Generative Modeling

Variational Autoencoders

Denis Derkach, Artem Ryzhikov, Sergei Popov

Laboratory for methods of big data analysis

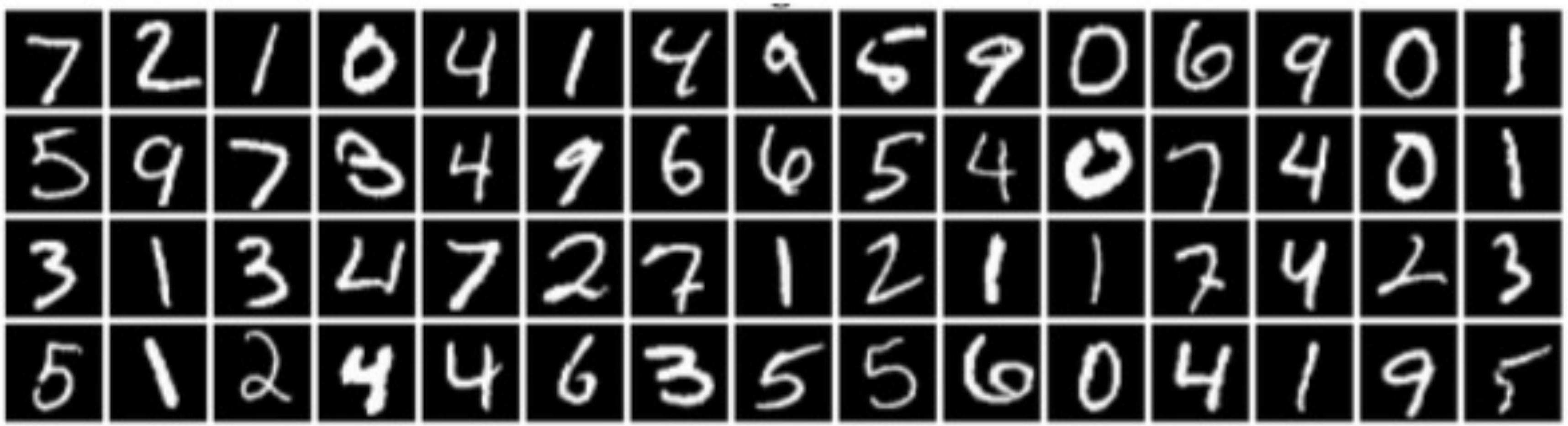
Spring 2023

In this Lecture

- Autoencoders.
- Gaussian Mixture Models.
- Variational Autoencoders:
  - Construction.
  - ELBO.
  - Problems.

Idea

Latent variables - MNIST

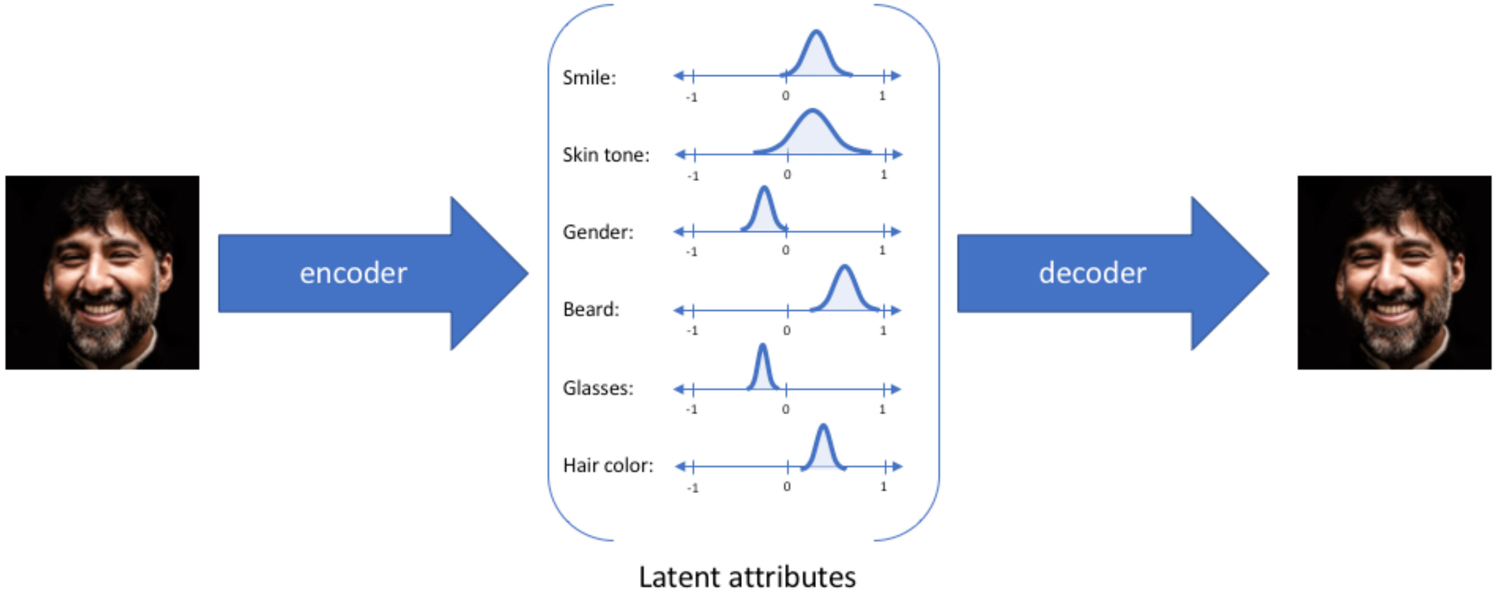


What is common in these images? How can they be characterized?

Latent Variables - Celeb

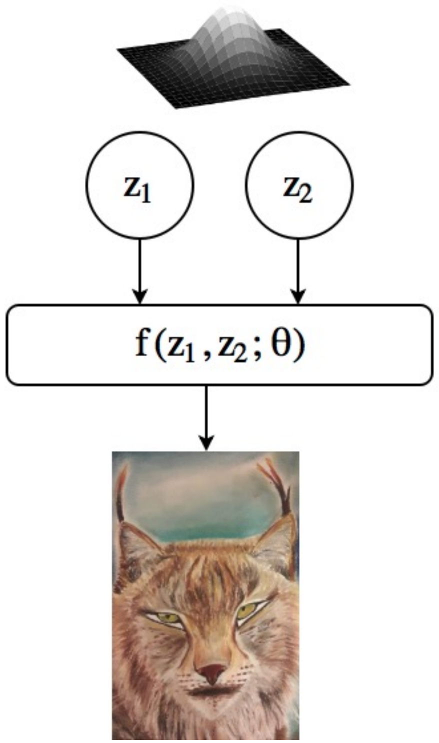
Variability due to eye and hair color, gender, race. However, unless images are annotated, these factors of variation are not explicitly available (latent).

Latent Variables - Celeb



Variability due to eye and hair color, gender, race. However, unless images are annotated, these factors of variation are not explicitly available (latent).

Latent Variable Models: Motivation



- **Observe** X in the data sample. Latent observables Z are hidden.
- Knowing latent variable space correctly can be beneficial as $p(x|z)$ can be described **simpler** than $p(x)$.
- **Hard** to find the real Z space manually, we need unsupervised learning.

Gaussian Mixture Model

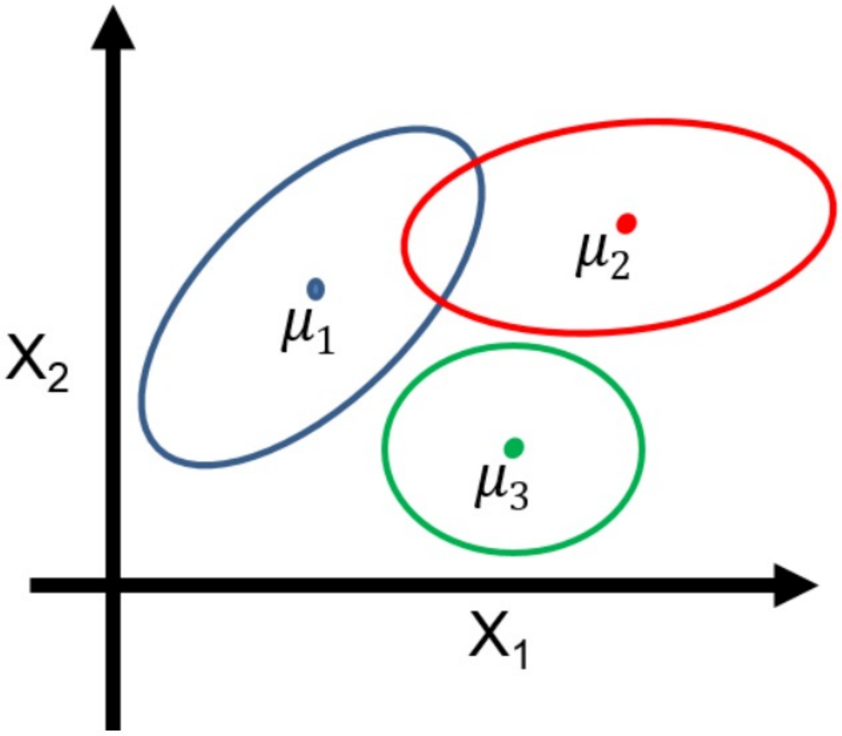
We can use a mixture of distributions (like, Gaussians) to have a better estimate of true PDF. Gaussian Mixture Model proposes:

$$p_{GMM} = \sum_{l=1}^{K} \pi_l \phi(x; \mu_l, \sigma_l^2),$$

where $\pi_l \geq 0$ are the weights: $\sum \pi_l = 1$. Here the number K is a tuning parameter that specifies the number of Gaussians in our model.

Mixture of Gaussians as Latent Variable Model

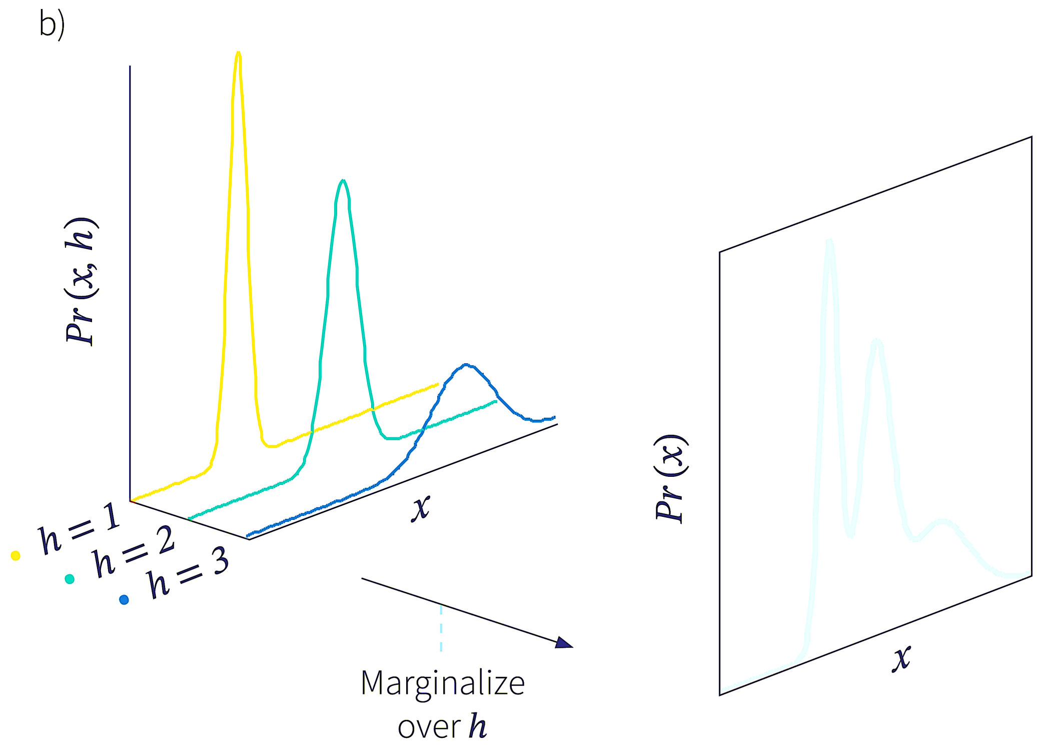
- GMM is a latent variable model.
  - $z \sim \text{Categorical}(1, K)$
  - $p(x|z = k) = N(\mu_k, \Sigma_k)$



- Generative process
  - Pick a mixture component k by sampling z.
  - Generate a data point by sampling from that Gaussian.

