Deep Generative Modeling
- part 1 -

Hana Sebia
hana.sebia@inria.fr

December, 2023

AIstroSight, Inria, Lyon Center

Which face is real ?

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn function to map
$x \rightarrow y$

Examples: Classification, regression, object detection, semantic segmentation, etc.

Unsupervised Learning

Data: x
x is data, no labels!

Goal: Learn some *hidden* or *underlying structure* of the data

Examples: Clustering, feature or dimensionality reduction, etc.

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn function to map
$x \rightarrow y$

Examples: Classification, regression, object detection, semantic segmentation, etc.

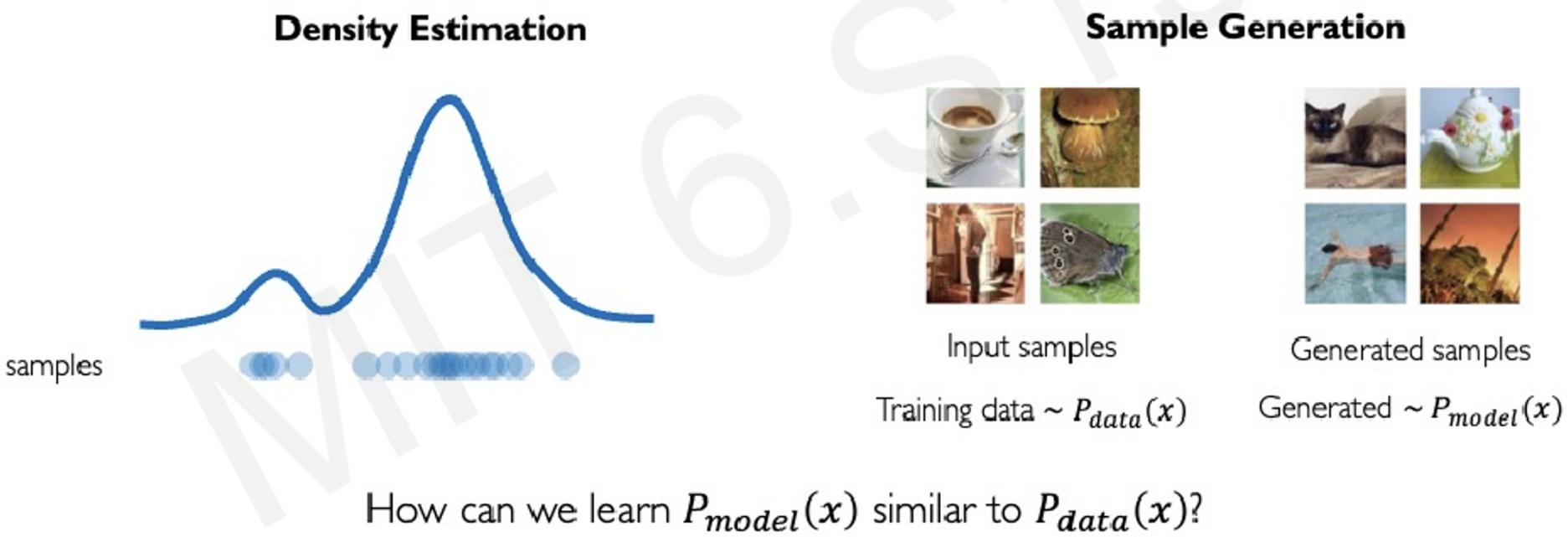
Unsupervised Learning

Data: x
x is data, no labels!

Goal: Learn the *hidden* or *underlying structure* of the data

Examples: Clustering, feature or dimensionality reduction, etc.

Goal: Take as input training samples from some distribution and learn a model that represent that distribution



Why Generative Models ? Debiasing

Capable of uncovering **underlying features** in a dataset

VS

Homogeneous skin color, pose

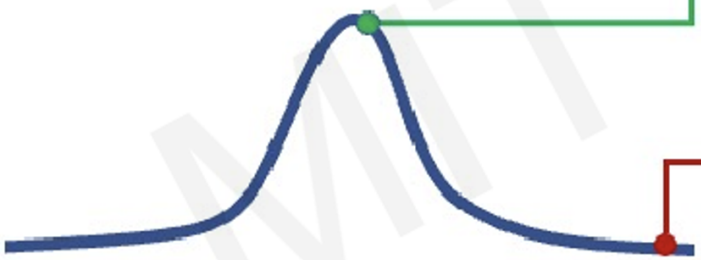
Diverse skin color, pose, illumination

How can we use this information to create fair and representative datasets?

Why Generative Models ? Outlier detection

- **Problem:** How can we detect when we encounter something new or rare?
- **Strategy:** Leverage generative models, detect outliers in the distribution
- Use outliers during training to improve even more!

95% of Driving Data:
(1) sunny, (2) highway, (3) straight road



Detect outliers to avoid unpredictable behavior when training

Edge Cases

Harsh Weather

Pedestrians

Generative Modeling

Generative AI

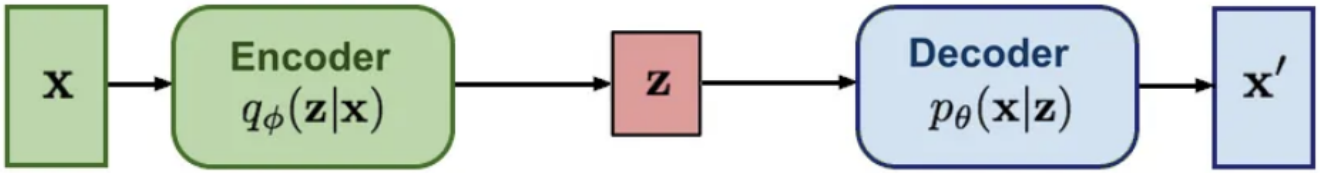
- Unsupervised process
- Take training samples from some unknown distribution and learn a model that can capture this distribution

