Variational Auto-Encoders (Part 2)

What we'll learn about Variational Auto-Encoders:

Part 1 (previous): The "simple" explanation of a VAE, as a regularized AE

- What is a VAE
- How to train VAEs
- How do different terms of training loss influence what VAE learns
- How does a VAE relate to the basic AE

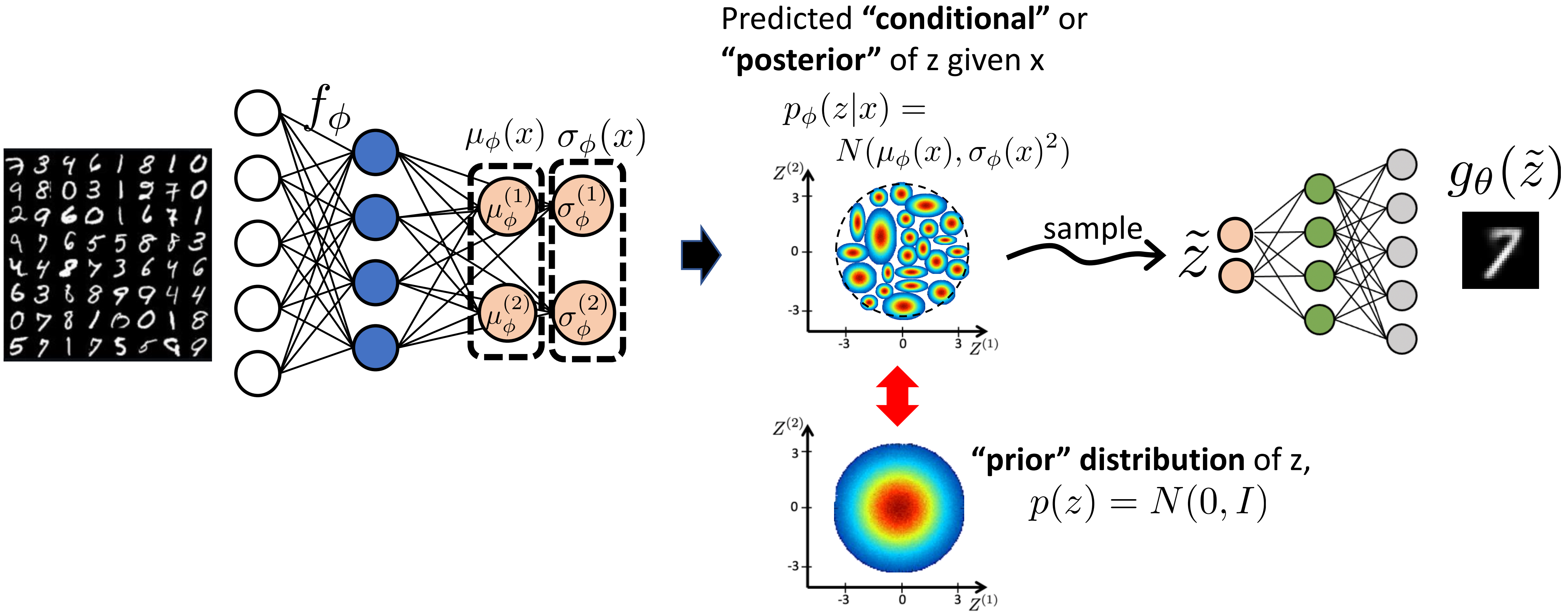
Part 2 (this video): Applications of VAE for…

- Generation of new data points
- Modifying data via interpolating between 2 inputs
- Modifying specific feature of an input
- Compression => Reconstruction

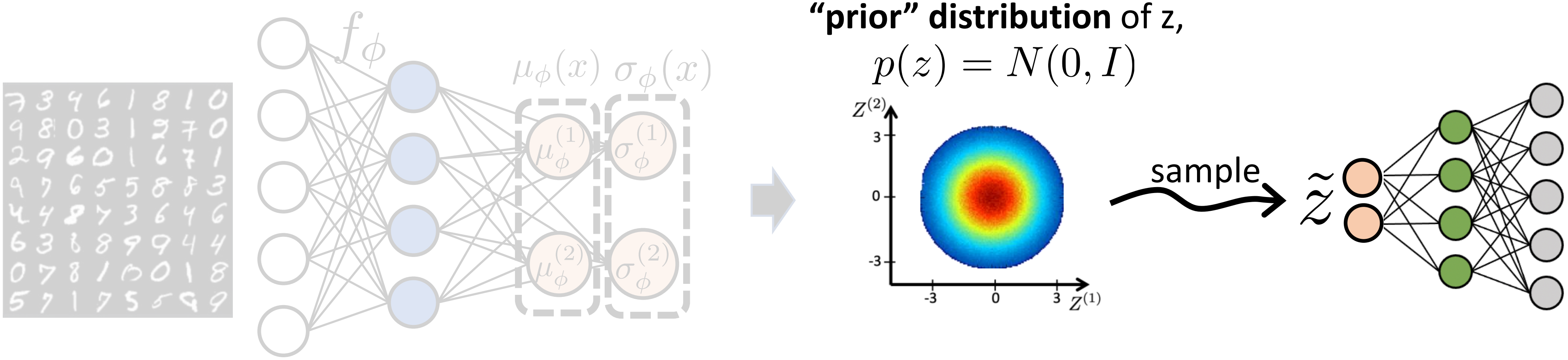
Part 3: The probabilistic derivation of a VAE
(Optional, non-assessed. To be published at a later date)

