Summary and discussion of: “Understanding disentangling in Beta-VAE”

CASIA report

Li Donghao 20653877

1 Summary

1.1 Quick summary of this paper

This paper [1] tries to present new intuitions and theoretical assessments for why β-VAE [2] could automatedly discover of interpretable factorized latent representations from raw image data in a completely unsupervised manner.

Their key hypothesis is that the different components that β-VAE found have different contributions to the reconstruction loss. Therefore, different components may align with different features which might be data generative factors.

Then, they propose a new training regime of β-VAE, that progressively increases the information capacity of the latent code during training. Their intuition is that by doing so, the most important latent variable could show up quickly while others remain hidden. Then with increasing information capacity, the contribution of the most important latent variable run out and less important latent variable start to play a role. Finally, all the potential components might be found by the model.

Comparing with standard β-VAE training, the authors claim that the modified version could offer robustness to the training procedure and do not need previous trade-off in reconstruction accuracy and disentanglement.

My code can be found at https://github.com/Lidonghao1996/MATH-5472-project1

1.2 Background Information

1.2.1 Learning disentanglement representation

It is important to introduce the goal of this paper: Learning disentanglement representation without supervised information. Intuitively, it means each dimension of the learned representation have different contributions to the data. For example, we have face image data of many people. Then a good model could learn a disentanglement representation. And each dimension have related to different features like the size of eyes, hairstyle. Moreover, if we change a single dimension of the representation, only the corresponding face feature would change. This is an important problem for artificial intelligence because it is consistent with the way that humans perceive the world. It could offer interpretability for artificial intelligence and boosting the performance of applications like reasoning about new data, zero-shot inference, and so on.

To evaluate the disentanglement, there is a commonly used technique called traversal. It means that we first get posterior latent z—x and then modify each dimension of it and keep others fixed. Then we see how the x—z changed w.r.t. the modified latent variable. If a representation is disentangled, we expect that changing only one dimension will only affect one factor of the x—z like the scale of the content or the position of the content.

1.2.2 Variational Auto-encoder (VAE)

Then we need to introduce the Variational Auto-encoder (VAE) [3] framework. The goal of VAE is to model the marginal likelihood of the data in such a generative way. Assume we have dataset x and they are generated by ground truth factors z. Then we assume that the ground truth factors z follow a standard normal distribution. The generative process is:

$$\max_{\phi,\theta} E_{q_\phi(z|x)}[logp_\theta(x|z)]$$

Where, ϕ, θ parametrize the distributions of the VAE encoder and the decoder respectively. Since we assume that the latent factors z is gaussian, we need a penalty to make z follows a normal distribution. So, the objective function can be written as:

$$Obj = D_{KL}(q(z|x)||p(z)) + L(\theta, \phi, x, z)$$

Where D_{KL} stands for Kullback–Leibler divergence. Since the output of the encoder is a parametrize distribution. The optimization process needs a reparametrization trick. It means that we can first generate a standard normal random variable ϵ and then do:

$$z_i = \mu_i + \sigma_i \epsilon$$

Then we can have a gradient for μ_i, σ_i, and updating the model parameters.

1.2.3 $\beta-$VAE

The $\beta-$VAE is a modification of VAE. The derivation is quite straight forward. It use KKT condition. The optimazition problem is:

$$\max_{\phi,\theta} E_{q_\phi(z|x)}[logp_\theta(x|z)]$$

$$s.t. D_{KL}(q(z|x)||p(z)) < \epsilon$$

Then with KKT condition, we can have:

$$F(\theta, \phi, \beta; x, z) = E_{q_\phi(z|x)}[logp_\theta(x|z)] - \beta(D_{KL}(q(z|x)||p(z)) - \epsilon)$$

Since $\beta, \epsilon \geq 0$ We can rewrite it as:

$$F(\theta, \phi, \beta; x, z) \geq L(\theta, \phi, \beta; x, z) = E_{q_\phi(z|x)}[logp_\theta(x|z)] - \beta D_{KL}(q(z|x)||p(z))$$

Here, the additional parameter β controls the capacity of the latent information channel, and it puts implicit independence pressure to the posterior distribution because the prior Gaussian distribution have diagonal covariance matrix. Then, if $\beta = 1$, it is just original VAE, while when $\beta > 1$ it puts a stronger constrain on the latent z. Empirically, this would lead to a larger degree of learned disentanglement. And this is the main motivation for this β-VAE.