Combination of $p(x)$

- Combine simple models into a more complex and expressive one.
- Obtain PDF using simple formula.



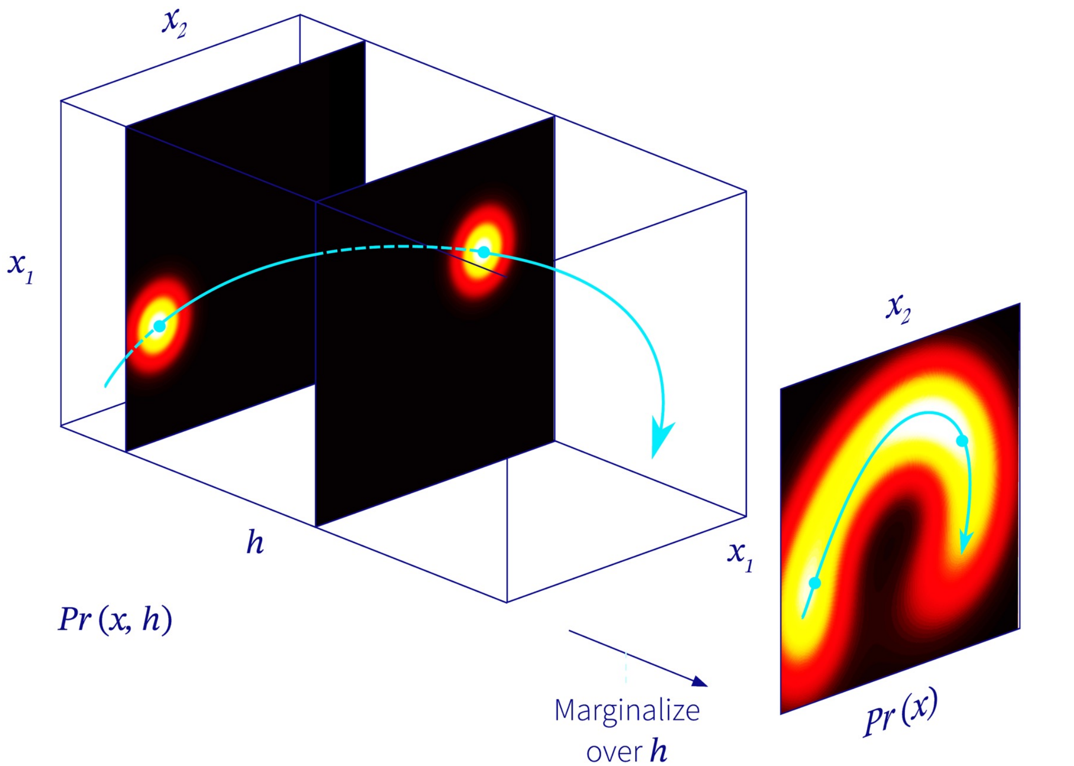
$$p(x) = \sum_{all\ z} p(x,z) = \sum_{all\ z} p(z)p(x|z) = \sum_{k=1}^{K} p(z=k)N(x;\mu_k,\Sigma_k)$$

Combination of $p(x)$

- Combine simple models into a more complex and expressive one.

- Obtain PDF using simple formula.

- What if we have a continuous observable?



$$p(x) = \sum_{all\ z} p(x,z) = \sum_{all\ z} p(z)p(x|z) = \sum_{k=1}^{K} p(z=k)N(x;\mu_k,\Sigma_k)$$

$$p(x;\theta) = \int_z p(x,z;\theta)dz = \int_z p(x;\theta|z)p(z)dz$$

Finding parameters θ

How do we find the parameters θ? The most natural way is to maximise the log likelihood:

$$\log \prod_{i=1}^{D} p(x^i;\theta) = \sum_{x\in D} \log p(x;\theta)$$

In the following we will need to:

- define a loss function to optimize
- calculate gradients of the loss function with respect to the parameters θ.
- apply a variant of stochastic gradient descent.

Maximizing the likelihood

Consider one sample, x. Then to maximize the likelihood, we need to calculate:

$$p(x;\theta) = \int_z p(x,z;\theta)dz = \int_z p(x;\theta|z)p(z)dz$$

- Easy to calculate in special cases:
  - in case of independent observables $x|z \perp z$.
  - special cases of conjugation.
- In general, x depends heavily on z (by construction).
- Integral is intractable.

Estimating likelihood: Naive Monte-Carlo

We can use the same algorithm as we have seen in GMM:

$$p(x;\theta) = \sum_{all z} p(x, z; \theta) = |Z| \mathbb{E}_{z \sim Uniform(z)}(p(x, z; \theta)).$$

For this, we can fix the distribution followed by z.
This is really only tractable when x is relatively low-dimensional. In case of image, we run into the curse of dimensionality, where we need to grab many samples to get an accurate view of x.

Wrap-up

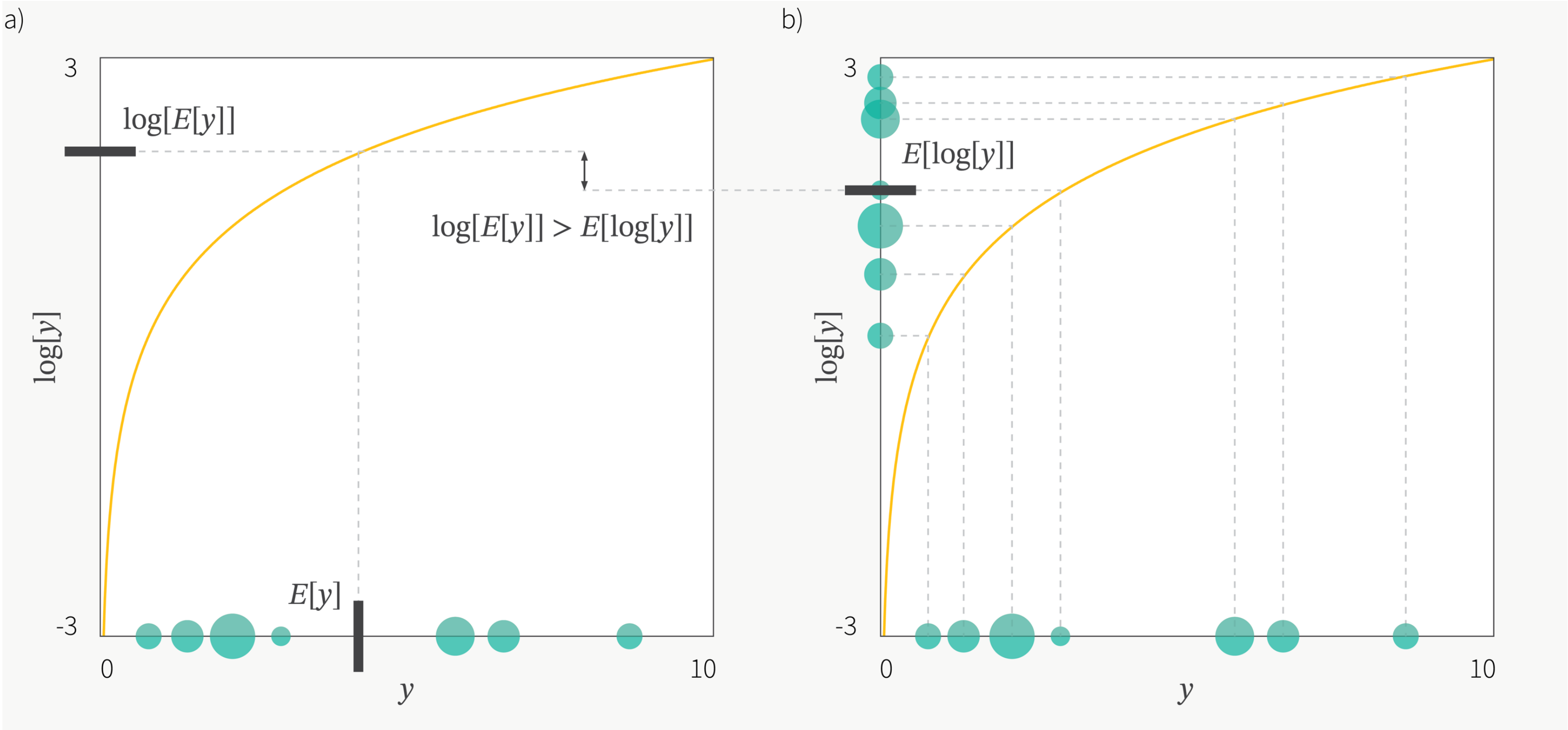
- Problems:
  - We can't calculate p(x).
  - We can't write the maximum-likelihood objective.
- We need a different approach or objective function. Idea:
  - if we can't write likelihood, let's instead derive a lower bound on it.
  - If the lower bound is tractable, then we can optimize the parameters with respect to the lower bound.
  - If we are making the lower bound larger, we are making the likelihood larger.

Evidence Lower Bound

Jensen's inequality

For concave function g(.):

$$g[\mathbb{E}[y]] \geq \mathbb{E}[g[y]].$$



$$\log \int p(y) y dy \geq \int p(y) \log y \, dy \,.$$

Figure: BorelisAI

Deriving the Evidence Lower BOund

- Start multiplying and dividing $q(z)$ (arbitrary function):

$$\log p(x;\theta) = \log \int p(x,z;\theta)dz \; = \log \int q(z)\frac{p(x,z;\theta)}{q(z)}dz\,.$$

- Jensen's inequality for convex functions (log):

$$\log \int q(z)\frac{p(x,z;\theta)}{q(z)}dz \geq \int \boldsymbol{q(z)\log\frac{p(x,z;\theta)}{q(z)}dz}\,.$$