- Derivation of VAE as a probabilistic model for Density Estimation

VAE: Training

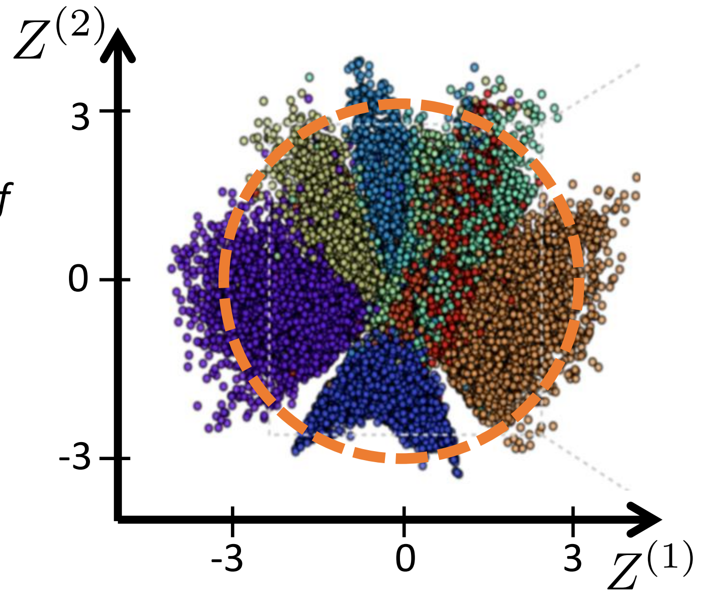


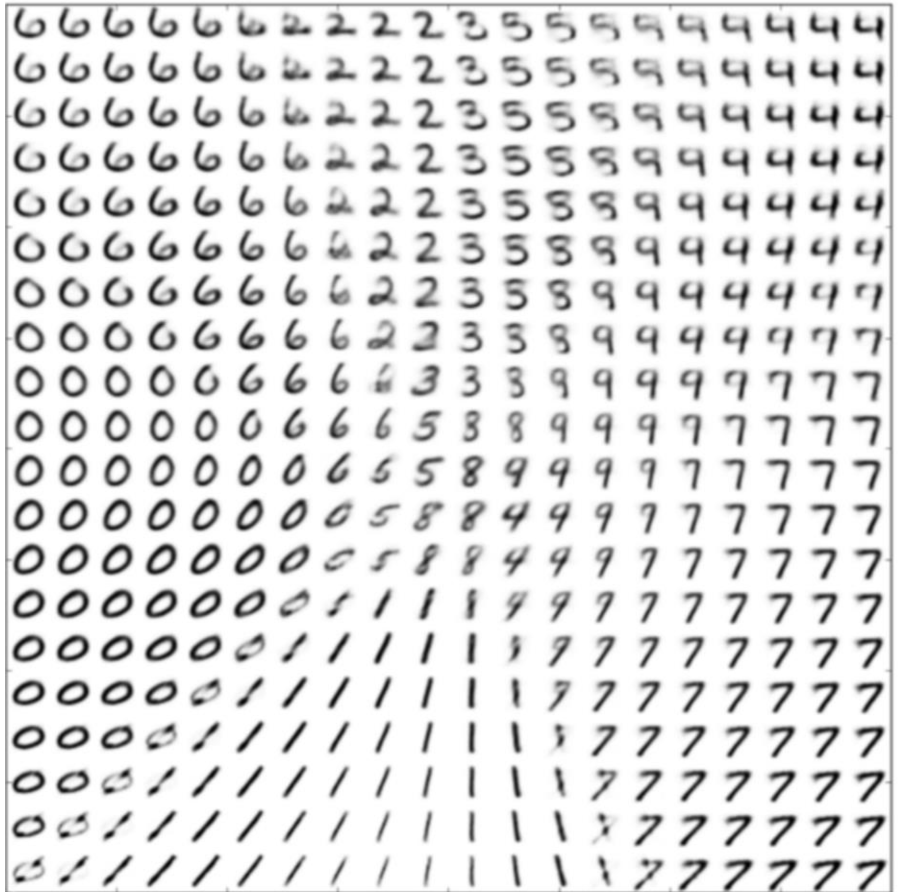
VAE: Generating new data by sampling from prior



Generating (synthesizing) new data with VAE

*Each dot is the <u>mean code</u> of a **training sample**,* $\mu_\phi(x)$



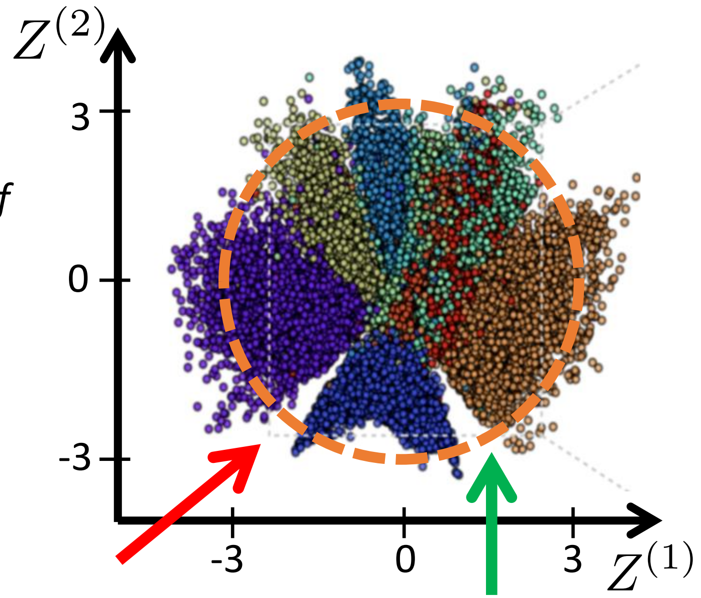


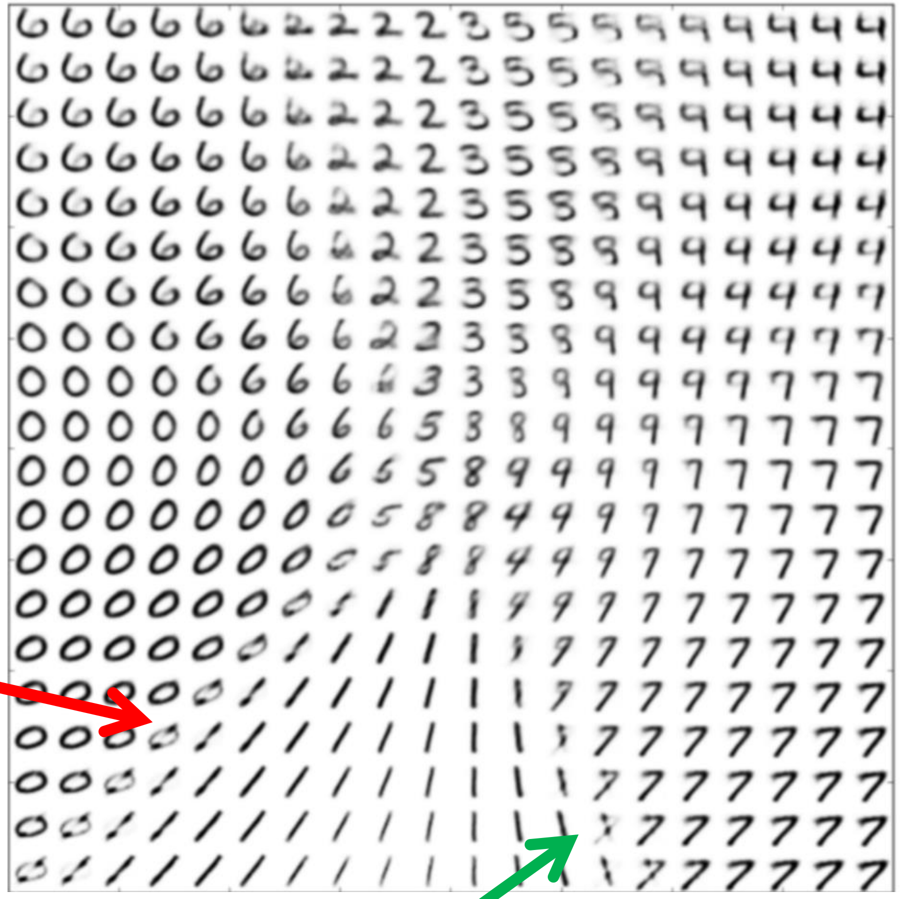
Values of z were sampled on a grid, under the area (-3,+3) covered by the prior.

