|  |  |  |
| --- | --- | --- |
|  | به نام خدا |  |
| **دانشگاه تهران**  **دانشکده‌ مهندسی برق و کامپیوتر**  **مدل‌های مولد عمیق**  **تمرین اول** | | |

|  |  |
| --- | --- |
| سیدرضا مسلمی | نام و نام خانوادگی |
| 810103326 | شماره‌ دانشجویی |
| 12 آبان 1403 | تاریخ ارسال گزارش |

فهرست

[سؤال اول 3](#_Toc181475156)

[بخش اول 3](#_Toc181475157)

[بخش دوم 3](#_Toc181475158)

[بخش سوم 4](#_Toc181475159)

[بخش چهارم 4](#_Toc181475160)

[بخش پنجم 5](#_Toc181475161)

[سؤال دوم 6](#_Toc181475162)

[بخش اول 6](#_Toc181475163)

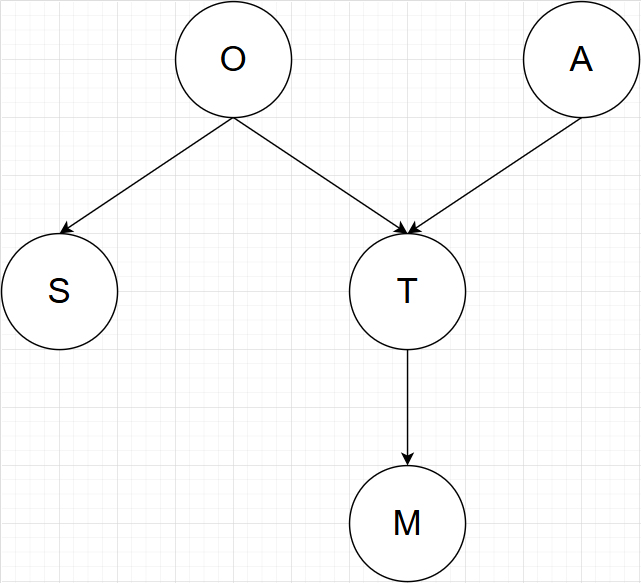
[بخش دوم 11](#_Toc181475164)

[مراجع 15](#_Toc181475165)

# سؤال اول

## بخش اول





1. P(O, A, T, S, M) = P(O) P(A) P(S|O) P(T|O, A) P(M|T)
2. 1. True, O can not flow through T, since it is in v-structure form and is unobserved.
   2. False, O can flow through T, since it is in v-structure form and its child has been observed.
   3. False, S can flow through S <= O => T => M, since O and T are unobserved.
   4. True, S can not flow through S <= O => T => M, since O is observed.

## بخش دوم

1. False, I can flow through I, G, C, E, F, since they all are unobserved.
2. False, B can Flow through B, C, G , I, since they all are unobserved.
3. True, A can not flow through C, since it has been observed.
4. True,

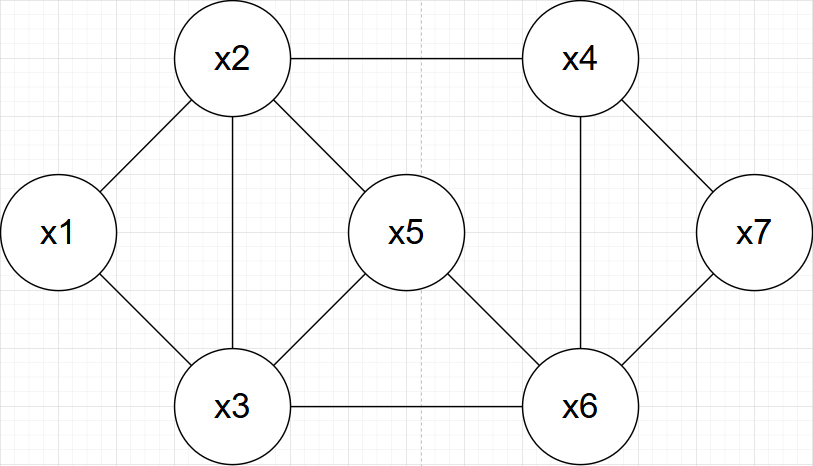
p(b|cdf) = p(b|cd) = p(bcdf)/(p(c)p(d|c)p(f|cd))