1.2.4 Information bottleneck

To understand how β-VEA works, it is useful to learn information bottleneck principle:

$$max[I(Z;Y) - \beta I(X;Z)]$$

Where I(,) stands for the mutual information and β is a Lagrange multiplier. It describes a constrained optimization objective where the goal is to maximize the mutual information between the latent bottleneck Z and the task Y while discarding all the irrelevant information about Y that might be present in the input X. Task Y here is the reconstruction of the image in the context of VAE.

1.3 Summary of intuition of this paper

1.3.1 β-VAE through the information bottleneck perspective

The paper tries to understand the extra pressure caused by $\beta >> 1$. The KL term pushes the latent z towards Gaussian distribution with a diagonal covariance matrix. And the reparametrization process is adding a Gaussian noise to the latent μ. In the information-theoretic perspective, we can view the encoder is a noisy information transmitting process, and the KL term is an upper bound of the amount of transmitted information. When KL term is zero, it gives every sample a latent standard Gaussian distribution and there is no information transmitted. If the encoder wants to transmit some information, the conditional distribution must be away from the prior distribution of z. And that could be achieved by increasing the mean of the posterior or reducing the variance of the posterior. If there are no constraints on KL term, the VAE will have degenerated into Autoencoder which has deterministic latent $z|x$.

1.3.2 Why β-VAE better than VAE in disentangled representation

The paper has explained a lot to answer this problem, but the idea is not hard to understand. There are two key ideas. The first one is that adding a harder pressure makes the posterior $z|x$ tends to overlap. Since the prior is standard normal. Then large β will make the variance goes to I and the mean goes to 0. That will make the posterior for different x have an overlap that makes the decoder hard to construct the image correctly. Next is that I think that this paper assumes that the generator factor is a concise coding for the data. So, once we have a larger pressure for the KL term, the model should be more carefully arrange the encoded space and make sure the overlapping is smallest while the posterior variance and mean is very close to I and zero. Then if the latent z corresponding to the generator factor, it can be minimized by the second assumption. Thus, the model could learn a disentangled representation.

If there is not enough pressure, the encoding process is much easy since the posterior could be very concentrated and there is no need for the encoder to learn a disentangled representation which is concise.