Images from: Cloudera Fast Forward (link)

Generating (synthesizing) new data with VAE

*Each dot is the mean code of a **training sample**,* $\mu_\phi(x)$





Values of z were sampled on a grid, under the area (-3,+3) covered by the prior.

Even for VAE, some "gaps" can still be left in space of Z.
Why?

1. Reconstruction loss encourages separation of dissimilar data, "opposing" the regularizer.
2. SGD optimization does not find global optimum.

Images from: Cloudera Fast Forward (link)

Interpolating between different inputs with VAE

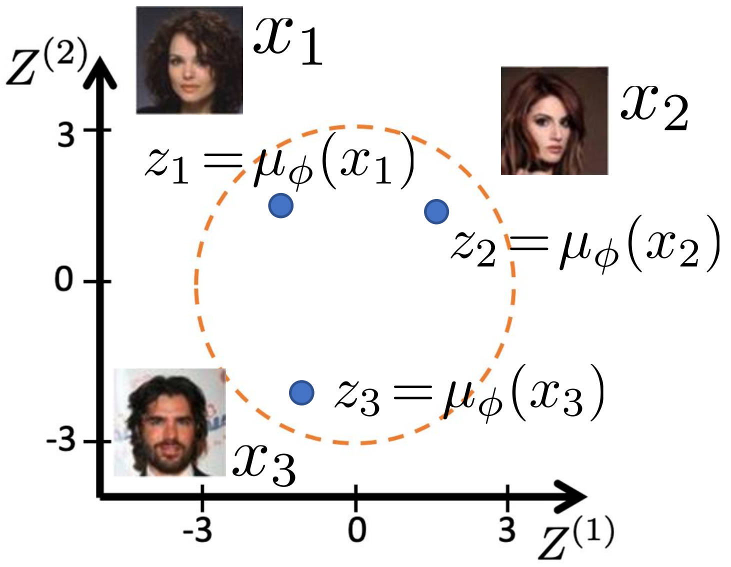


Image from: Fathy Rashad (link)

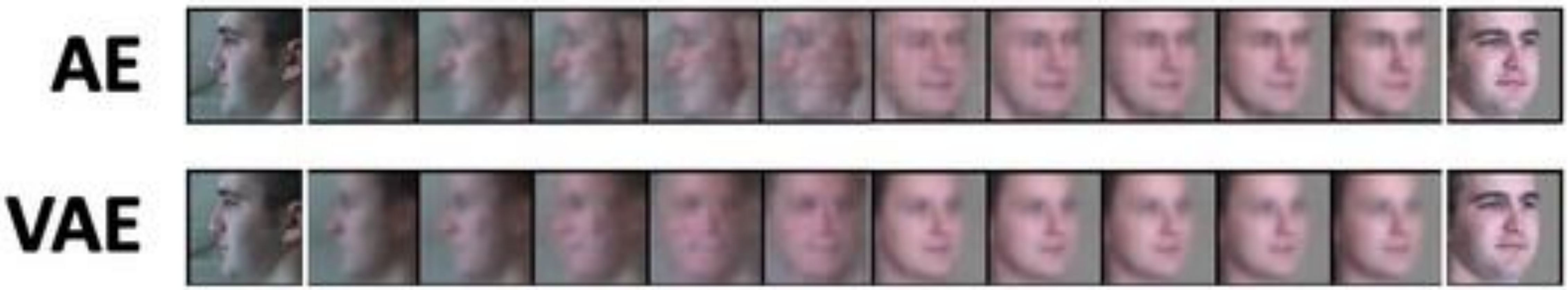
Algorithm:

1. **Encode** inputs x and get $\mu_\phi(x)$ as z.
   E.g. $z_1 = \mu_\phi(x_1), z_2 = \mu_\phi(x_2), z_3 = \mu_\phi(x_3)$
2. Create **new z** code by **interpolation**
   E.g. $z = z_1 + \alpha(z_2 - z_1) + \beta(z_3 - z_1)$
3. **Decode z** with decoder.

Can be for 2 or more inputs.