p(f|bcd) = p(bcdf)/(p(c)p(d|c)p(b|cd)) => p(b|cd) = p(bcdf)/(p(c)p(d|c)p(f|bcd))

p(f|cd) = p(f|bcd) --> True

## بخش سوم

1. P(x1,...,6) = p(x1) p(x2|x1) p(x3|x1) p(x4|x2) p(x5|x2, x3) p(x6|x3, x5) p(x7|x4, x6)
2. x3, x5, x4, x7



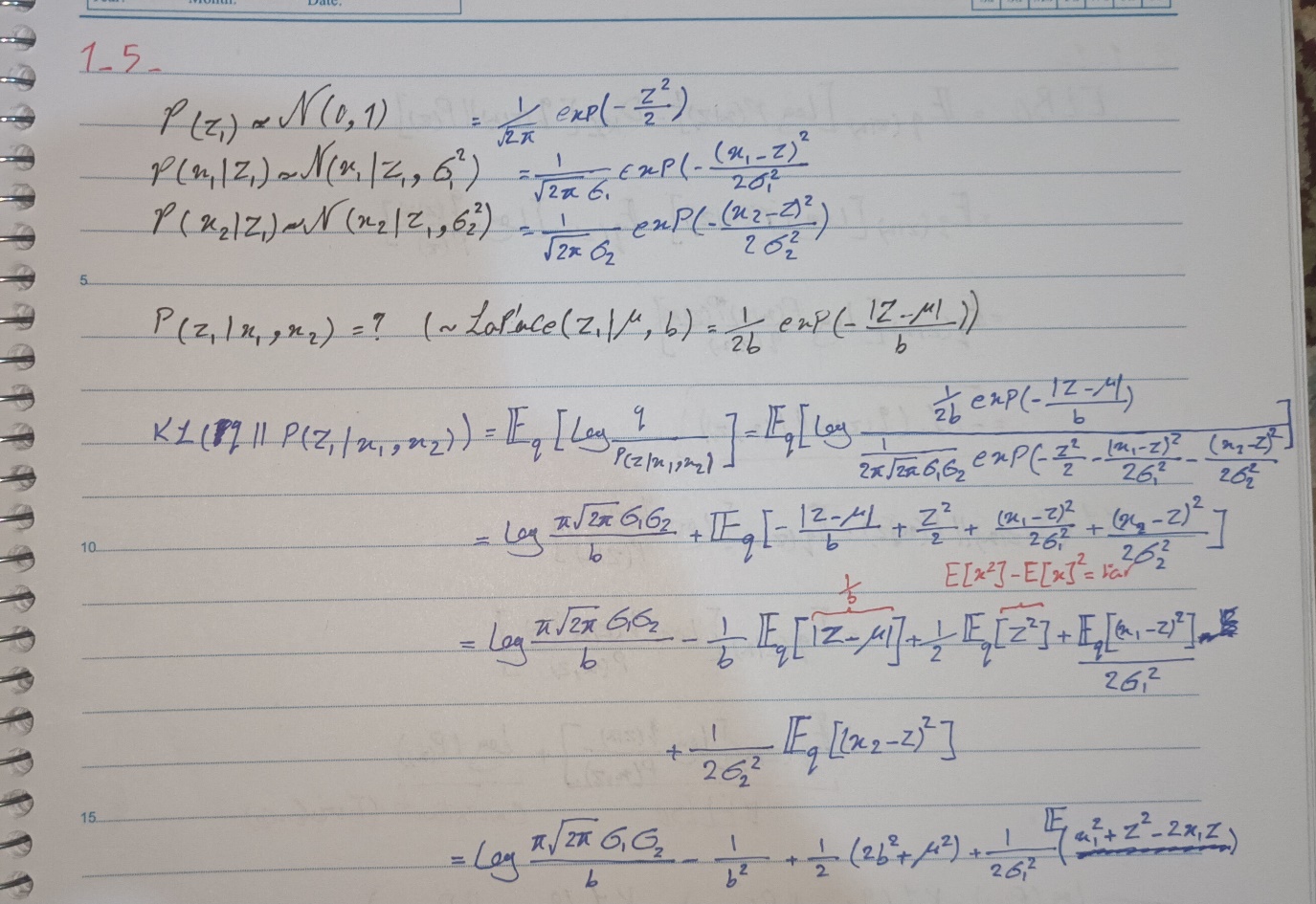
1. No. Since it contains additional edge related to v-structure.
2. No. Because there exists a loop with length of 4.

