
Layer	Channels	Kernel Size	Stride	Padding	Layer	Channels	Kernel Size	Stride	Padding
1	32	4x4	2	1	1	512	1	1	0
2	32	4x4	2	1	1	64	4x4	1	1
3	64	4x4	2	1	2	64	4x4	2	1
4	64	4x4	2	1	3	32	4x4	2	1
5	512	4x4	2	0	4	32	4x4	2	1
6	latent size	1	1	0	5	3	4x4	2	1

Comparison Between different variation of VAE objectives.

Model\Dataset	Dsprites		
-\Measure	Disentglement		Recons
	Training	Test	
β-VAE	0.74	0.731	42.66
CC-VAE	0.761	0.761	27.03
Joint-VAE	**0.801**	**0.792**	**15.32**
Factor-VAE	0.705	0.696	26.73
β-TCVAE	0.713	0.717	28.42

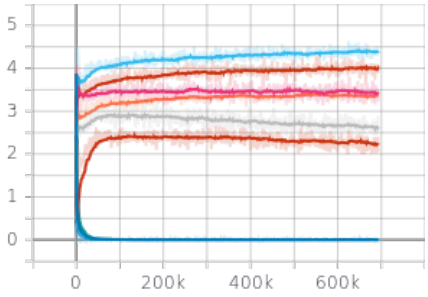
Table 2: On dsprites dataset, the model with Joint-VAE objective function had the best results, both in disentanglement accuracy and reconstruction loss. This is will also be appear for the visual outputs of this model. With regard to 3Dshapes dataset, not enough experiments were done to make a fair comparisons. Therefore these initial results are displayed not for the sake of comparison in this case. Rather for an overview of how results might look like. Hyphen marks either missing or irrelevant.

all failed to encode the shape as generative factor, and instead some models entangled it with other factors. Therefore, one can see that traversing some latent dimensions will sometimes cause the shape to change. Joint-VAE model had the best visual results. For some outputs traversing the discrete units, corresponded to a change in the shape of the output. In addition to that, one can also observe the sharp and less blurry outputs compared to other models.

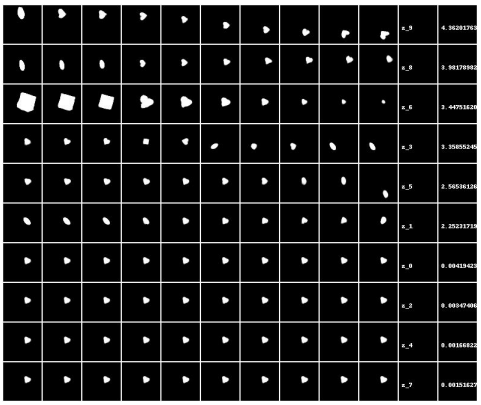
5 Remarks and future works

The experiments on Dsprites dataset with models of different variations of the VAE objective function, showed that β-VAE objective function, and its two other adaptions CC-VAE and Joint-VAE outperformed the objective functions where TC is calculated and **considered in**. Though this might sound counter intuitive, and contradict the claims of papers that published these works that involve TC in the objective function and claimed their models outperformed β-VAE. It can be explained that calculating TC was always based on estimations, which were not proven not be poor.These results cannot be stated as final, because of the limitation discussed earlier. However, it shows that it is not always the case where methods with TC outperforms the others

Other experiments were done through the course of this works. Pushing the active units in the latent embedding to have close and almost equal KL divergence amongst each other. This idea came after observing that the unit with the highest KL divergence often coded multiple factors, therefore it might be the case that if the units had similar capacity will prevent them from encoding more than one factor. However, experiments that were carried on a model with the CC-VAE objective. Showed that the networks was harder to train, had unstable gradient, and the end result was a model with very bad reconstruction. This can be explained that some factors are more important than other, and require latent units with a higher capacity than other. Additionally, pushing units to have similar individual bottleneck capacity, will cause them to have similar KL divergence with prior, and thus making the distributions of latent dimensions similar, which makes it hard for the network to encode distinct information for certain samples in the dataset. Experiments were also done on deep learning unsupervised generative clustering approach within the framework of VAE [27] named