Interpolating between inputs: basic AE vs VAE

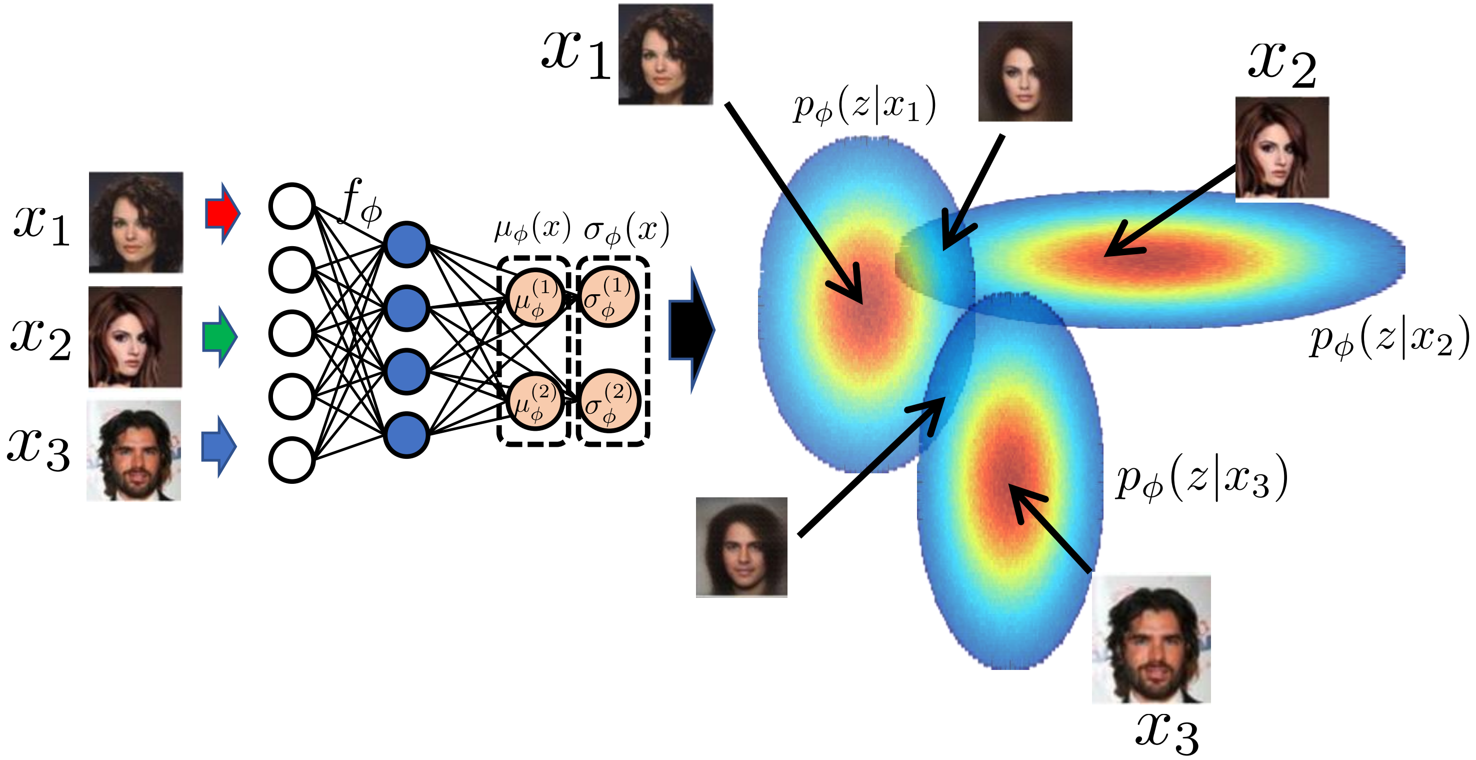
Problematic results, and a **sudden (non-smooth) change**



VAE tends to gives **smoother interpolation**

Image from: Yan et al, "Semantics-Guided Representation Learning with Applications to Visual Synthesis", 2020

Interpolation with a VAE gives “**smoothly**” changing output



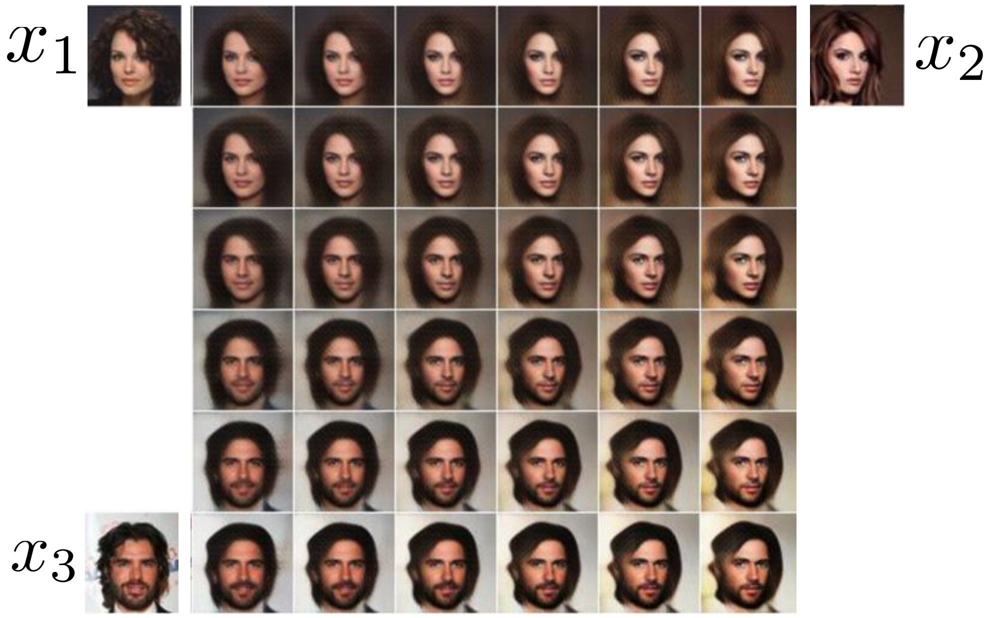
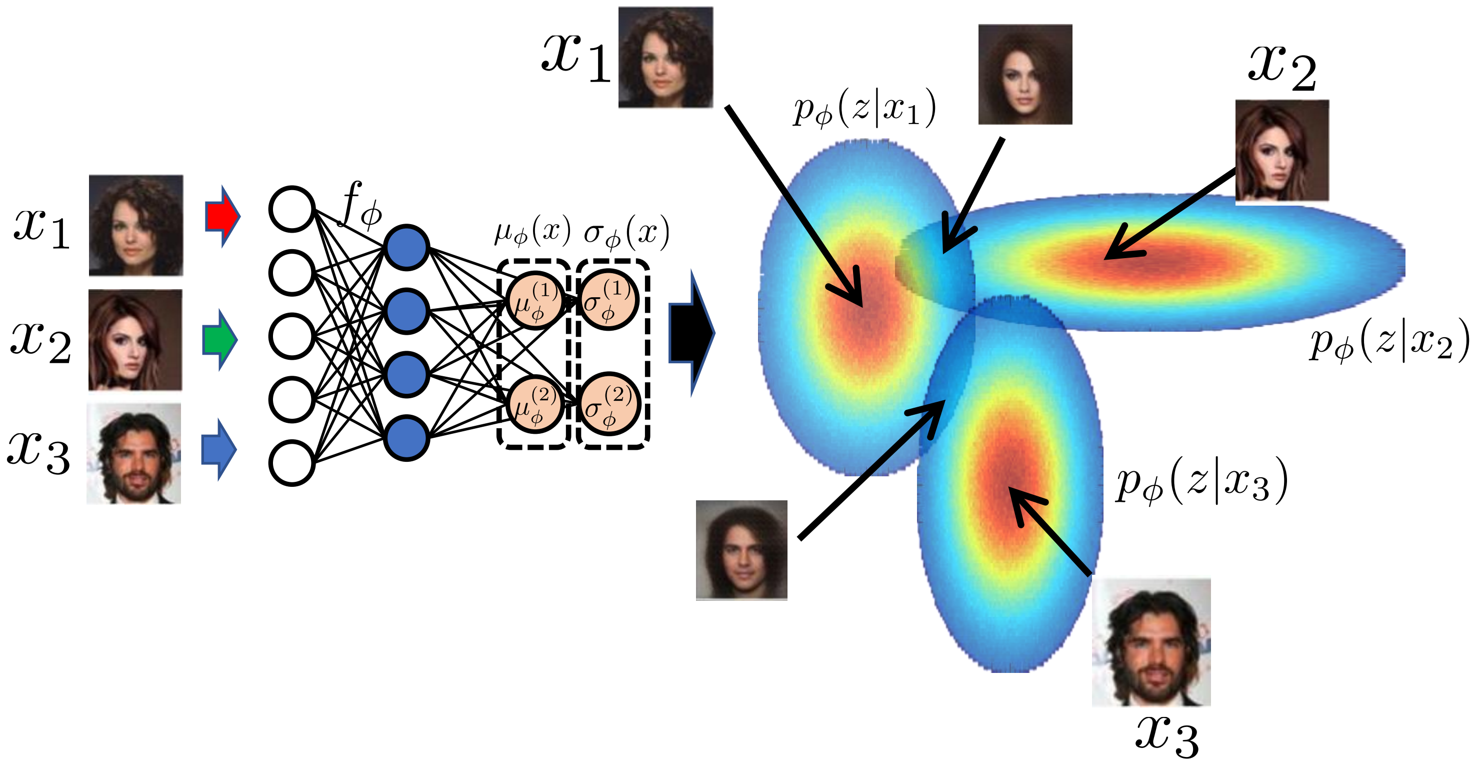