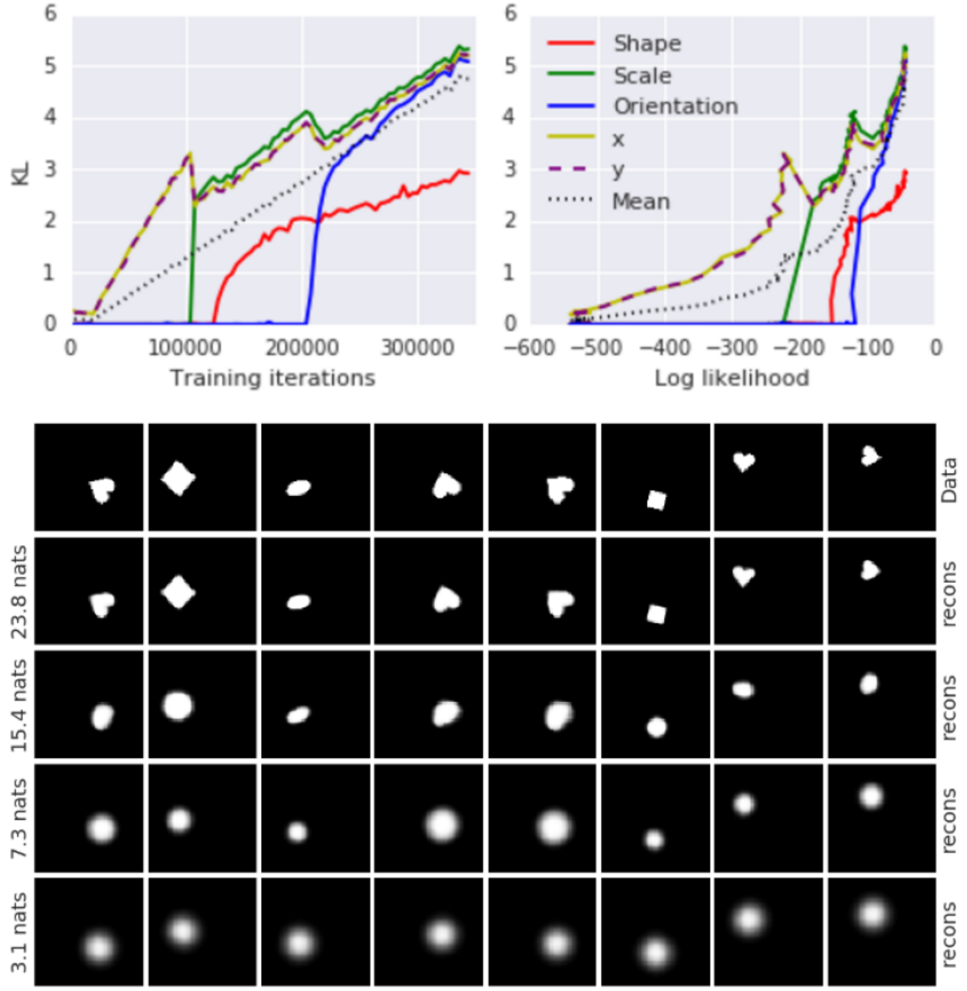


Figure 1: Motivating Example in the paper: reconstruct dSprites data with ground truth factors. (Figure 3 in the paper) Note that it just uses ground truth factors to reconstruct dSprites data instead of training a VAE. So it just a motivating example. It shows the training process with a restricted information bottleneck. The top left figure shows that when the information increased in the training process, the first two factors that are learned are location x and y while the others stay undiscovered. As the training process goes on, the scale factor is learned (the green curve). Finally, when the information constrains are not very strict, all the factors will be learned. The bottom figure shows some reconstruction images to give us an intuitive feeling.

1.3.3 Improving disentangling with controlled capacity increase

In the last section, the paper explains thy β-VAE better than VAE in disentangled representation. Then they try to improve the training process of the β-VAE. Their key hypothesis is that β-VAE finds latent components that make different contributions to the log-likelihood term of the cost function. And different generative factors also have different contributions to the generated data. If we correspond them together, the learned latent representation might be more disentangled.

Then, how to correspond to them together? The paper suggests that we can control the information encoded in the posterior. Then the model would only learn the important information and discard the others since there exists a information constrain. Then paper gives an example (See Figure1) of reconstructing the dSprites data. This data is an image generated by some base shapes like heart, square using relocation, rescale, rotation. Then we can easily figure out that the location is the most important factor for the log-likelihood. The model also finds this and when the information transmitted is limited (3.1 nats) the model only learns the location factor and discards others. Then as we increase the information, the model learns the scale. Finally, the model learns the final latent components rotation.

This is an example just show the idea of the modification of the training process, that first constrains the information transmitted within a very low level to force the model to learn the latent component that has the largest impact of the reconstruction. Then gradually relax the constraint and let the model learn some other latent components.

Finally, the paper modifies the training process with changing the objective function as follows:

$$L(\theta, \phi, \beta; C, z) = E_{q_\phi(z|x)}[logp_\theta(x|z)] - \gamma(D_{KL}(q(z|x)||p(z)) - C)$$

Where C is the expected KL divergence in the training process and it increases from a small value (0.5 nats) to a high value (25 nats) linearly. Then to make the real KL divergence very closed to the expected C, γ is usually very large (like 1000).

2 Result and Discussion

This the part, I will show my implementation of the β-VAE. First I will show reproduced results of the paper. Then show some detailed information that the paper does not mention. Also, I will compare the original VAE and β-VAE that the paper does not show as well.

The implementation is not very easy and I have met some troubles, I will also point them out. To deal with the problems, I also tried to modify the algorithm, some of the modification indeed helps. I will explain them later.

2.1 Reproduce results

The main experiments in the paper train a VAE and show its disentanglement by traversal. I trained models on dSprites, CelebA, and 3DChairs dataset following the hyperparameter setting in the paper. See Figure 2 , Figure3, Figure4.