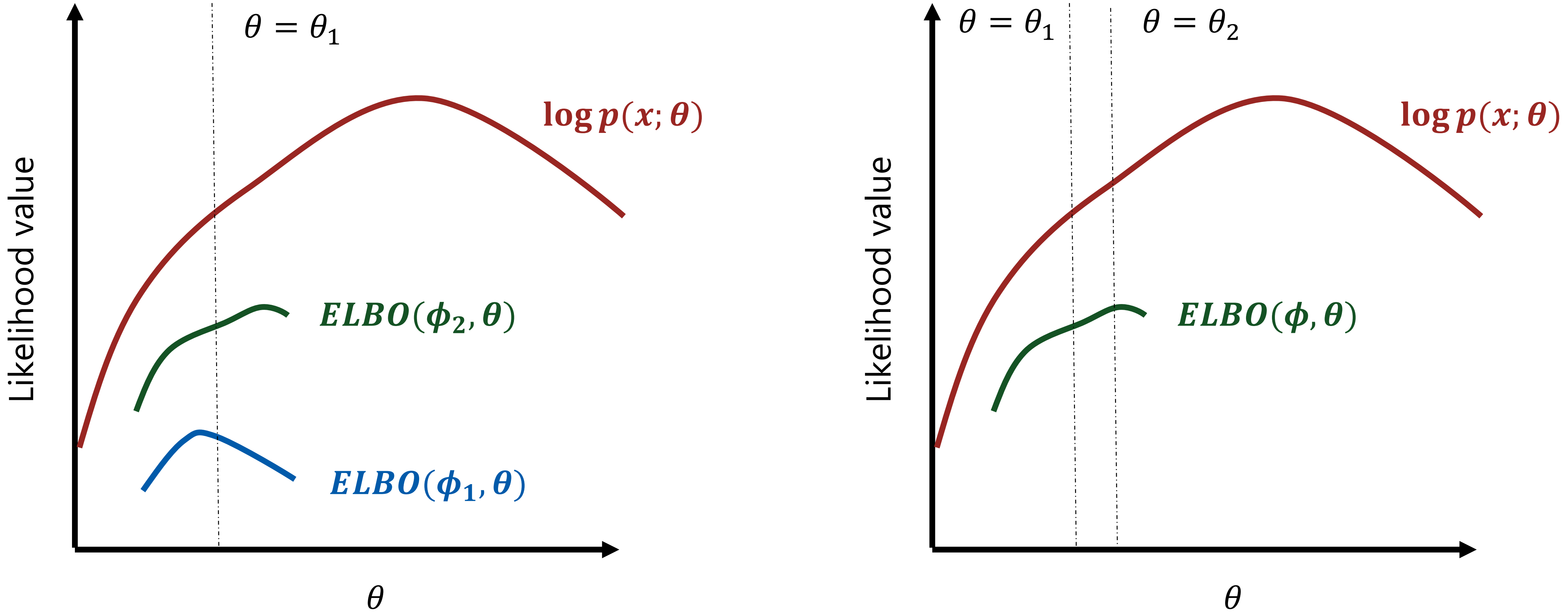
The rhs is **evidence lower bound**

- $q(z)$ has different set of parameters, ϕ

$$ELBO(\phi,\theta) = \int q(z;\phi)\log\frac{p(x,z;\theta)}{q(z;\phi)}dz$$

ELBO so far

$$ELBO(\phi, \theta) = \int q(z;\phi) \log \frac{p(x,z;\theta)}{q(z;\phi)} dz$$



Tightness of bound

$$ELBO(\phi, \theta) = \int q(z; \phi) \log \frac{p(x, z; \theta)}{q(z; \phi)} dz$$

- We use chain rule

$$ELBO(\phi, \theta) = \int q(z; \phi) \log \frac{p(x; \theta) p(z|x; \theta)}{q(z; \phi)} dz$$

- Separate part independent on z:

$$= \int q(z; \phi) \log p(x; \theta)\, dz + \int q(z; \phi) \log \frac{p(z|x; \theta)}{q(z; \phi)} dz =$$

$$= \log p(x; \theta) + \int q(z; \phi) \log \frac{p(z|x; \theta)}{q(z; \phi)} dz =$$

- Definition of KL divergence:

$$= \log p(x; \theta) - KL(q(z; \phi) || p(z|x; \theta)).$$

- Bound is tight iff KL = 0.

M. Hoffman, M. Johnson, ELBO surgery, NIPS2016

Variational Autoencoder

Dependence on prior on z

- We use chain rule

$$ELBO(\phi, \theta) = \int q(z;\phi) \log \frac{p(z)p(x|z;\theta)}{q(z;\phi)} dz \ =$$

- Separate KL divergence part:

$$= \int q(z;\phi) \log p(x|z;\theta)\, dz + \int q(z;\phi) \log \frac{p(z)}{q(z;\phi)} dz =$$

- Definition of KL

$$= \int q(z;\phi) \log p(x|z;\theta)\, dz - \ KL(q(z;\phi)||p(z)).$$

Reconstruction loss **Latent Parametrization agreement**

Latent Parametrization Agreement

- Measures the degree to which the auxiliary distribution, $q(h, \theta)$, matches the prior $p(z)$:

$$KL(q(z;\phi)||p(z)).$$

- The KL divergence has closed from if both are distributions are known parametric (e.g., Gaussian):

$$q(z;\phi) = N_z(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$
$$p(z) = N_z(\mathbf{0}, \boldsymbol{I}).$$

- In this case:

$$KL[q(z|x,\theta), \Pr(z)] = 12(Tr[\Sigma] + \mu^T\mu - D - \log[\det[\Sigma]])$$

D. Kingma et al. Auto-Encoding Variational Bayes, ICLR 2014

Latent Parametrization Agreement: training

- Measures the degree to which the auxiliary distribution, $q(h, \theta)$, matches the prior $p(z)$:

$$KL(q(z; \phi)||p(z)).$$

- The KL divergence has closed from if both are distributions are known parametric (e.g., Gaussian):

$$\boldsymbol{q(z|x; \phi) = N_z\left(g_\mu(x; \phi), g_\sigma(x; \phi)\right)};$$
$$p(z) = N_z(\mathbf{0}, \boldsymbol{I}).$$

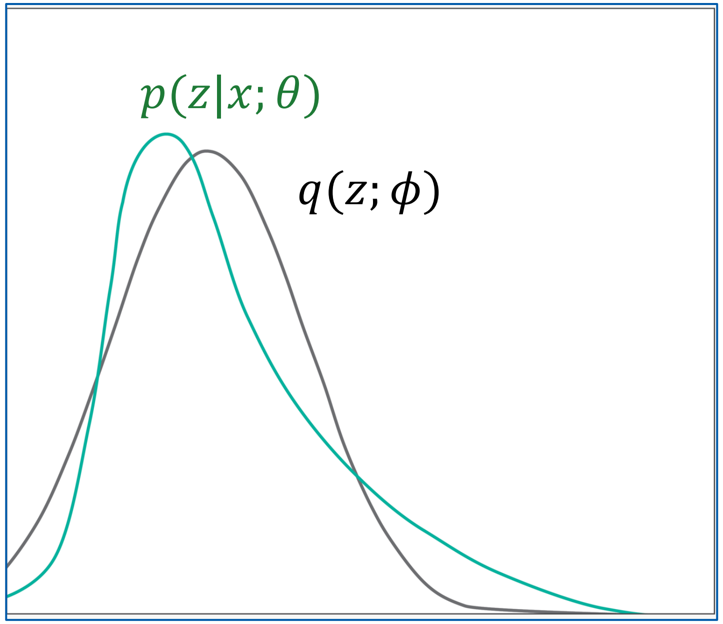
- In this case:

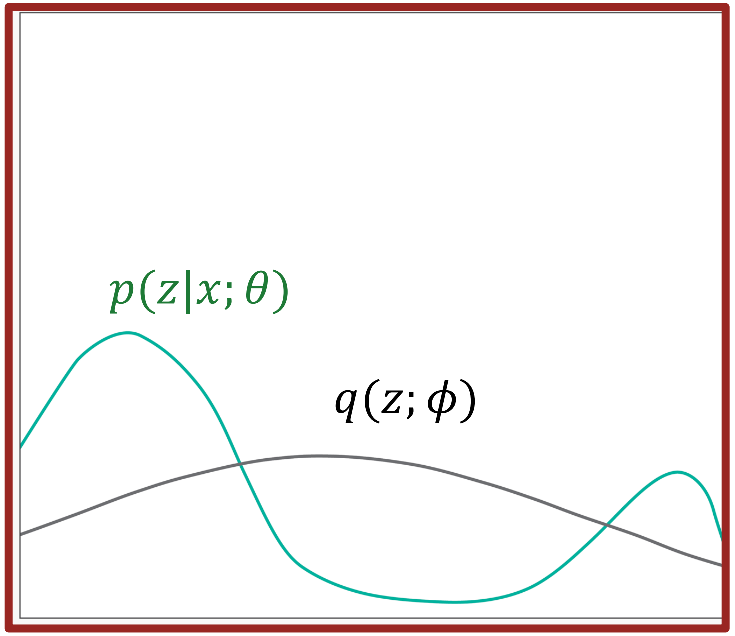
$$KL[q(z|x, \theta), \Pr(z)] = 12(Tr[\Sigma] + \mu^T\mu - D - \log[\det[\Sigma]])$$

Problems of parametrization choice

- For tightness of bound:

$$KL(q(z;\phi)\big|\big|p(z|x;\theta)\big) \to 0.$$





- The tight bound might be unreachable.

Figure: BorelisAI

Reconstruction Loss

- Reconstruction loss:

$$\int q(z|x;\phi)\log p(x|z;\theta)\,dz\,.$$

- Still troubles to compute directly, we can approximate it with mathematical expectation (even with one sample z^*) and thus have:

$$ELBO(\theta,\phi) = \int q(z;\phi)\log p(x|z;\theta)\,dz - KL\big(q(z;\phi)\big|\big|p(z)\big) \approx$$

$$\approx \log[p(x|z^*;\phi)] - KL[q(z|x,\theta),p(z)].$$

Variational Autoencoder

Maximize

$$\mathcal{L}_{VAE} = \mathbb{E}_z \log[p(x|z;\theta)] - KL[q(z|x,\phi), p(z)]$$

Autoencoder, as we reconstruct $x \to z \to x$.
Variational, as it constructs Gaussian approximation to the posterior distribution.

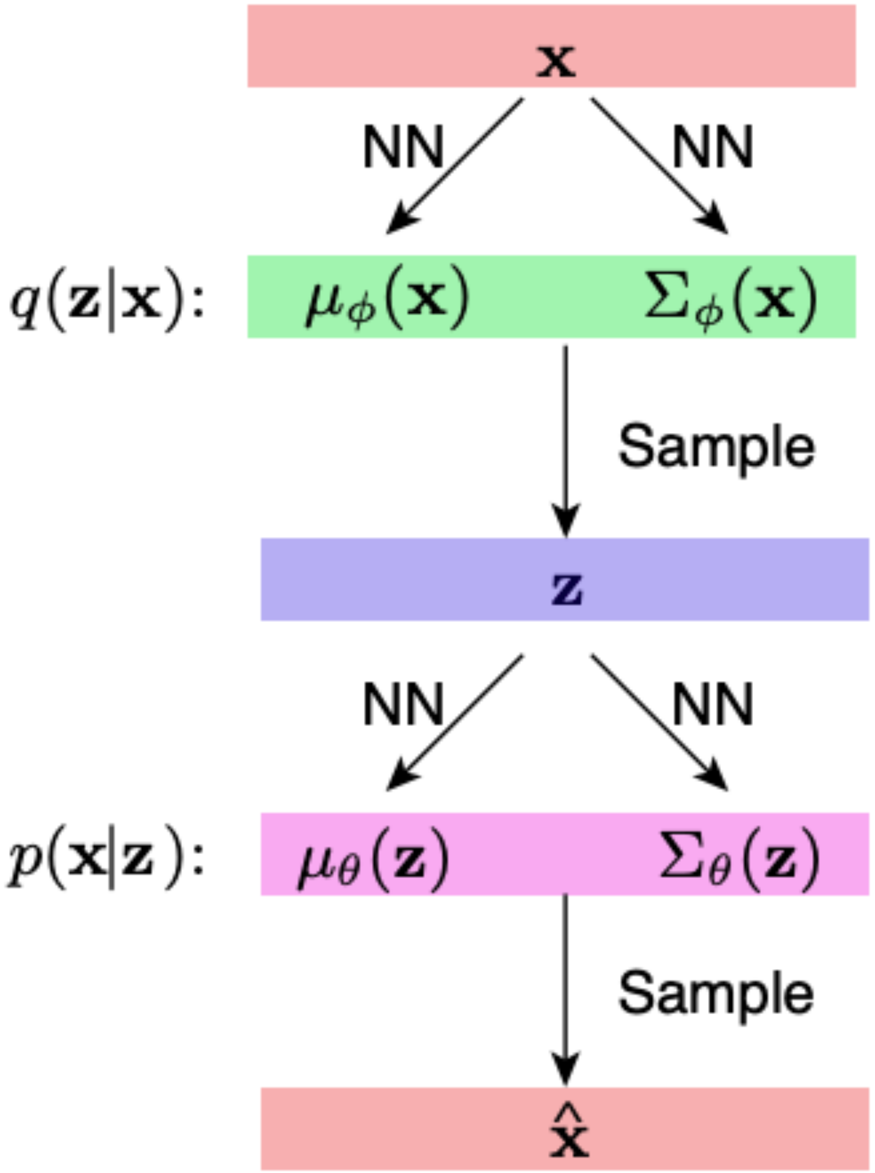


Fig: Kao UCLA DL lectures

Training VAE

We can estimate gradients for ELBO:

- gradient wrt θ is easy:

$$\nabla_\theta \mathcal{L}(x; \theta, \phi) = \nabla_\theta \log p(x, z; \theta)$$

  and can be done using a simple MC estimation.