Image from: Fathy Rashad (link)

Interpolation with a VAE gives “**smoothly**” changing output



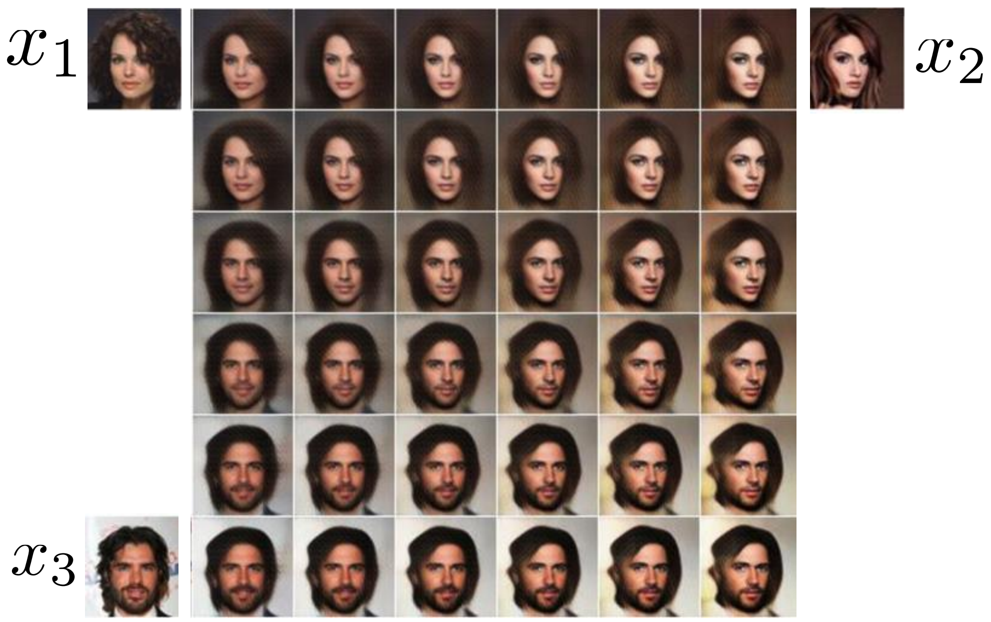
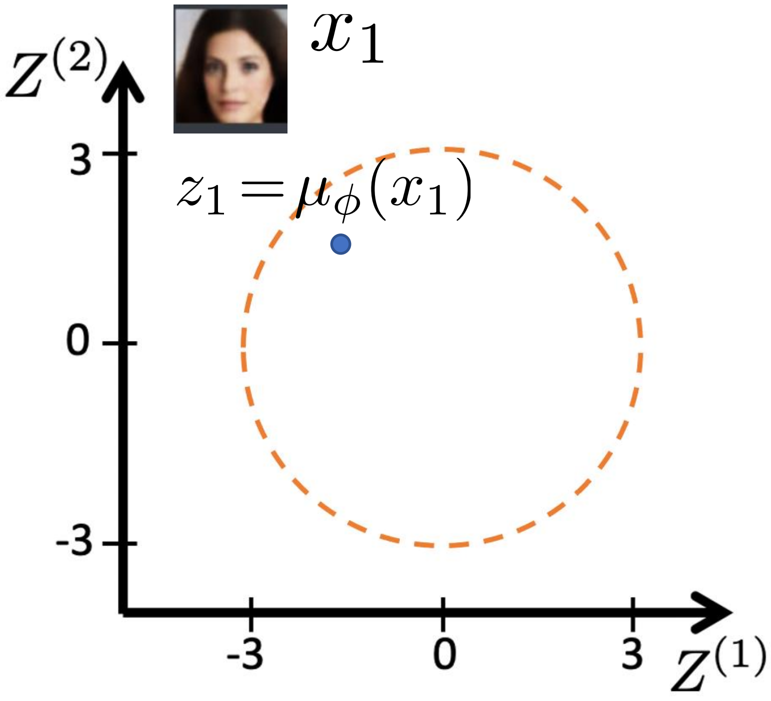