2.2 During my own implementation

This paper uses deep neural networks to model the encoder and decoder, so it is not very easy to implement, it might need careful parameter tuning and also some small changes to the code may lead to an unsuccessful reproduction. Actually, I make a small mistake during coding: I choose to reduce the loss of reconstruction by taking the mean of the whole image dimensions, while the proper implementation should take the sum of all the dimensions. Actually, it just a scale difference for the reconstruction loss and would not

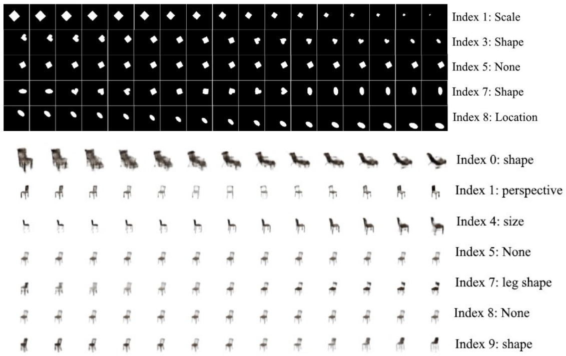


Figure 2: These two figures show effect of changing single latent variable (traversal) for dSprites (Top) and 3DChairs (Bottom) dataset. We can clearly see that the latent variables indeed have some relationship with generating factors.

affect the minimum of the model. However, after combining with the KL-divergence term, the trade-off of them is changed. The model would consider the reconstruction loss less (by 64*64*3 times).

Under that circumstance, the model will tend to output a mean image of the data since the posterior is just standard normal for all the samples. So, I check the mean for posterior and realize it is always zero while the log variance goes to negative infinity. That means only the variance part contributes to the KL-divergence part and the result is that the input of the decoder is just a zero vector so it can only output a mean image to minimize the reconstruction loss.

2.3 Why penalize small KL-divergence? Try Hinge loss?

When I did not realize that, I tried to rethink the constraints on the KL-divergence. In the paper, it uses absolute value to control the KL-divergence. However, I think that it is not reasonable to penalize small KL-divergence, and Hinge loss like penalty might be more suitable. Because when the encoder wants to minimize the KL-divergence, there are two ways: decreasing the variance or make mean value away from 0. Decreasing the variance is feasible for all the samples while for the sample with positive mean the model should make a mean increase and for others should decrease the mean. So when we apply too tight constrain on small KL-divergence, the model tends to decrease the variance instead of making mean away from 0. That is not a good way of making use of limited KL-divergence. (See Figure 6)

So, I have tried this hinge loss like penalty and found that it helps the training process of the mistake setting. Although the output has limited improvement it still like an average image, the posterior mean is not always 0 anymore.

Would this help the training for a correctly implemented model? I have also tried hinge loss and found that this indeed helps to prevent log variance goes to negative infinity. Since now the reconstruction loss dominant, this effect is not very obvious, However, I also find it might help the separation of different KL-Divergence training paths. (See Figure 7) I think separation paths is very important and is the intuition of this paper. So, it is important to use hinge loss like penalty to control the KL-Divergence.