- for ϕ situation is much more tricky, since we calculate expectation over q:

$$\nabla_\phi \mathcal{L}(x; \theta, \phi) = \nabla_\phi \mathbb{E}_{q(z|x;\phi)}[\log p(x, z; \theta) - \log q(z|x; \phi)].$$

  luckily, we can use reparameterization.

Reparameterization trick

- Sampling:

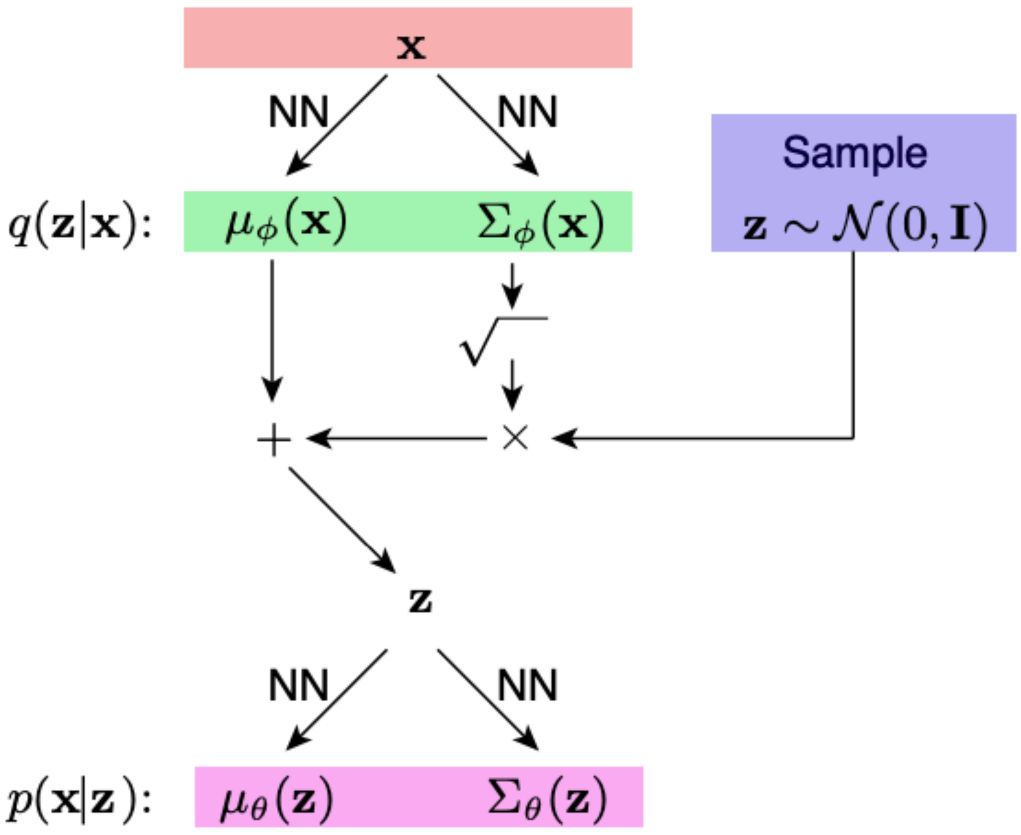
$$z \sim q(z|x)$$

- Can be done:

$$\varepsilon \sim N(0, I);$$

$$z = \mu_\phi(x) + \Sigma^{\frac{1}{2}}(x)\varepsilon.$$

- Backpropagation now does not pass through sampling.



Final Per-event Objective and Results

- Write out

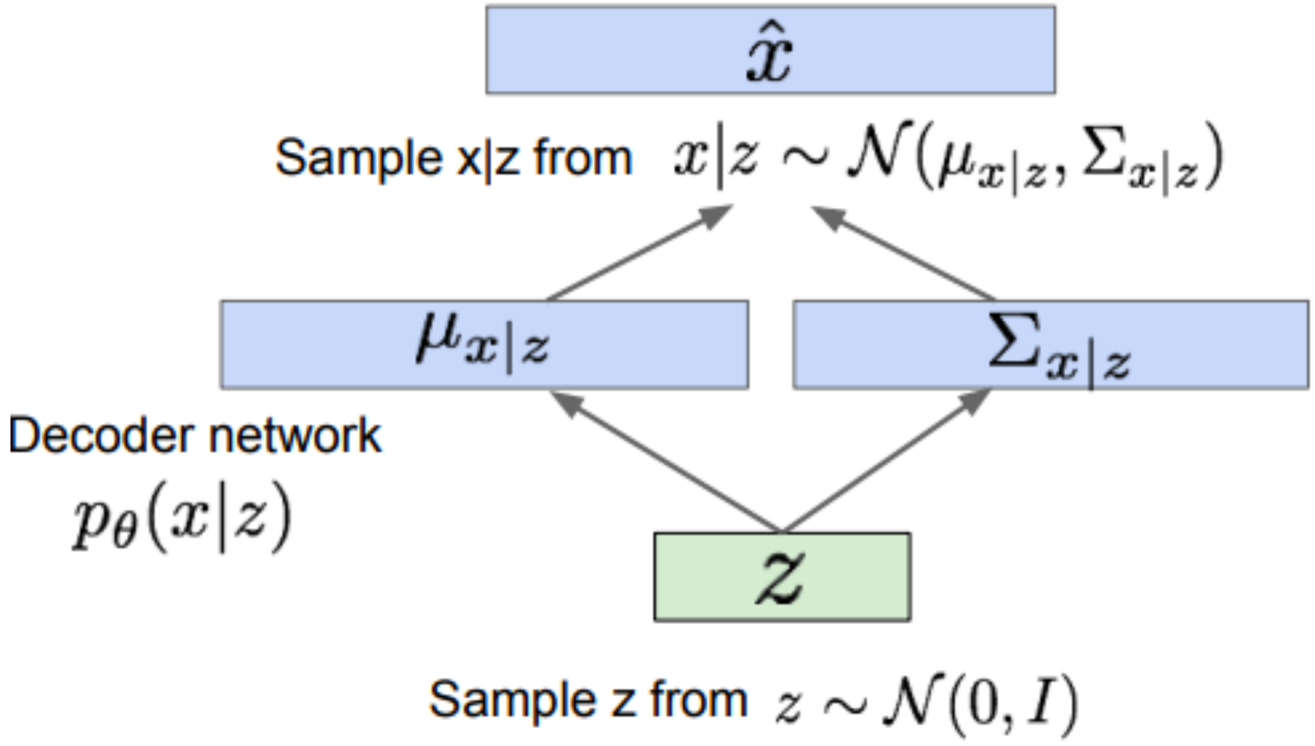
$$\mathcal{L}_{VAE}\,(x_i;\theta,\phi) = \mathbb{E}_{z\sim p(\varepsilon)}\log[p(x_i|z;\theta)] - KL[q(z|x_i,\phi),p(z)]$$

- We can even use a single noise sample:

$$\boxed{\mathcal{L}_{VAE}\,(x_i;\theta,\phi) = \log[p(x_i|z;\theta)] - KL[q(z|x_i,\phi),p(z)]}$$

- Note that the gradient estimates from per-event ELBO will be unbiased.

Sampling



- We now can use the same idea like in GMM.
- Sampling from prior.

Results

- The latent space is well organized.
- The results are good.

(b) Learned MNIST manifold

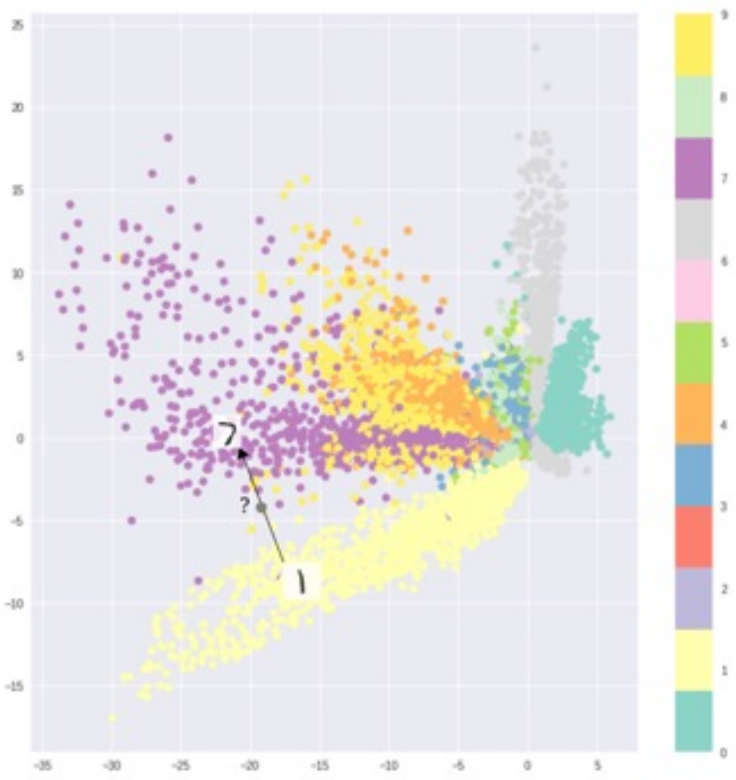
Results

(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

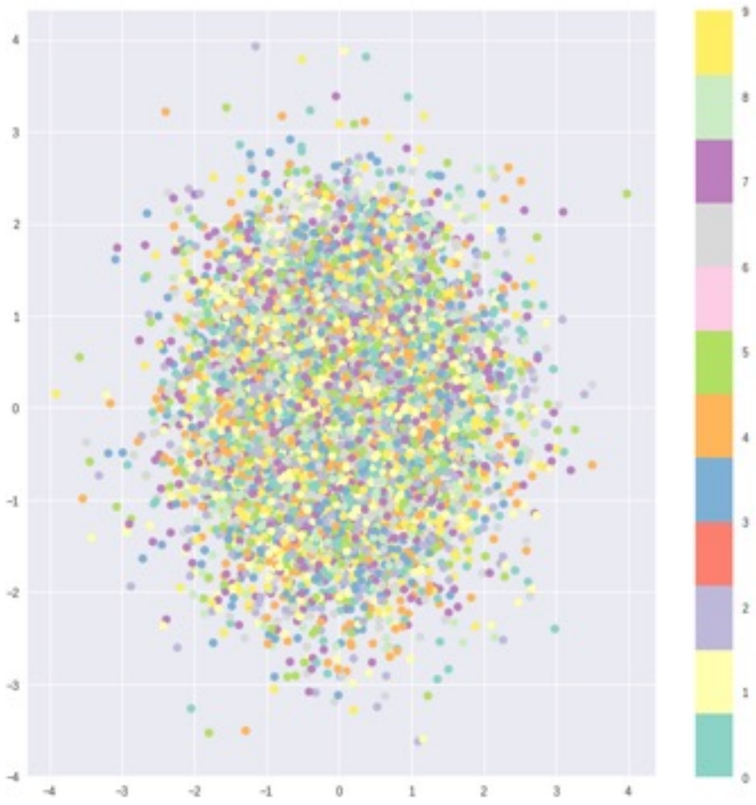
Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.