## بخش چهارم

1. Since G is a perfect I-map for L, it covers all and only the conditions in L.
2. Removing an edge from G, the new graph G’ might introduce new independencies not exist in L.
3. Therefore, G’ is not an I-map for L, because it is not a subset of L.

For both Bayesian networks and Markov networks, removing an edge from a graph G that is a perfect I-map for an independence list L results in a new graph G’ that is not an I-map for L.

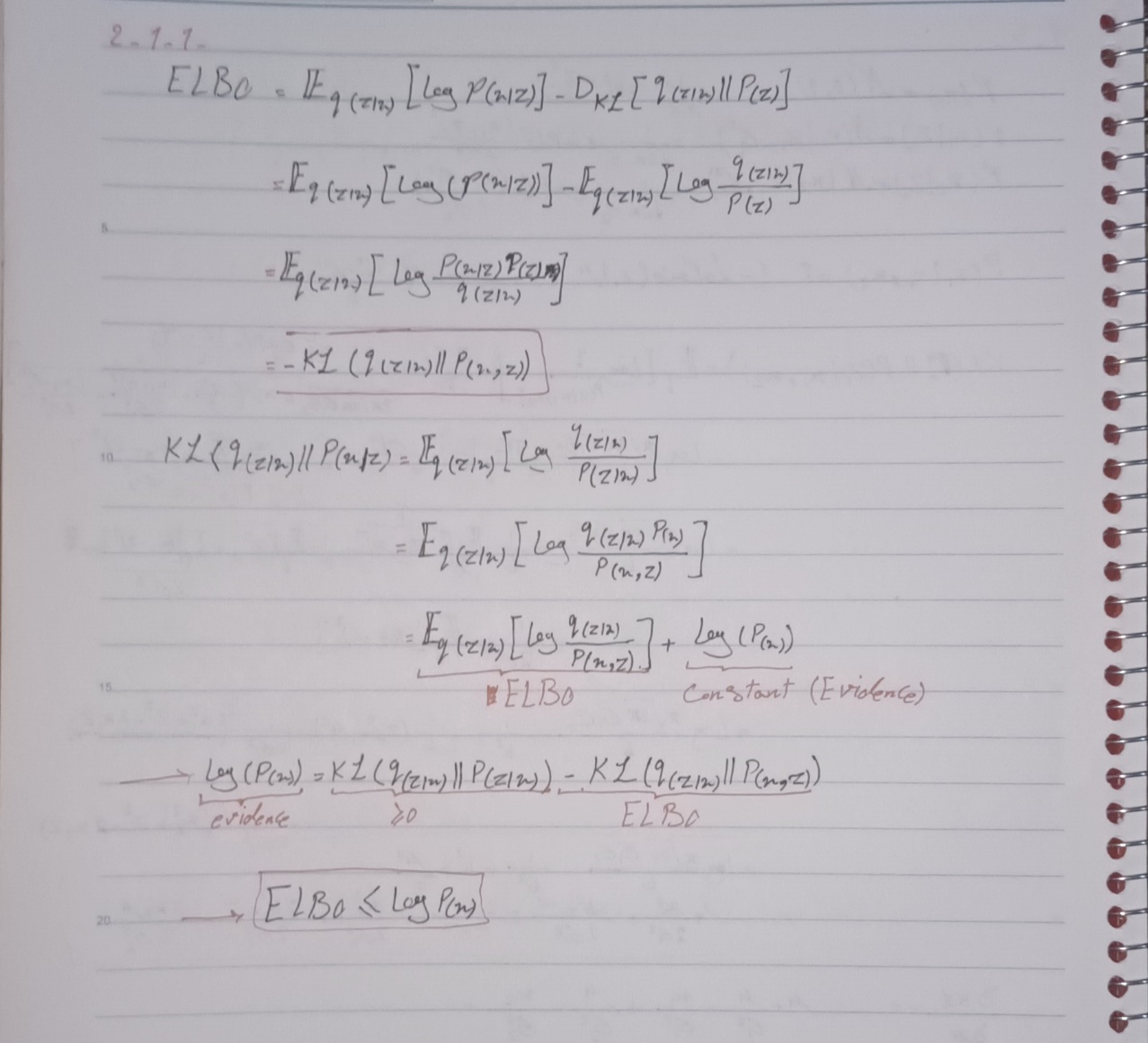
## بخش پنجم

A close-up of a paper with mathematical equations

Description automatically generated

# سؤال دوم

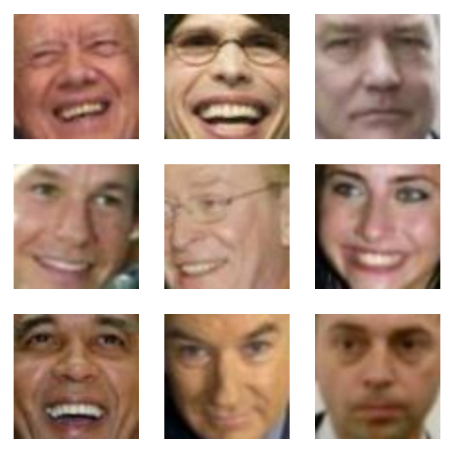
## بخش اول



In variational Autoencoders, minimizing the KL term encourages the learned latent distribution *q(z|x)* (posterior) to be close to a prior distribution *p(z)*, usually a standard Gaussian *N(0,I)*. This alighnment leads to two main benefits:

* 1. **Regularization:** By making q(z∣x) close to p(z), the model avoids overfitting.
  2. **Sampling Quality:** An aligned latent space means that samples from the prior *p(z)* will produce qualified outputs using the decoder.

1. Reparameterization trick. Since sampling directly from q(z∣x) (a Gaussian) breaks the backpropagation chain, the trick transforms this sampling process. This trick separates the randomness from the parameters, making it possible to train the VAE using standard gradient descent methods.
   1. **Latent Variable Sampling:** Assume *z∼N(μ,σ2)*, where μ and σ are outputs of the encoder network.
   2. **Reparameterization:** We express *z* as *z=μ+σ\*ϵ* Where *ϵ∼N(0,1)* is a random variable sampled from a standard normal distribution.
   3. Using this transformation, *z* is now a differentiable function of *μ* and *σ*, allowing the VAE to backpropagate during optimization.