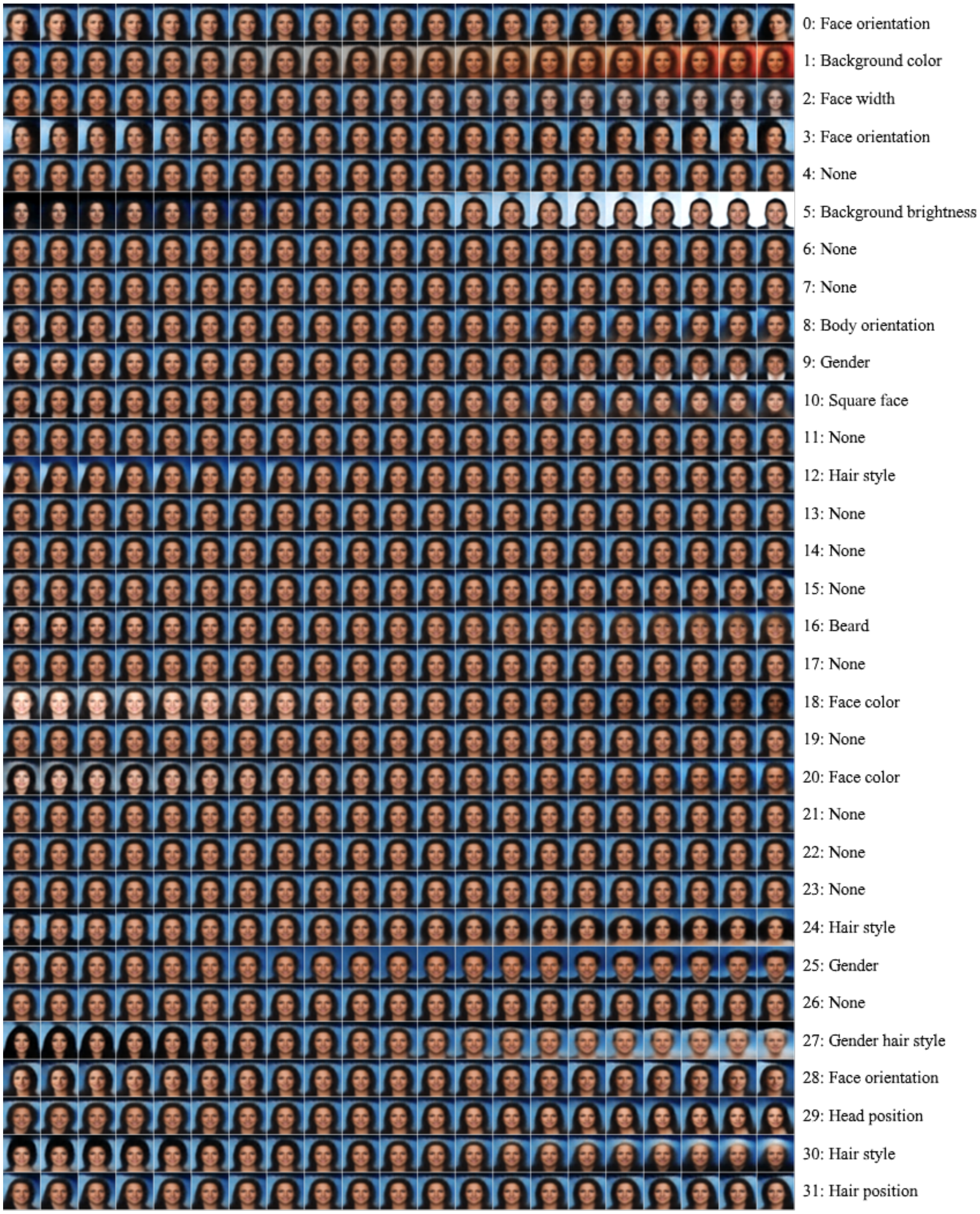


Figure 3: This figure shows the traversal of CelebA dataset and for latent all variables. We can see that many of the latent variables represent none of the features. And the other latent variables indeed learns a disentangled representation that changing only one feature will not affect other factors of the output image.

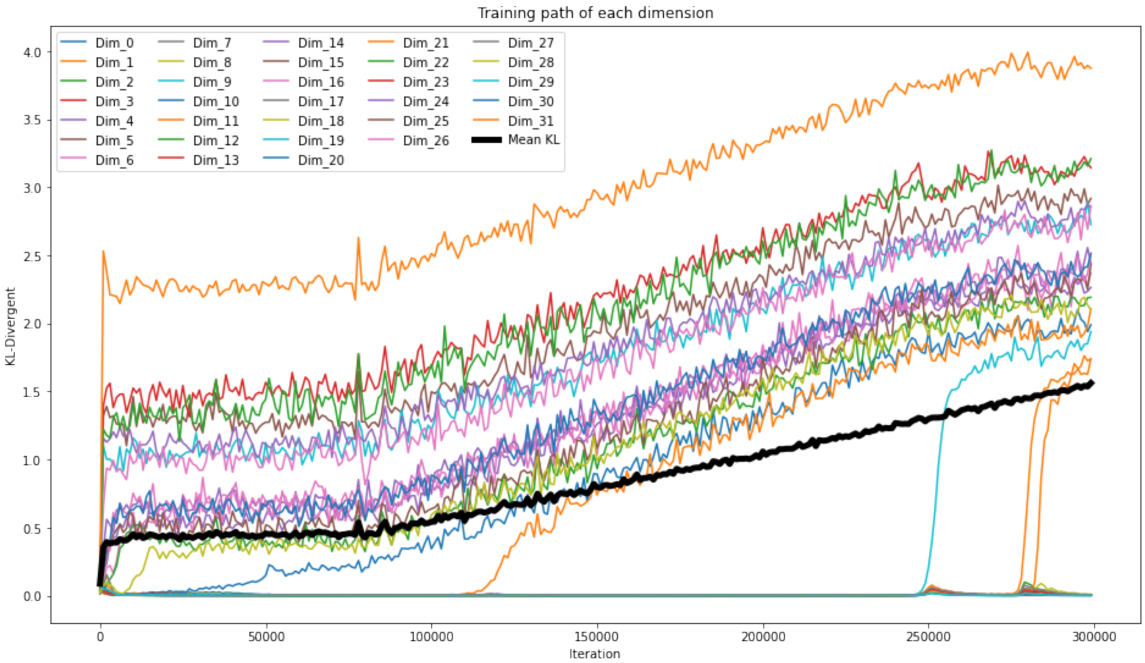
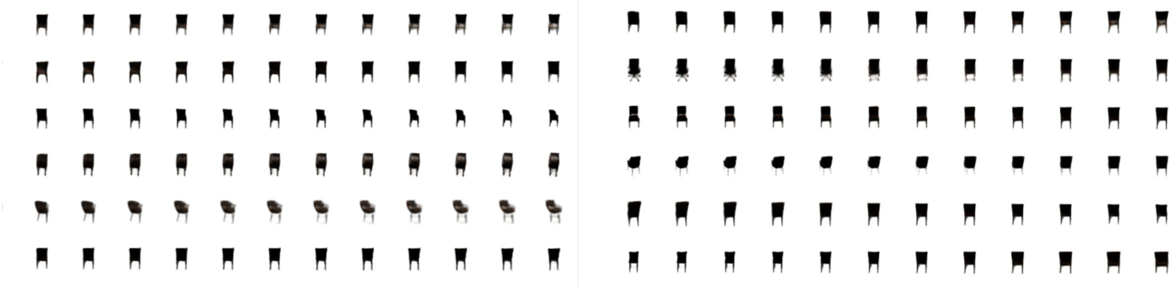