Results Discussed

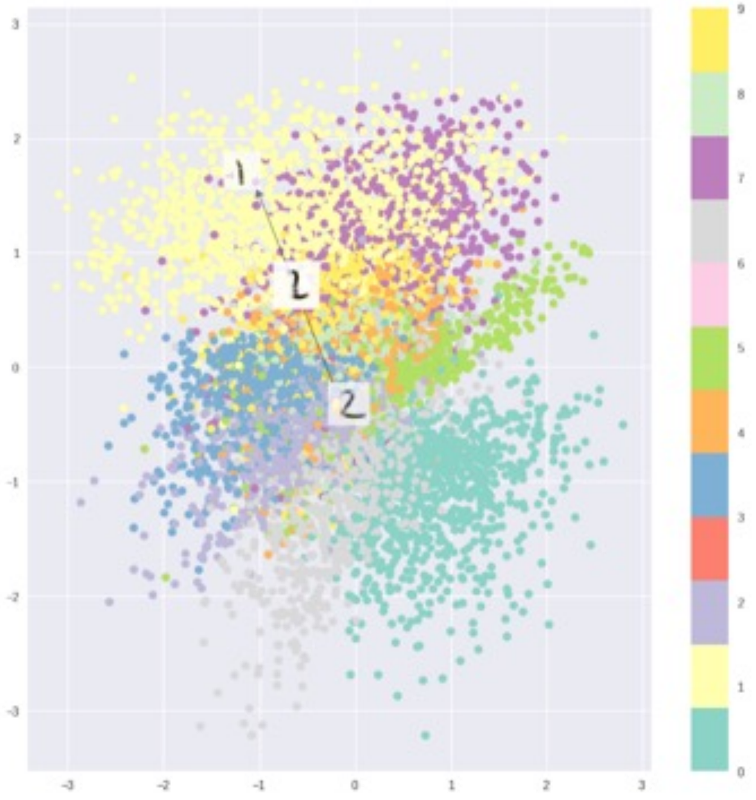
Only reconstruction loss



Only KL divergence



Combination



Shafkat's blog

Amortized Inference

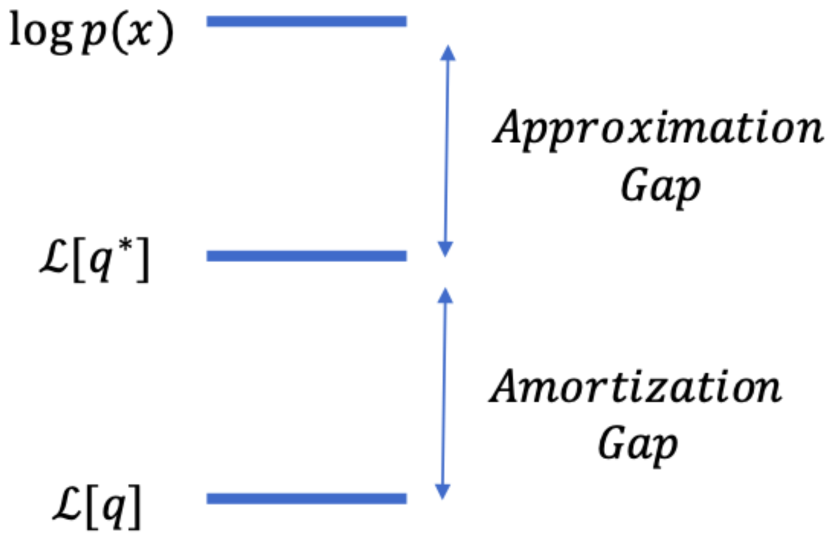


Figure 1. Gaps in Inference

- variational parameters ϕ_i should be calculated for every data point x_i;
- for big datasets this is unsustainable;
- forget about it and make a single model.

Cremer et al Inference Suboptimality in Variational Autoencoders, ICML 2018

Images for Amortization Gap

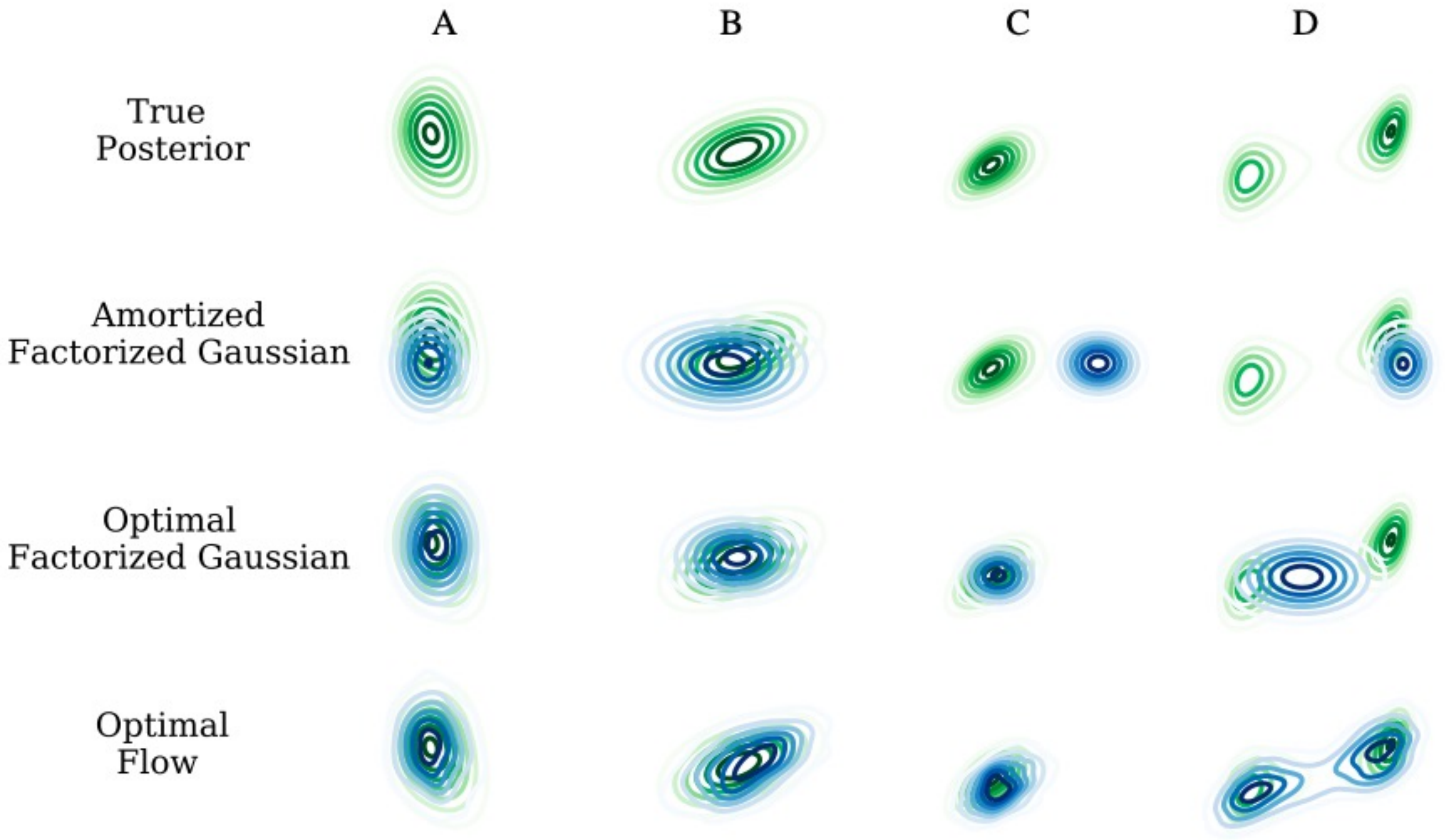


Figure 2. True Posterior and Approximate Distributions of a VAE with 2D latent space. The columns represent four different datapoints. The green distributions are the true posterior distributions, highlighting the mismatch with the blue approximations. Amortized: Variational parameters learned over the entire dataset. Optimal: Variational parameters optimized for each individual datapoint. Flow: Using a flexible approximate distribution.

VAE: starting optimization problem

- Stochastic optimization can get stuck:
  - in the beginning, likelihood term $\log p(x|z)$ is relatively weak, such that an initially attractive state is where $q(z|x) \approx p(z)$.
- Several ideas possible:
  - optimization schedule - gradually increase weight of KL.
  - free bits - ensure that on average, a certain minimum number of bits of information are encoded per latent variable.

D. Kingma et al., An Introduction to Variational Autoencoders

VAE: posterior collapse

Can happen to weak encoders or noisy data.

The sampling does not depend on the latent model.

An averaged figures are produced.

KL-divergence term annealing is proposed.

Fig: J. Chou Generated Loss and Augmented Training of MNIST VAE

VAE: blurred problem

Due to minimization of KL
$KL(q(z;\phi)||p(z|x;\theta)) \to 0.$

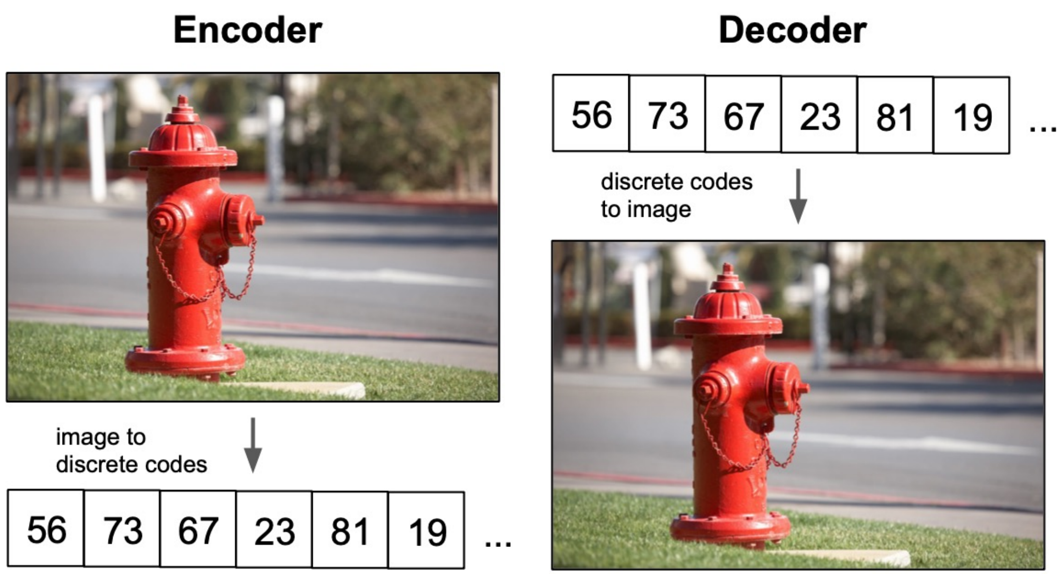
Can be countered by choosing a sufficiently flexible inference model, and/or a sufficiently flexible generative model.