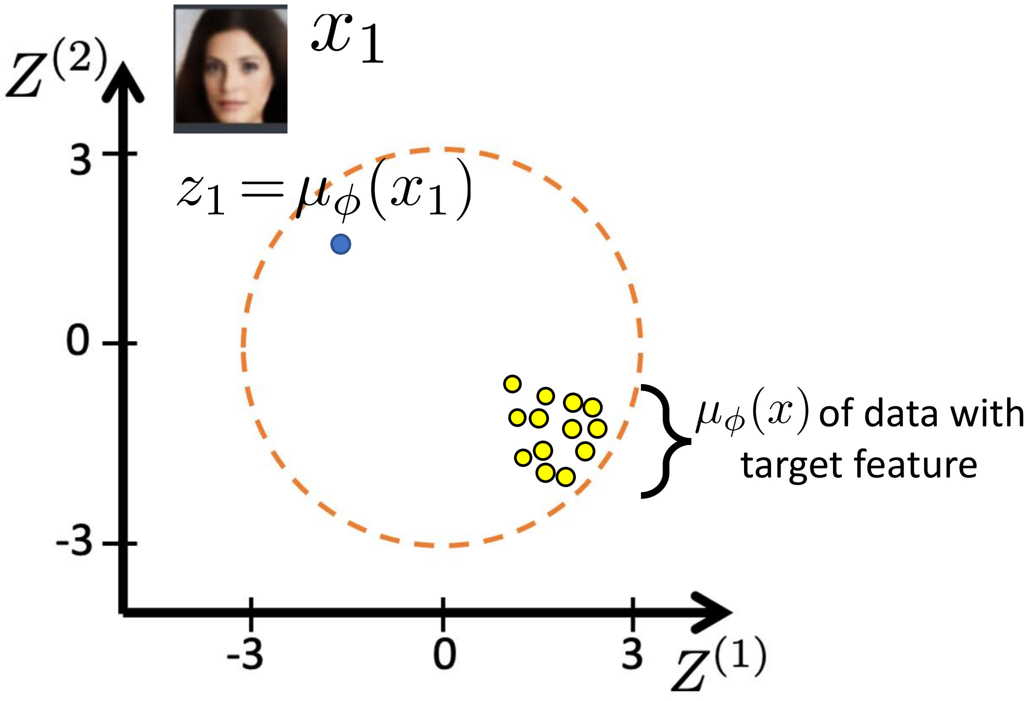
Image from: Fathy Rashad (link)

Smoothness: No *sudden* changes between 2 nearby z values. Thanks to training with Gaussian distribution $p_\phi(y|z)$ for encoding. In VAE, values of z between 2 training datapoints are associated with probability $p_\phi(z|x_1)$ code of one training image x_1, and probability $p(z|x_2)$ code of another training image x_2, where the 2 Gaussians predicted by encoder overlap. Therefore, decoding such intermediate values z leads to an image that has characteristics of both images that their p(z|x) overlap at that value of z. As we move in space of Z, Gaussians p(z|x) vary smoothly, and so characteristics in decoded images will vary smoothly.

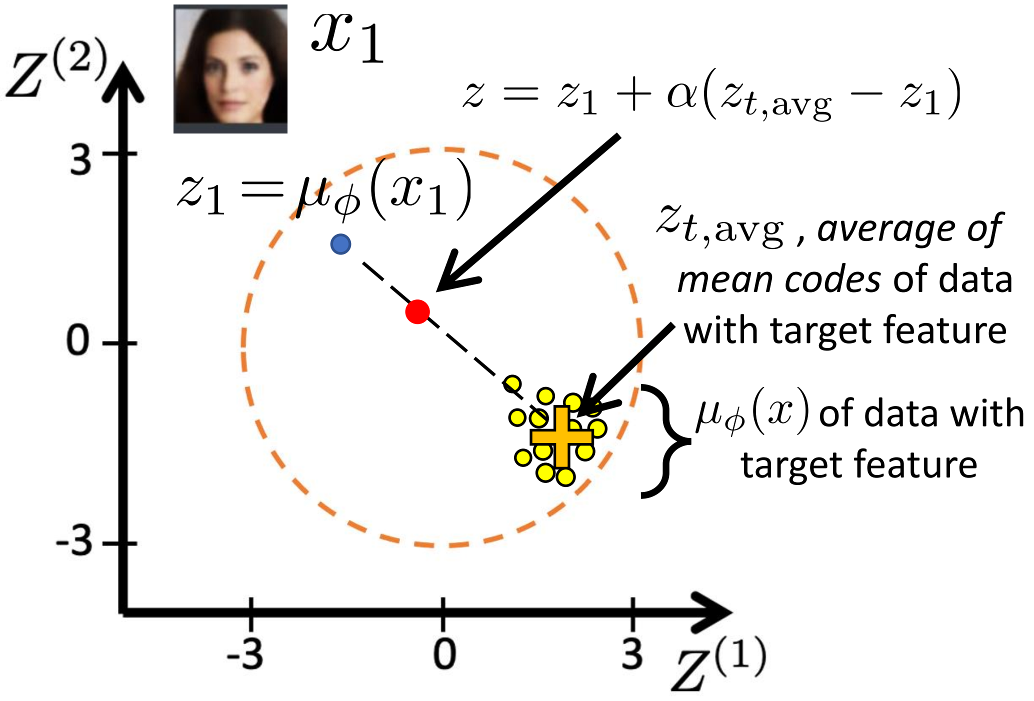
Altering specific features of data with VAE