Figure 4: This figure shows the training process of CelebA dataset for latent variables. traversal figures are too large to show the training process together. We can see that there are many dimension with zero KL-divergence, they are corresponding to None features. Then, others are non-zeros and it seems that most of them jump out of zero at the very beginning and gradually increase as the training goes on.



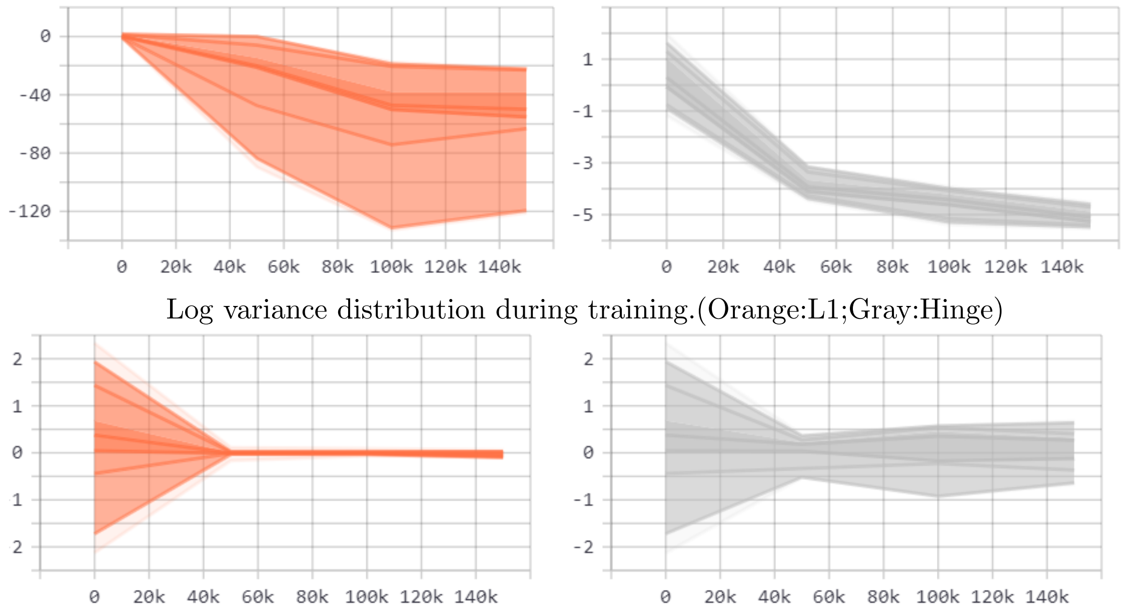
(a) VAE

(b) β-VAE

Figure 5: This two figure shows traversal for VAE and β-VAE model on CelebA dataset. We can see that the *beta*-VAE model is more sensitive to traversal and shows some kind of disentanglement features like the second row only affect the legs of chair.