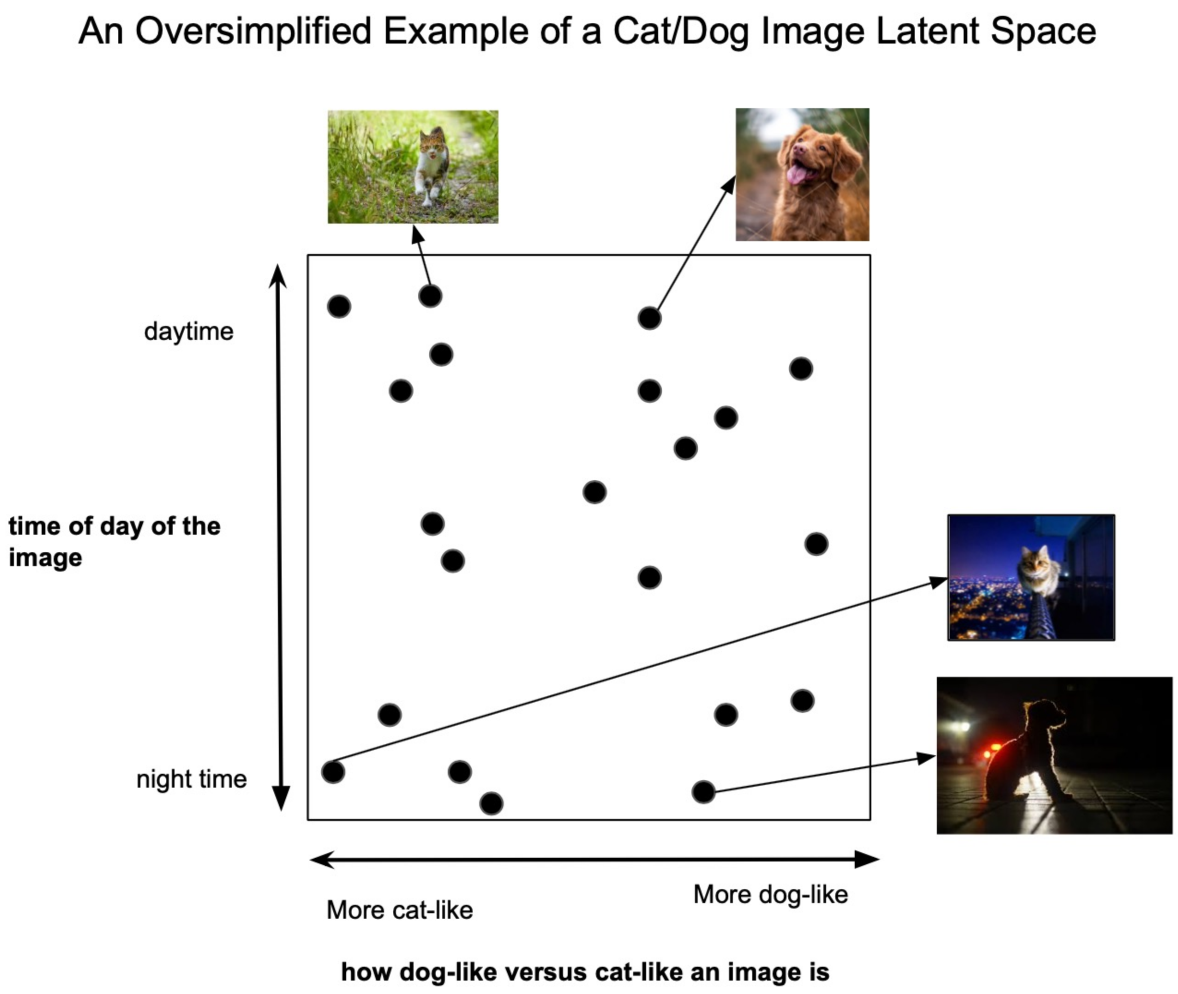
VQ-VAE

Latent Spaces

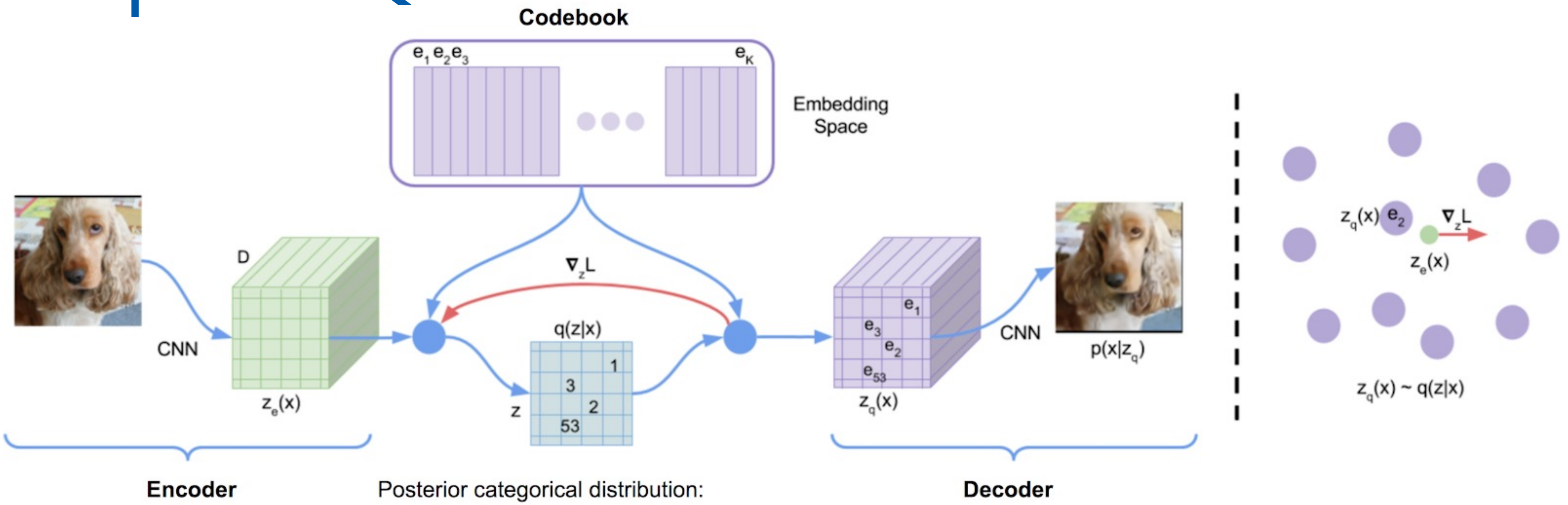
VAE gives a good representations

Can we study it?





Latent Space Quantization



- Reconstruction loss:

$$z_q(x) = \mathrm{argmin}_i \left|\left|z_e(x) - e_i\right|\right|$$

$z_e(x)$ - encoder vector for some raw input x

e_i - i-th codebook vector

$z_q(x)$ - resulting quantized vector that is passed as input to the decoder

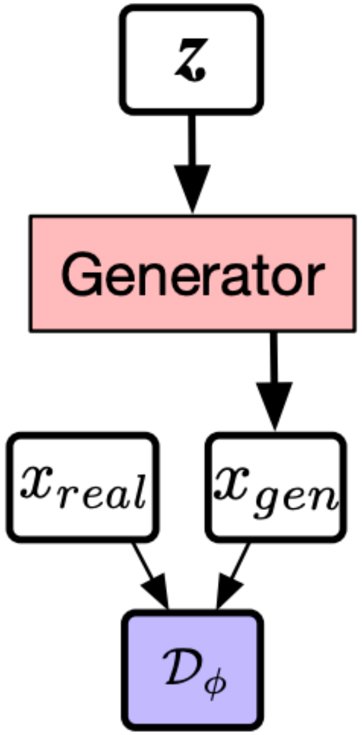
Aaron van den Oord et al., Neural Discrete Representation Learning

VQ-VAE results

Figure 2: Left: ImageNet 128x128x3 images, right: reconstructions from a VQ-VAE with a 32x32x1 latent space, with K=512.

VAE+GAN

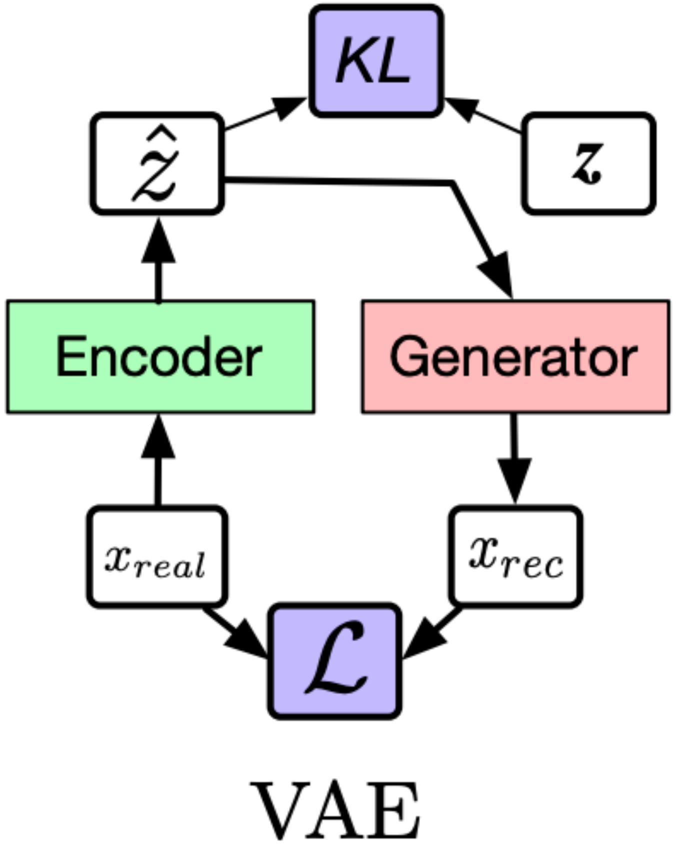
Improving GANs



DCGAN

- Main problems with GAN:
  - mode collapse;
  - intractable likelihood.
- Idea:
  - use likelihood-based mode;.
  - have easier inference;
  - diversify sampling.

Reminder: VAE structure



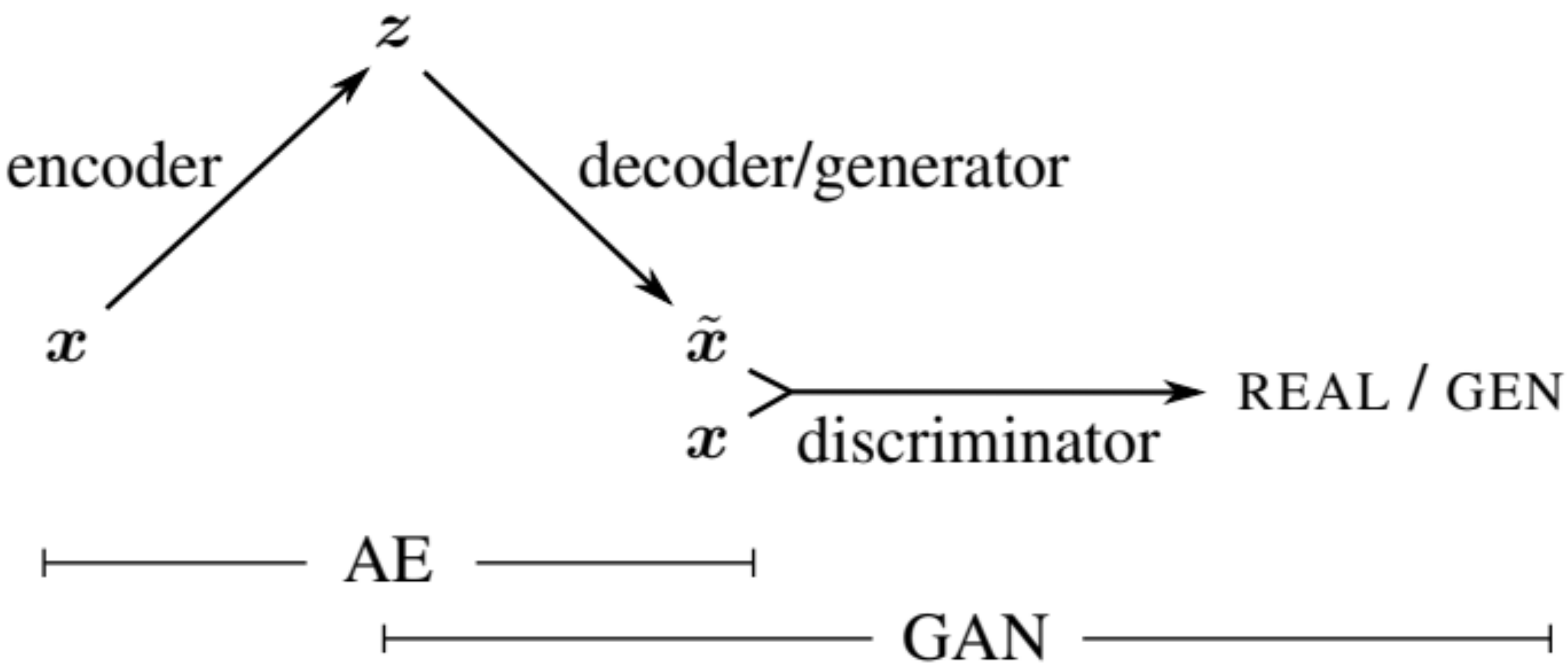
$$\mathcal{L}_{prior} = KL(q_n(z|x)||p(z))$$