(a) Training diagram



(b) Visual outputs

Figure 1: β-TCVAE. Figure (a) shows the KL divergence of each unit plotted against training iteration. Figure (b) shows traversal of latent dimensions. The dimensions are order by their KL divergence. These two figures are of model trained on Factor-VAE objective, and inspired the idea of pushing KL divergences to be similar, rather having one dimension encoding several factors. One can observe for this case, the dimension with highest KL divergence encoded shape, orientation, position.

VADE. Clusters are created according to the latent embedding generated by the encoder. The prior of latent embedding is a Mixture-of-Gaussians(MoG). Making VaDE suitable for clustering by design. The drawback of such a model is the requirement of initializing the clusters with GMM. Therefore, experiments we carried by initializing the network with some the discussed VAE objective functions, such as β-TCVAE. The results with regard to cluster accuracy and reconstruction error were not any better than an initialization with GMM. This however does not cancel the idea of making use of disentangled representation for clustering, as some works have already experimented with that [28, 29] Experiments in the opposite direction were also done, to check whether using VADE results in a disentangled latent spaces. However, the results were comparable to model with the original VAE objective function; almost no signs of disentanglement. For example, VADE failed to cluster dsprites dataset according to the shape, and traversing the latent space resulted in a random variation in the output.

Future works may give more focus on making systematic and objective comparisons, with more experiments done with datasets with known generative factors. The datasets with known generative factors as mostly artificial. Therefore metric with no requirement for a known generative factors is needed for real artifacts datasets.An already available example of such metric is Perceptual Path Length [8], which also does not require an encoder layer. Also, GANs should be more considered, because they do not require a known prior distribution for objective function, they have a better quality outputs, and are now easier to train with recent to techniques to stabilize its training process. As final remark, transparency and sharing an already trained models that claimed certain results should be highly encouraged.

6 Conclusion

This work showed that the new variations on β-VAE did not necessarily always show better quantitative and qualitative results. This work does not claim to be perfect, or aims to undermine other works. However, it advocates a more systematic, objective, transparent evaluation and comparison of models for disentangled representation learning.

References

[1] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[2] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[3] I. Higgins, A. Pal, A. Rusu, L. Matthey, C. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1480–1490. JMLR. org, 2017.

[4] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[5] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[7] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[8] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[9] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. *CoRR*, abs/1912.04958, 2019.

[10] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[11] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[12] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[13] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[14] E. Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, pages 707–717, 2018.

[15] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[16] H. Kim and A. Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[17] M. D. Hoffman and M. J. Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, page 2, 2016.

[18] S. Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.

[19] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56 (11):5847–5861, 2010.

[20] M. Sugiyama, T. Suzuki, and T. Kanamori. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012.

[21] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018.

[22] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr*, 2(5):6, 2017.

[23] D. L. Yann Dubois, Aleco Kastanos. Disentangled vae. https://github.com/YannDubs/disentangling-vae, 2019.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[25] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[26] C. Burgess and H. Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

[27] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

[28] J. Antoran and A. Miguel. Disentangling in variational autoencoders with natural clustering. *arXiv preprint arXiv:1901.09415*, 2019.

[29] M. Willetts, S. Roberts, and C. Holmes. Disentangling to cluster: Gaussian mixture variational ladder autoencoders. *arXiv preprint arXiv:1909.11501*, 2019.

Appendix

A Visual Results and training diagrams

These are visual results and training diagrams of plots KL divergence of each dimension against training iterations of models from which results are recorded for comparison. All models had latent embedding of 10 dimensions.