Altering specific features of data with VAE



Altering specific features of data with VAE



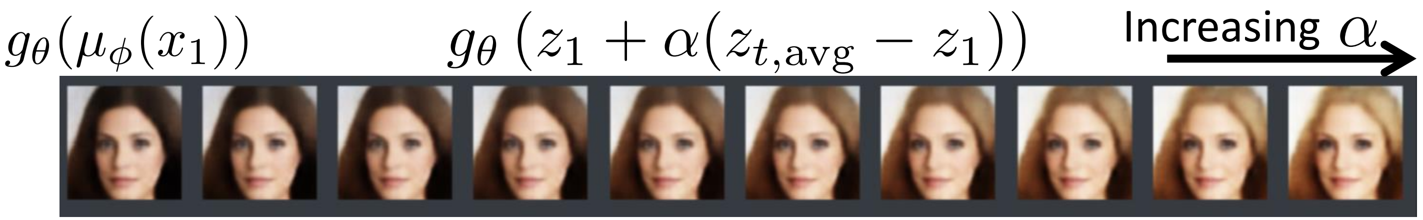
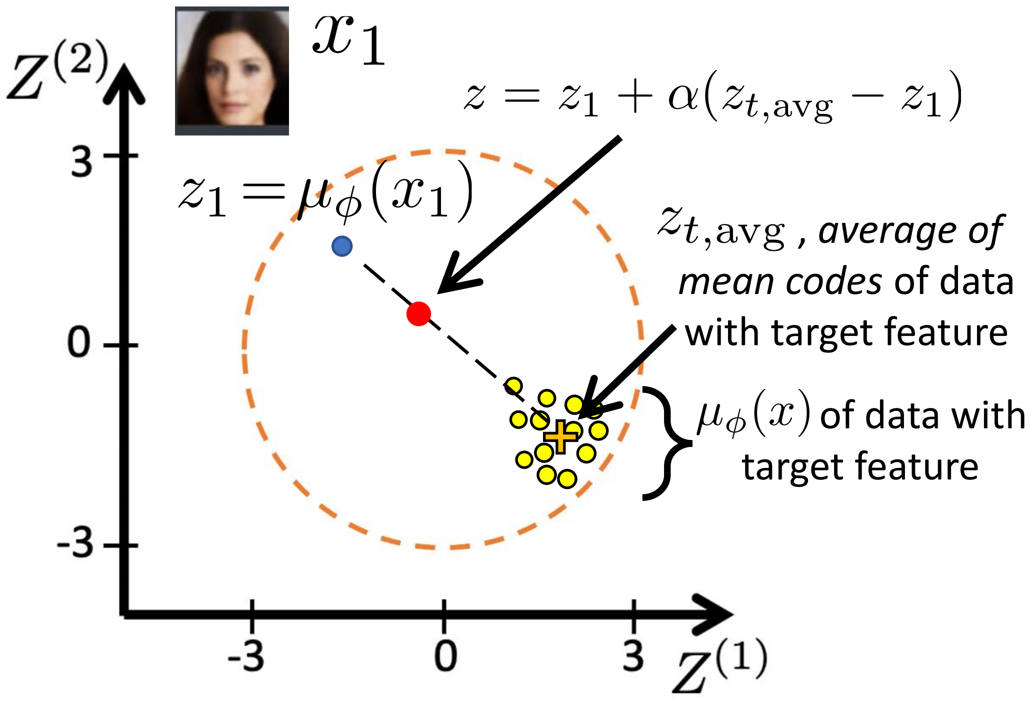


Image from: Steven Flores (link)

Altering specific features of data with VAE



Algorithm:

1. **Encode** original input x and use predicted $\mu_\phi(x)$ as its code:
   E.g. $z_1 = \mu_\phi(x_1)$
2. Identify all training samples that have the desired "target" characteristic. E.g. blondes. Assume these are $x_{t,1}, x_{t,2}, \ldots$
3. **Encode** all training samples with the target characteristic.
   Use mean of the Gaussian predicted by encoder as the code.
   E.g. $z_{t,1} = \mu_\phi(x_{t,1}),\ z_{t,2} = \mu_\phi(x_{t,2}), \ldots$
4. Compute **average** value of codes of all samples with target characteristic: $z_{t,\text{avg}} = average(z_{t,1}, z_{t,2}, \ldots)$
5. Create **new z** code by **interpolation**
   E.g. $z = z_1 + \alpha(z_{t,\text{avg}} - z_1)$
6. **Decode z** with decoder.

Possible with more than 1 target features, similarly to algo on Slide 7.

$g_\theta(\mu_\phi(x_1))$ $\quad$ $g_\theta\left(z_1 + \alpha(z_{t,\text{avg}} - z_1)\right)$ $\quad$ Increasing α

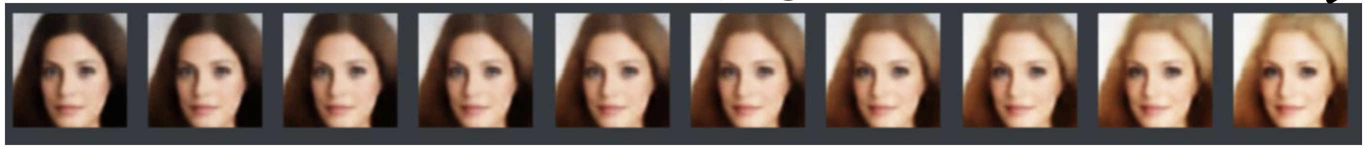


Image from: Steven Flores (link)

Requirement:
Need to know which data have the target characteristic (e.g. blonde hair)

Altering specific features of data with basic Auto-Encoder

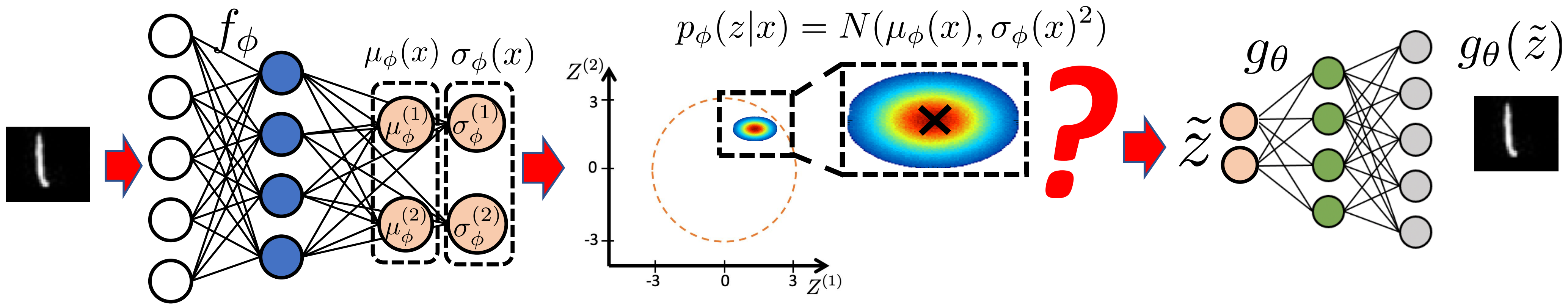
The same procedure can be done via a basic (deterministic) Auto-Encoder.

What issues do you expect and why?

Using VAEs for Compression and Reconstruction

After a VAE is trained, we may consider using it for compression and reconstruction.

Q: If we want **the best possible reconstruction of x, which code z** should we give as input to the decoder?