$$+$$

$$-\mathcal{L}_{pixel} = \mathbb{E}_{q_n(z|x)}(\log p_\theta(x|z)$$

$$= \mathcal{L}_{VAE}$$

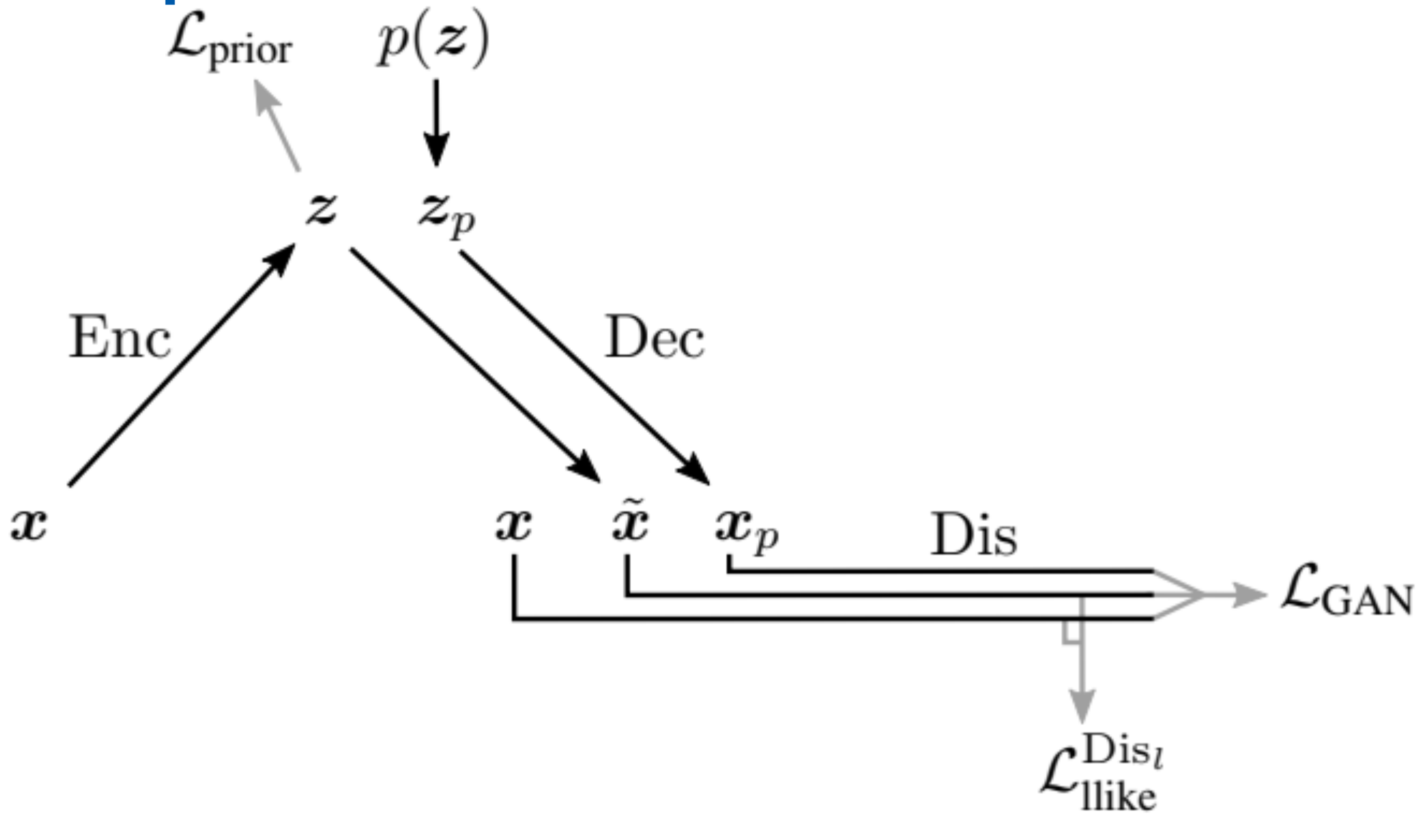
VAE+GAN



- problem of VAE is blurry output
- GAN overcomes due to the use of discriminator
- add GAN loss to VAE

https://arxiv.org/pdf/1512.09300.pdf

VAE+GAN expanded

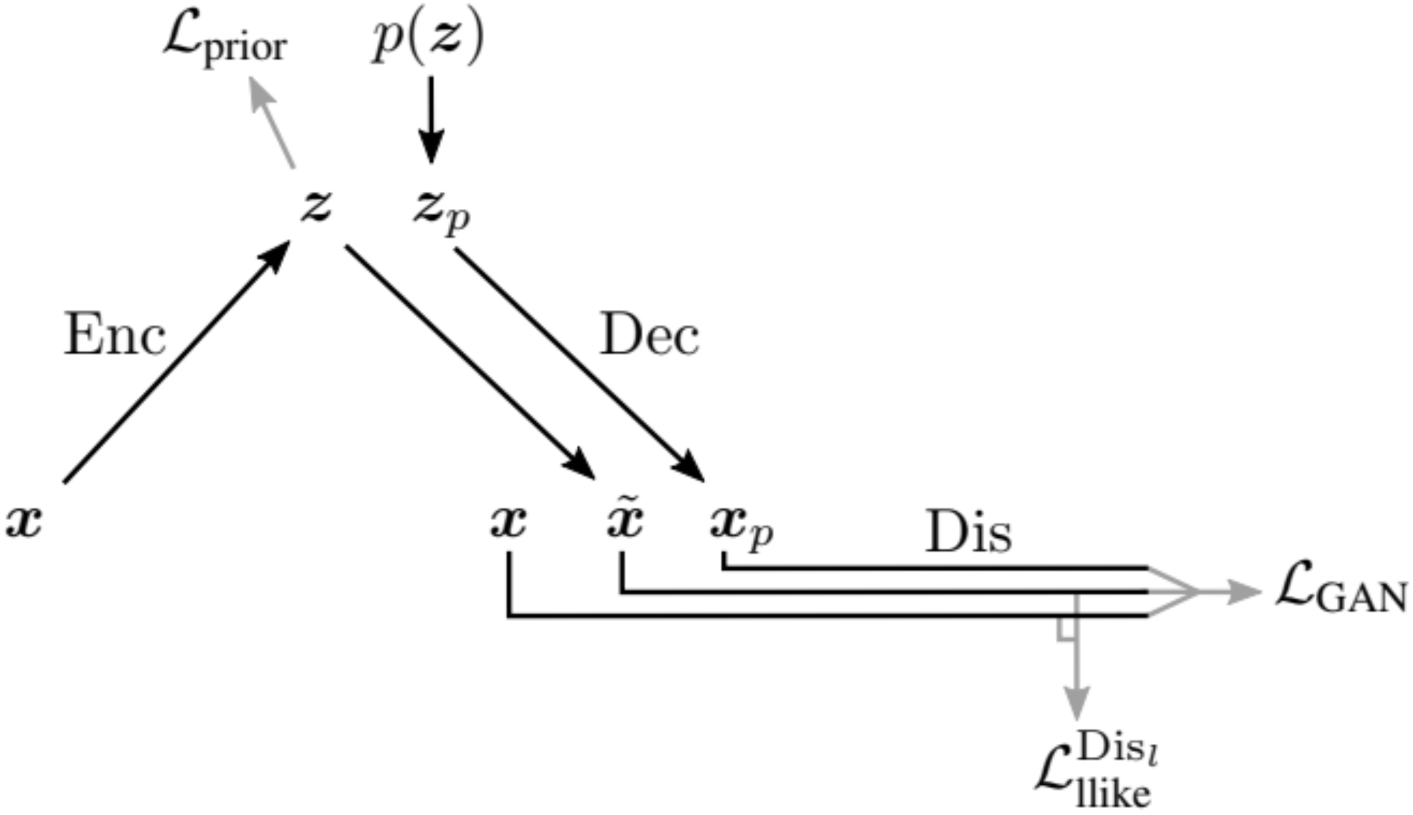


- Consider image as a whole:

$$-\mathcal{L}_{DIS} = \mathbb{E}_{q_n(z|x)} p(Dis_l(x)|z) = \mathbb{E}_{q_n(z|x)} N\left(Dis_l(x)|Dis_l(\tilde{x}), I\right), \qquad \tilde{x} \sim Dec(z)$$

where $Dis_l(x)$ denote the hidden representation of the l-th layer of the discriminator.

VAE+GAN final loss



$$\mathcal{L}_{GAN+VAE} = \underbrace{\mathcal{L}_{prior} + \mathcal{L}_{DIS}}_{\text{Image}} + \underbrace{\mathcal{L}_{GAN}}_{\text{Style}}$$