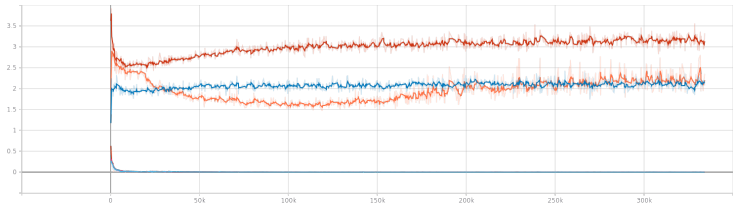


Figure 2: training diagram for β-VAE

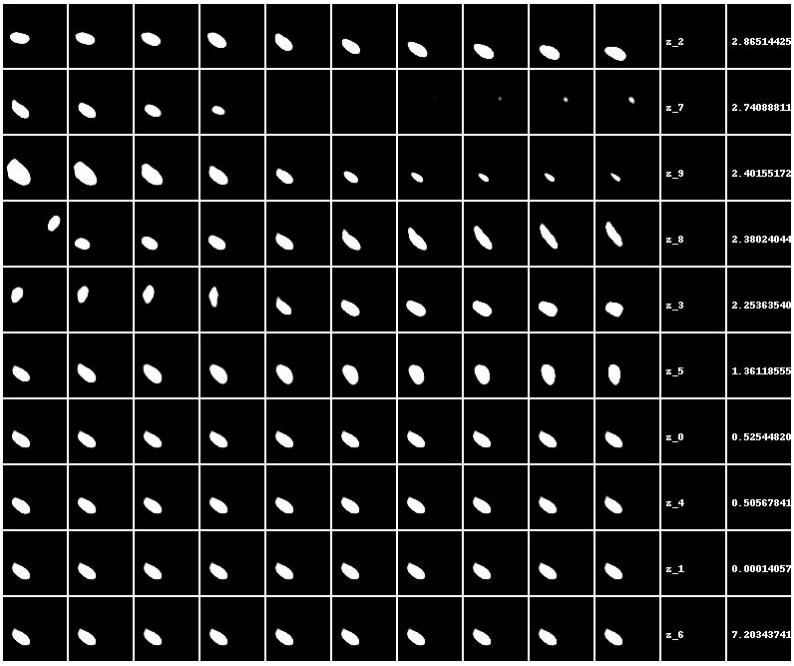


Figure 3: traversing continuous units for β-VAE. Last two columns refer to the unit and its KL divergence