* 1. Structure: vae.py, representations: representor.py, training process: main.py, Dataloader: dataloader.py

A graph showing a line

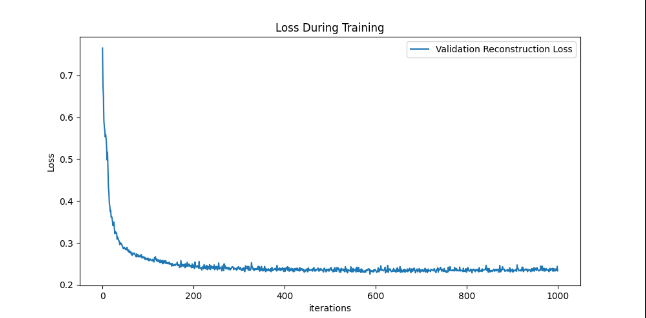
Description automatically generated

A graph showing loss during training

Description automatically generated

A graph showing a loss

Description automatically generated



A graph showing loss of training

Description automatically generated

A graph with blue lines

Description automatically generated

* 1. Decoded from real samples:

A collage of different faces

Description automatically generated

With Random noise:

A collage of different faces

Description automatically generated

* 1. Adjusment of smile:

A close-up of a person's face

Description automatically generated

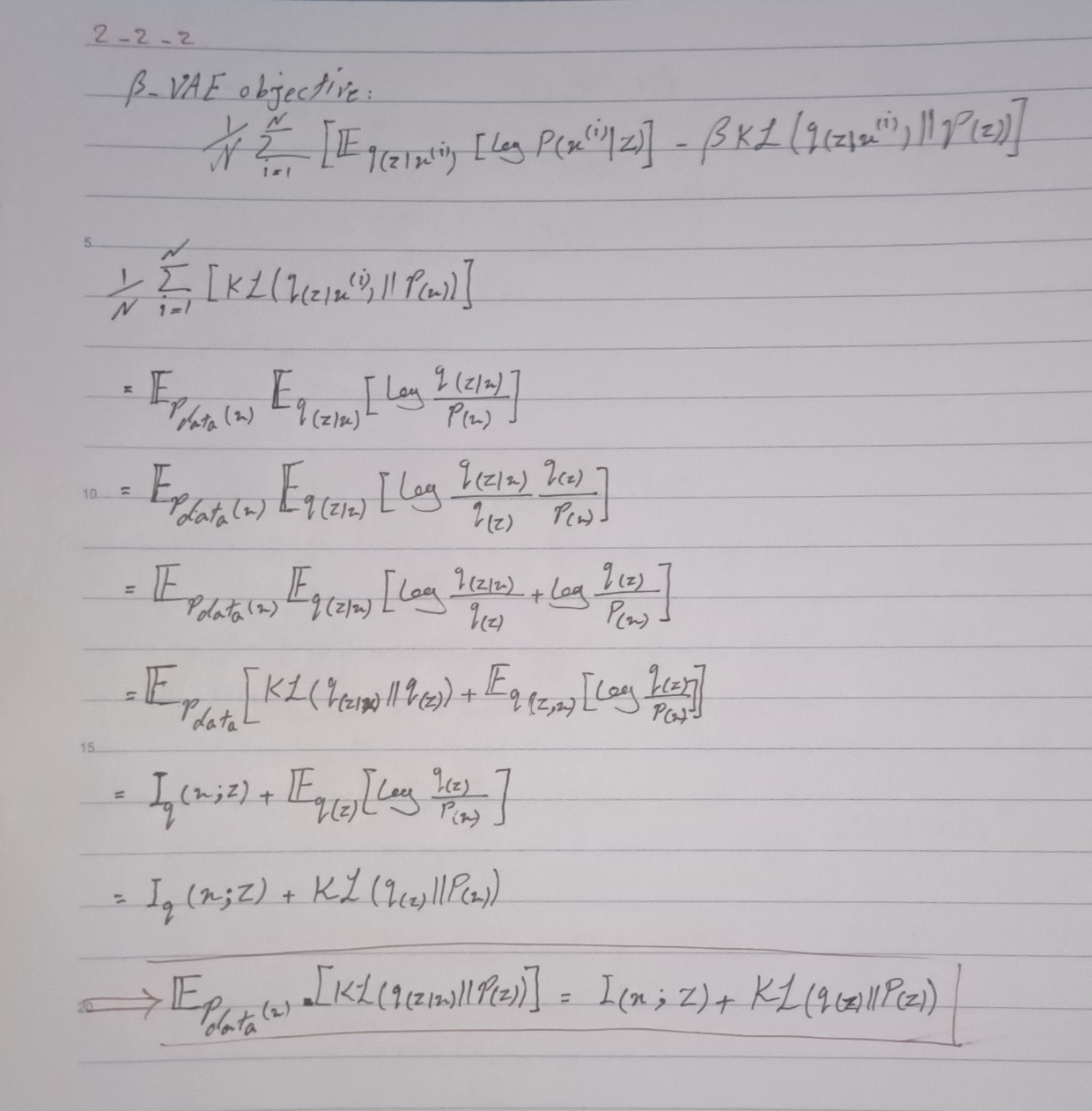
A collage of a person's face

Description automatically generated

## بخش دوم

1. We assume this as a constrained optimizing problem, where we maximize the reconstruction likelihood while constraining the KL:  
   *maximize\_q(z|x) E\_q(z|x)[ log p(x|z) ]* ***s.t.*** *D\_KL( q(z|x) || p(z) ) <= delta*  
   Convert the constrained problem to an unconstrained problem by introducing a Lagrange multiplier, beta:  
   *F = E\_q(z|x)[ log p(x|z) ] - beta \* ( D\_KL( q(z|x) || p(z) ) - delta )*Expanding the terms:  
   *F = E\_q(z|x)[ log p(x|z) ] - beta \* D\_KL( q(z|x) || p(z) ) + beta \* delta*  
   To ensure optimality, we apply the KKT conditions:  
   A math equations and formulas

   Description automatically generated with medium confidence  
     
   If *D\_KL( q(z|x) || p(z) ) = delta*: beta is non-zero, keeping the KL term active  
   else if *D\_KL( q(z|x) || p(z) ) < delta*: beta = 0, which leads to standard VAE loss without scaling  
     