Image **Style**

VAE+GAN results

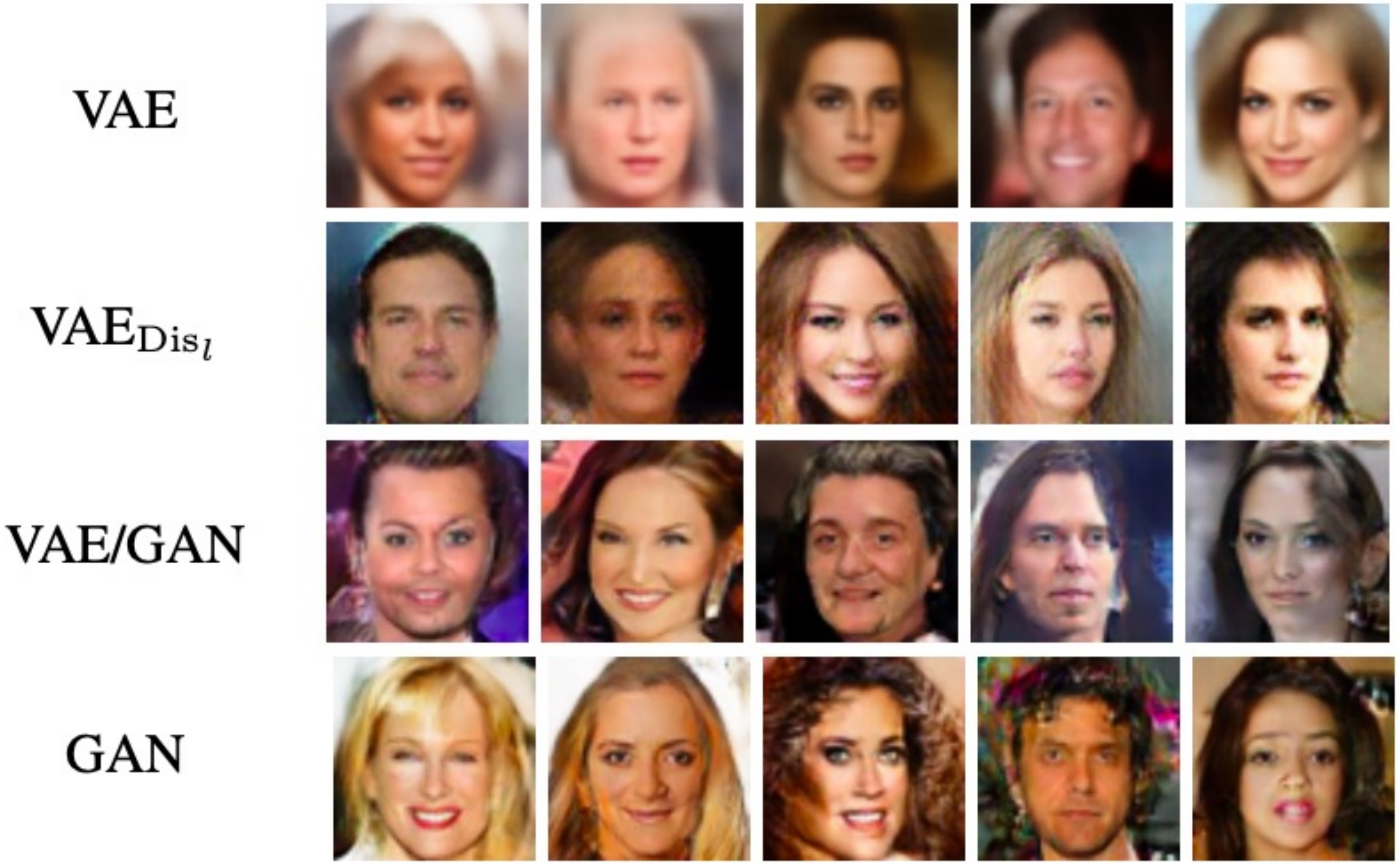


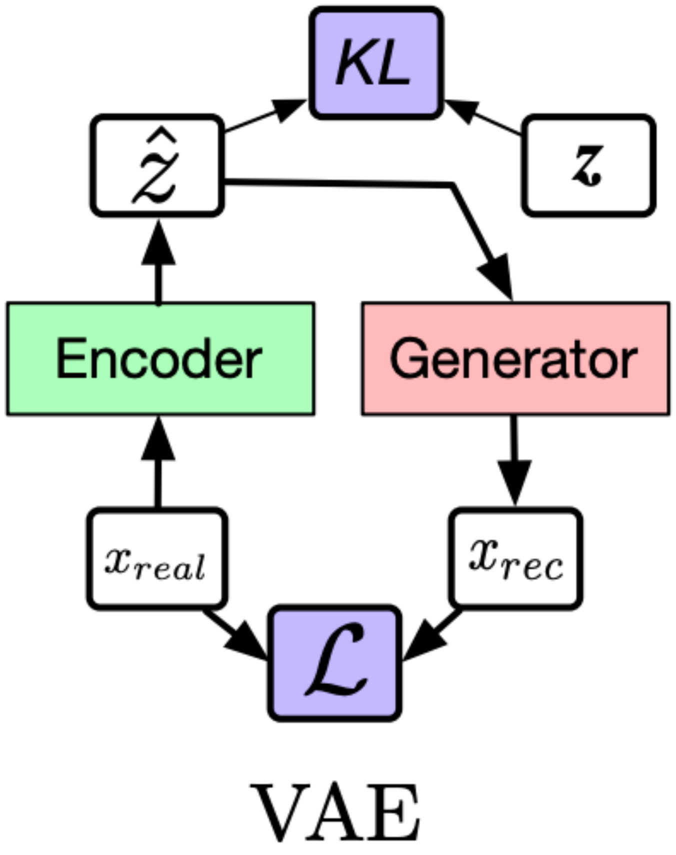
Figure 3. Samples from different generative models.

Wrap-up

- Benefits from combined loss and combined architecture.
- Better images than VAE or GAN standalone.
- Can be viewed as extension of VAE with adversarial selection.
- Can be viewed as extension of GAN with better latent space construction.

More VAEGANs

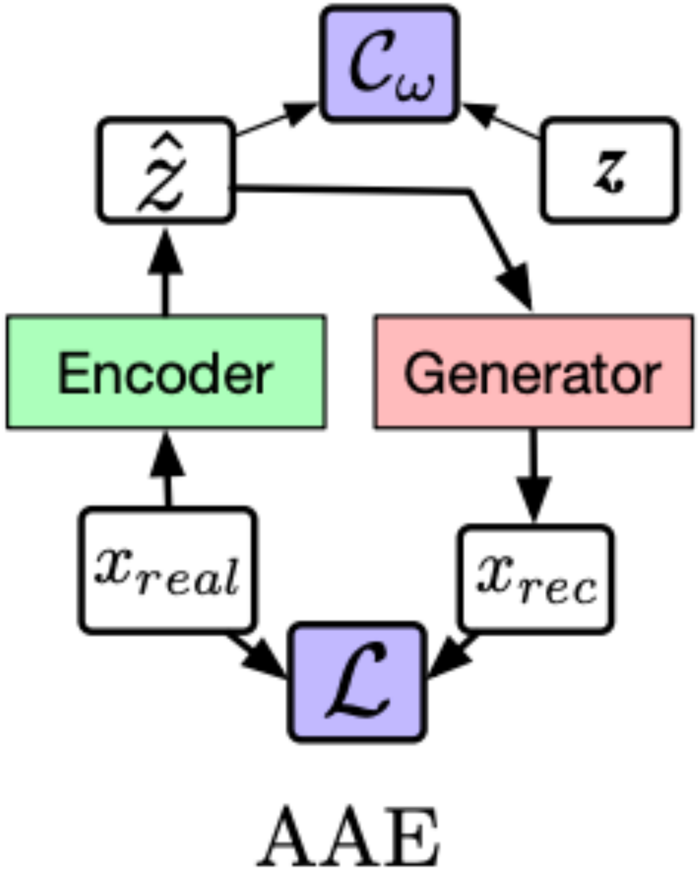
VAE structure



- KL divergence is considered as one of blurriness causes.
- We can switch this objective to GAN-like.

https://arxiv.org/pdf/1511.05644.pdf

AAE structure



- KL divergence is considered as one of blurriness causes.
- We can switch this objective to GAN-like.
- Price to pay: ELBO becomes less constrained.

https://arxiv.org/pdf/1511.05644.pdf

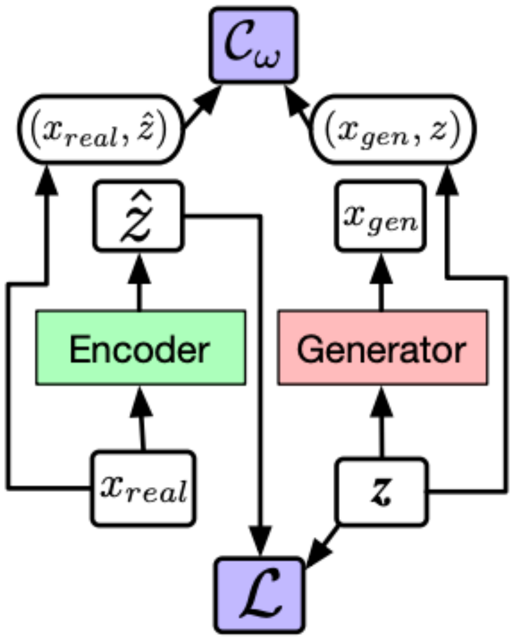
AAE results

(b) VAE samples

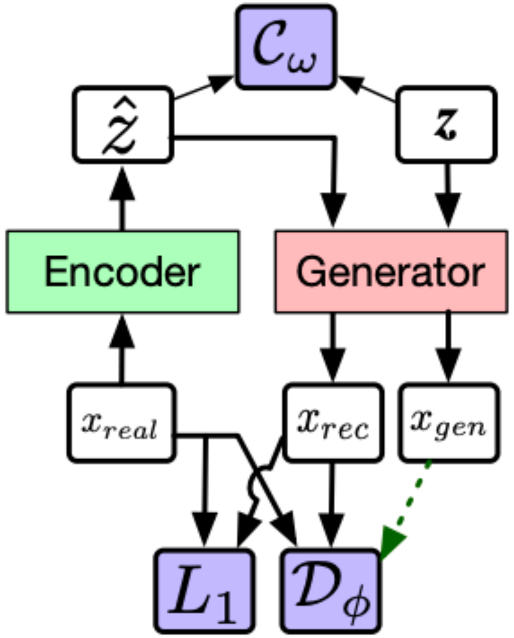
(c) AAE samples

https://arxiv.org/pdf/1802.06847.pdf

VEEGAN and VGH



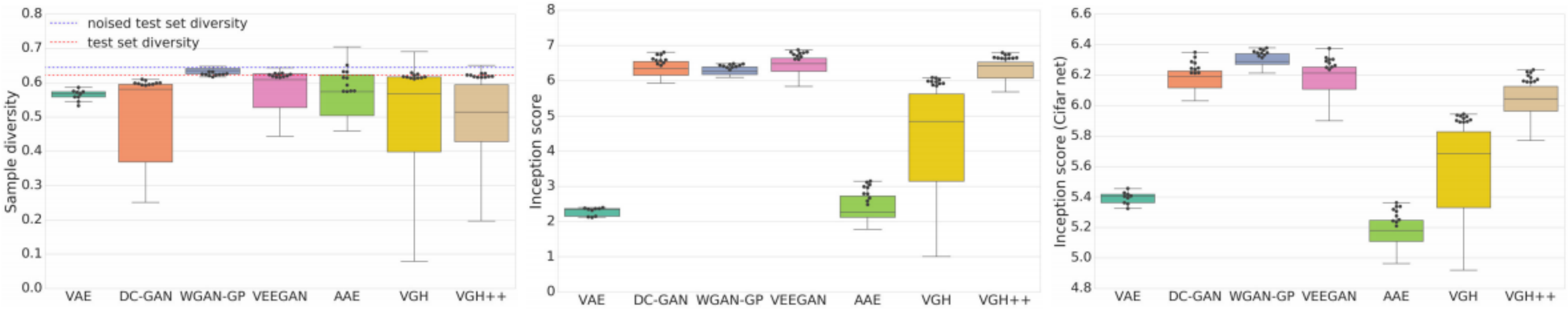
VEEGAN



VGH/VGH++

- VGH: Marginal matching and implicit distributions using GANs both in latent and visible space;
- VEEGAN: Directly match in joint space.

VAE+GAN merged



(a) Diversity score (CelebA) (b) Inception score (ImageNet) (c) Inception score (CIFAR)

Figure 8: (Left) Sample diversity on CelebA, and is viewed relative to test set: too much diversity shows failure to capture the data distribution, too little is indicative of mode collapse. We also report the diversity obtained on a noised-version of the test set, which has a higher diversity than the test set. (Middle) Inception scores on CIFAR-10. (Right) Inception scores computed using a VGG-style network on CIFAR-10. For inception scores, higher values are better. For test data, diversity score: 0.621, inception score: 11.25, inception score (using CIFAR-10 trained net): 9.18. Best results are shown with black dots, and box plots show the hyperparamter sensitivity.

- Our aim was to improve GAN mode collapse and VAE blurriness.
- The results are yet to be improved.

Conclusions

- VAE is one of the strong generative models that is used in current days.
- Several attempts were made to merge strong points of GANs and VAE.
- Work is ongoing.