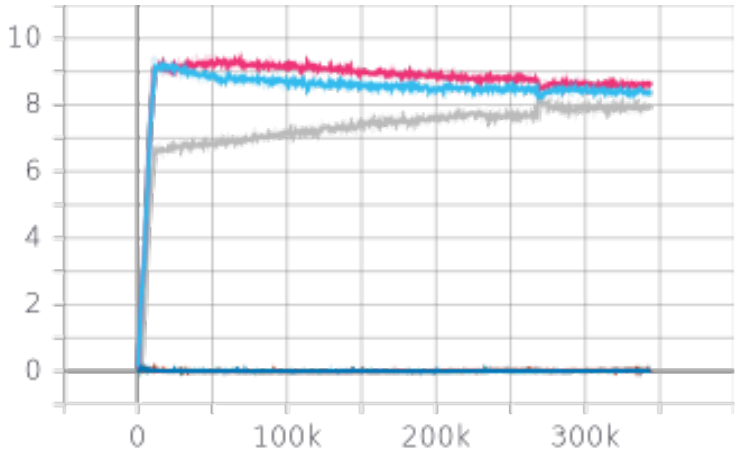


Figure 4: training diagram for CC-VAE

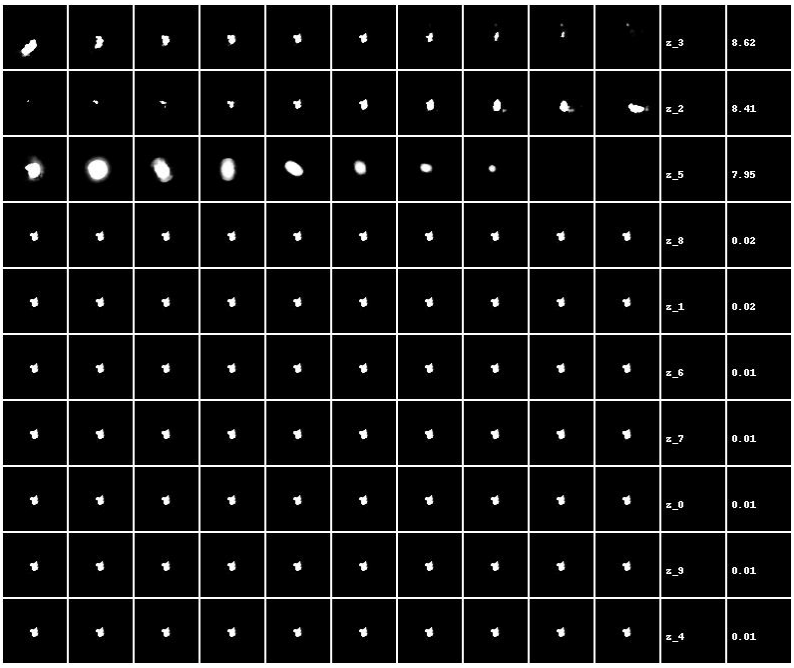


Figure 5: traversing continuous units for CC-VAE. Last two columns refer to the unit and its KL divergence

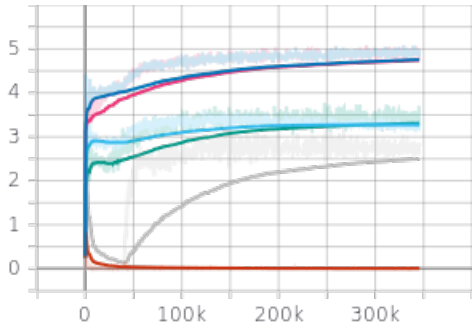


Figure 6: training diagram for Joint-VAE

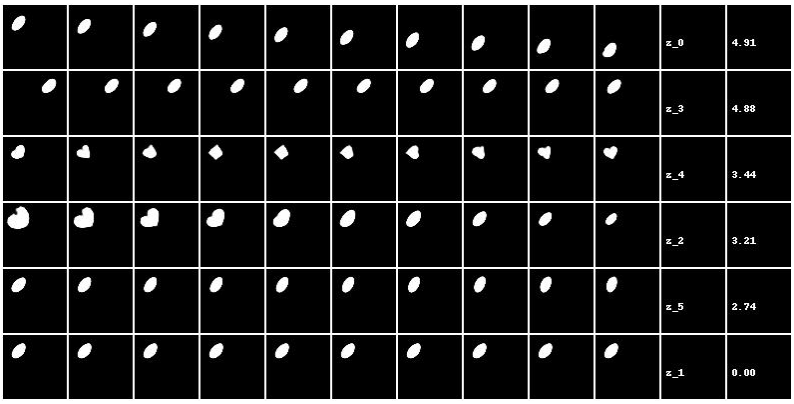


Figure 7: traversing continuous units for Joint-VAE. Last two columns refer to the unit and its KL divergence

Figure 8: traversing discrete units for Joint-VAE. Last two columns refer to the unit and its KL divergence