   Since beta and delta >= 0 (according to the complementary slackness), F can be rewritten to arrive the beta-VAE formulation, but with additional beta coefficient:  
   *F >= L = E\_q(z|x)[ log p(x|z) ] - beta \* D\_KL( q(z|x) || p(z) )*
2. In FactorVAE, the KL term from β-VAE is decomposed into two parts: mutual information (which makes sure the latents capture relevant information about *x*) and KL divergence between *q(z)* and *p(z)* (which encourages independence among latent dimensions). To approximate *KL(q(z)||p(z))*, FactorVAE adds a discriminator network to distinguish between samples from aggregated posterior *(q(z))* and prior *(p(x))*.



1. In the Factor-VAE, the main objective includes gamma-weighted penalty for the discrepancy between the aggregated posterior *q(z) = 1/n(∑(i=1:n) q(z|xi)* and the marginal distribution *q’(z) = ∏(j=1:d) q(zj)* where d is dimention of latent space, which requiers a heavy computation. To address this challange, authors suggest two methods to approximate the distribution and symplify the calculation:
   1. Density-ratio Trick (using a Discriminator):  
      Instead of directly calculaing the KL between q(z) and q’(z), thay use a ratio-trick, introducing a discriminator D which is trained to recognize samples from q(z) or q’(z). The discriminator’s output is an estimation of density ratio between q(z) and q’(z), then the KL is approximated by cross-entropy loss of the discriminator.
   2. Minimizing a GAN-like Objective:  
      This approximation assumes the problem is GAN-like. This approach reduces the direct computation of integrals and instead optimizes the adversarial loss.

Eventually the authers choose the Density-Ratio Trick using a Discriminator approach.

# مراجع

1. Chat.openAI (for simple representations and docstrings)
2. <https://github.com/rishabhd786/VAE-GAN-PYTORCH/blob/master/models.py>
3. <https://kvfrans.com/deriving-the-kl/>