2.4 Insight and discussion

This paper improved the training process by control the information transmitted by the posterior. In particular, during the training process, it increases the expected KL-divergence and under a fixed KL-divergence constrain, the model tried to only capture the dominating factors and discard others. So, the model might learn a disentangled representation. If we treat the KL-divergence as a regularization term (like L1 or L2 regularization), the modified



Log variance distribution during training.(Orange:L1;Gray:Hinge)

Mean distribution during training. (Orange:L1;Gray:Hinge)

Figure 6: Adding L1 loss could make mean goes to zero while hinge loss like penalty may solve this problem. We can see that the penalty on small KL-divergence would make mean stay zero.

training process looks like give a solution path for the task. Actually, in the motivating example, it looks like a LASSO solution path. And if we release the L1 penalty during training in linear models like linear regression, It would also give us a solution path. I think the difference is that training a VAE is a non-convex problem, so the final solution to the optimization problem might be affected by the changing of constraint during training. While in convex cases, if we have enough iterations under each constrain, then we can find the optimal solution under each constraint and the final solution might not be affected by the training process, which means it will be the same as just giving a fixed very loose constraint.

3 Conclusion

In this report, I summarize and reproduce [1]. I found that this could learn a disentangled representation with unsupervised training. However, I find that the original loss to control KL-Divergence may not be reasonable. So, I modified it into a hinge loss like term and find it is better in terms of improving the degree of separation of different KL-Divergence training paths. In particular, that means the information transmitted of each dimension becomes non-zeros in the training process instead of at the very beginning.

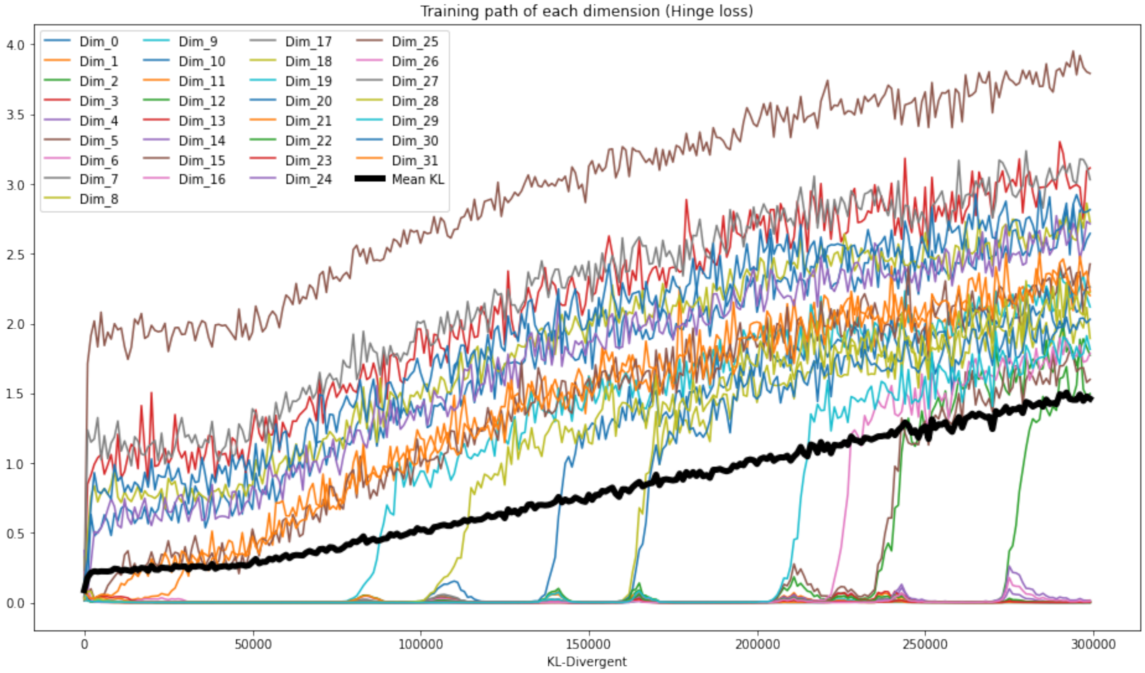


Figure 7: This figure shows the training process of CelebA dataset for latent variables with hinge loss. We can see that comparing with the original one, There are more variables jump out of zero in the training process instead of at very beginning. That is consistent with the motivating example and the intuition of this paper. I think it is reasonable not to penalize small KL-divergence.

References

[1] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae, 2018.

[2] 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. *Iclr*, 2(5):6, 2017.

[3] Diederik P Kingma and Max Welling. Auto encoding variational bayes, 2013.