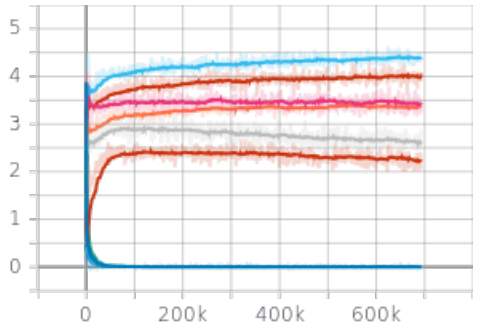


Figure 9: training diagram for Factor-VAE

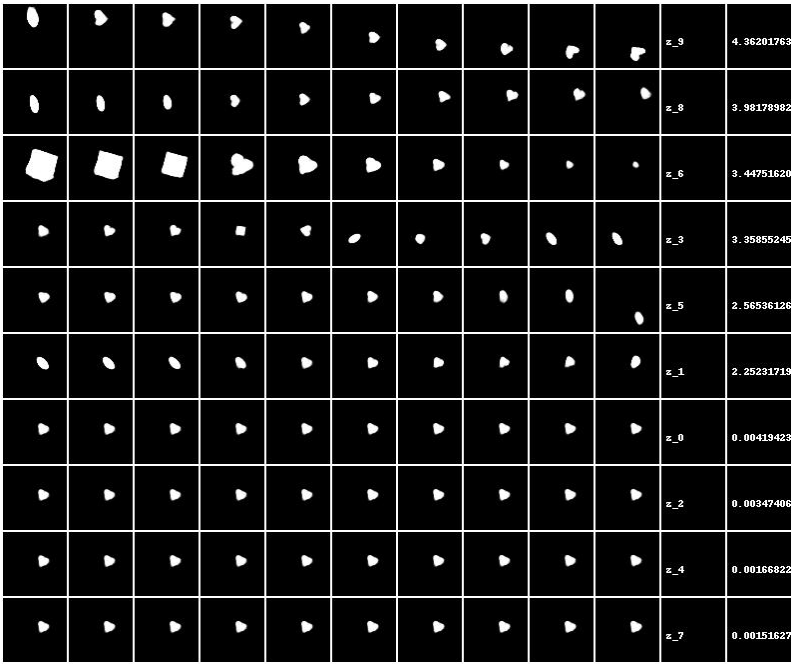


Figure 10: traversing continuous units for Factor-VAE. Last two columns refer to the unit and its KL divergence

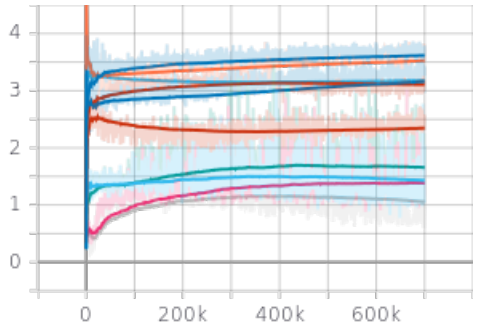


Figure 11: training diagram for β-TCVAE

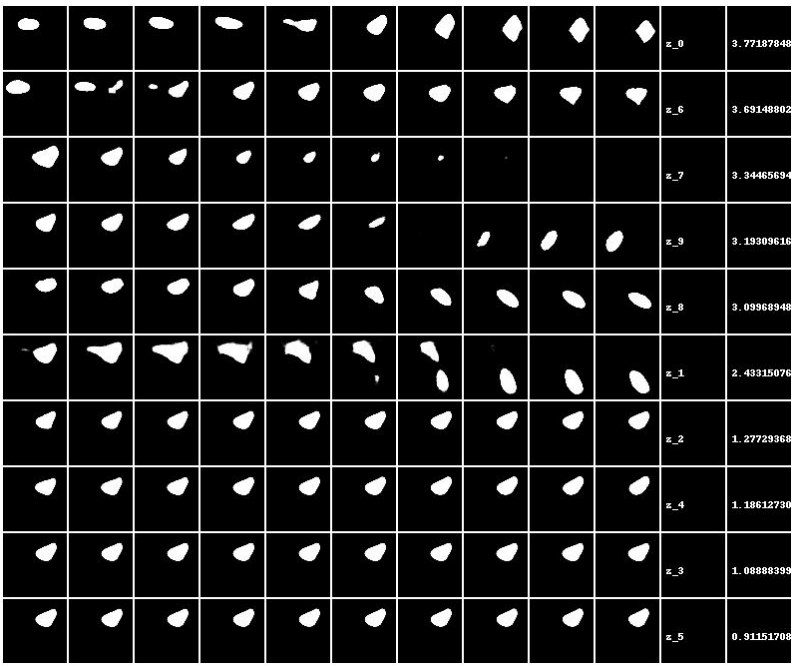


Figure 12: traversing continuous units for β-TCVAE. Last two columns refer to the unit and its KL divergence