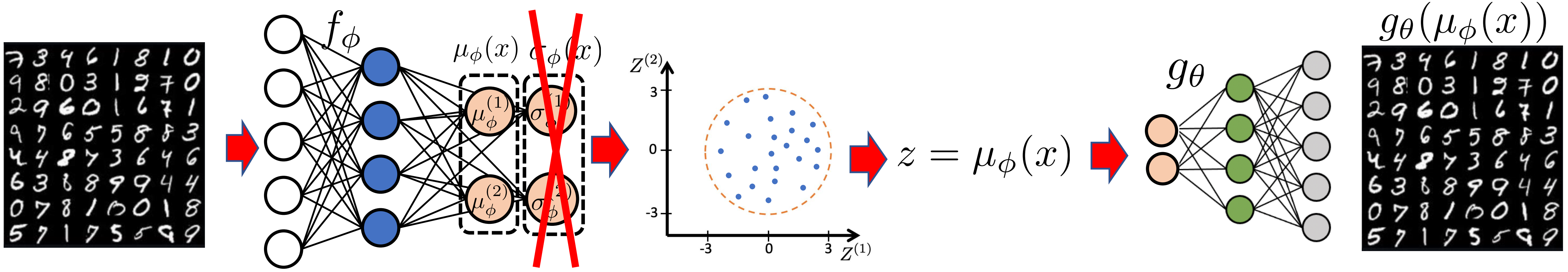


A: The predicted mean $\mu_\phi(x)$ by the encoder. This is the value of z that is most likely to be the correct encoding of x according to encoder.

Using VAEs for Compression and Reconstruction

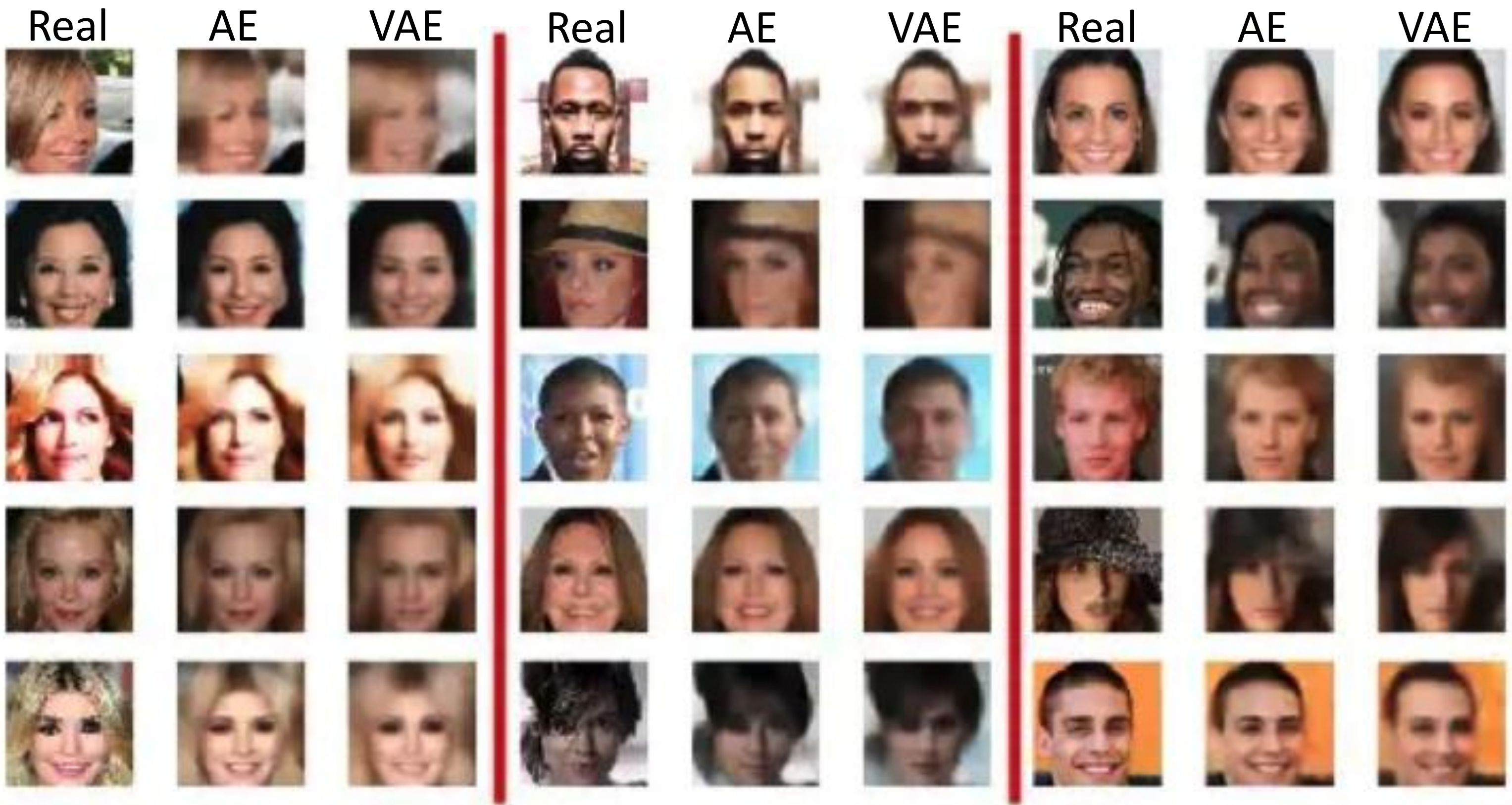
After a VAE is trained, we may consider using it for compression and reconstruction.

To get the best possible reconstruction from a VAE, we simply decode $z = \mu_\phi(x)$.
We do not use the predicted standard deviation, nor sample.



Reconstruction with VAE vs basic AE

VAE's capacity is used to optimize **both reconstruction and regularization** losses.
These two losses have **competing** goals!
Therefore its reconstructions may not be as good as those from a basic AE of similar capacity.
It is not what VAE is made for!



From: Steven Flores, Variational Autoencoders for Image Generation

In this video:

What we can use VAEs for:

- Generation of new data points
- Modifying data via interpolating between 2 inputs
- Modifying specific feature of an input

What VAE is not ideal for:

- Compression => Reconstruction

Next weeks:

- Generative Adversarial Networks
- Probabilistic derivation of VAEs (non-assessed, optional, if time allows)

Thank you very much