Generative Model Trilemma

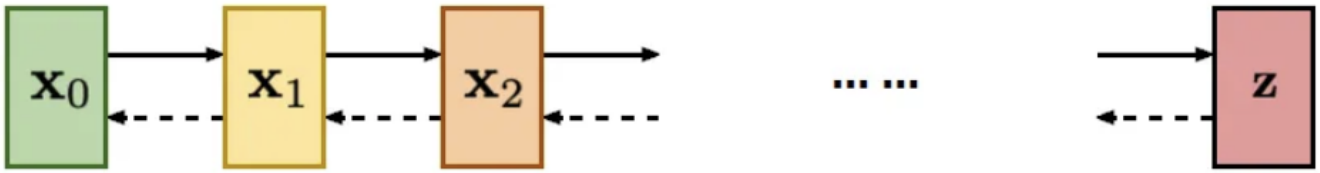
- **Quality of samples** : the generated data should be realistic and accurate compared to the actual data distribution
- **Speed of sampling** : the number of network passes required to generate a new sample
- **Mode coverage** : the model have to capture the entire distribution to ensure sample diversity

GAN: Adversarial training

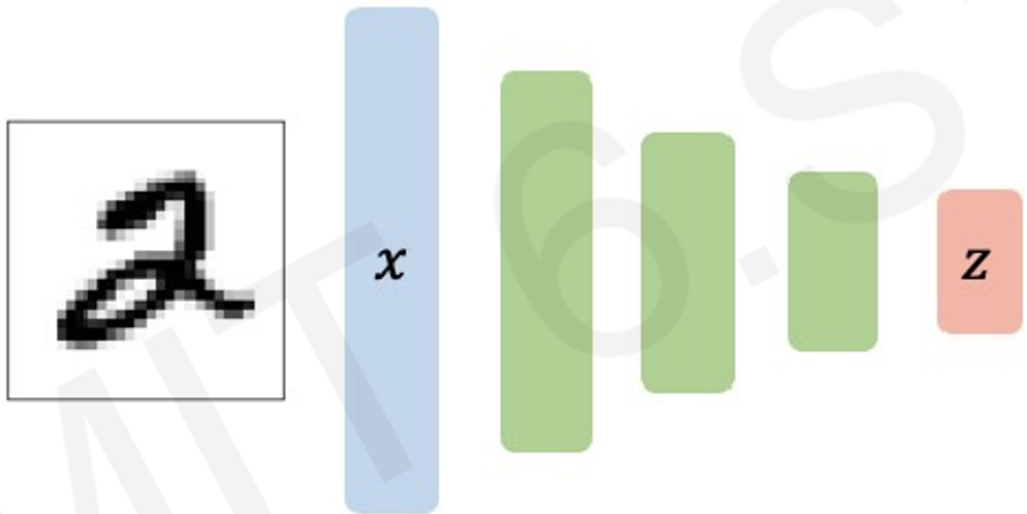
VAE: maximize variational lower bound



Diffusion models: Gradually add Gaussian noise and then reverse



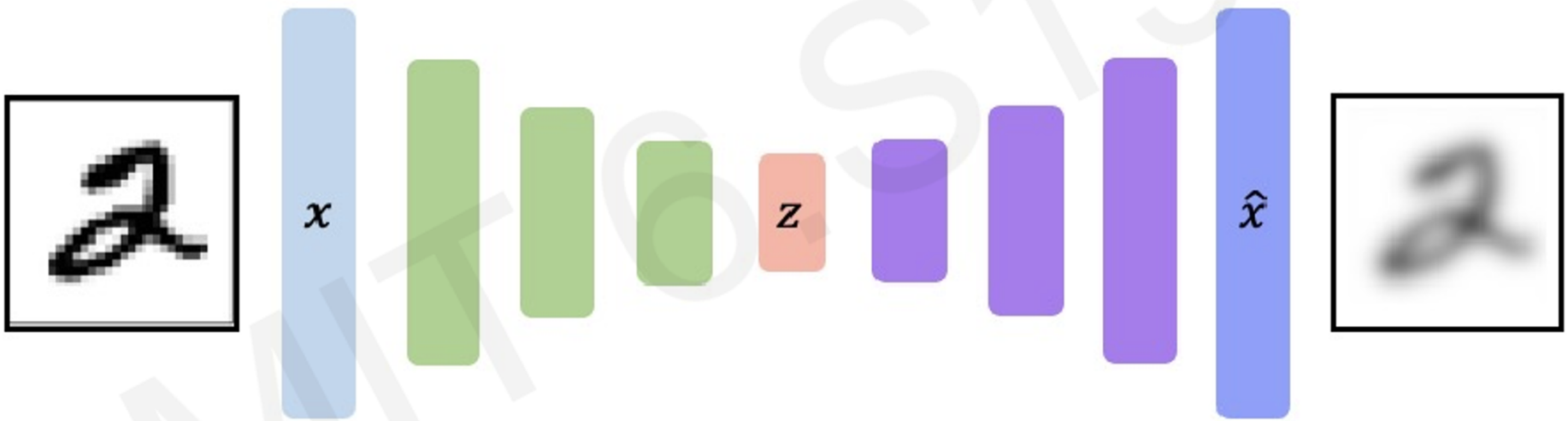
Unsupervised approach for learning a **lower-dimensional** feature representation from unlabeled training data



Why do we care about a low-dimensional z?

"Encoder" learns mapping from the data, x, to a low-dimensional latent space, z

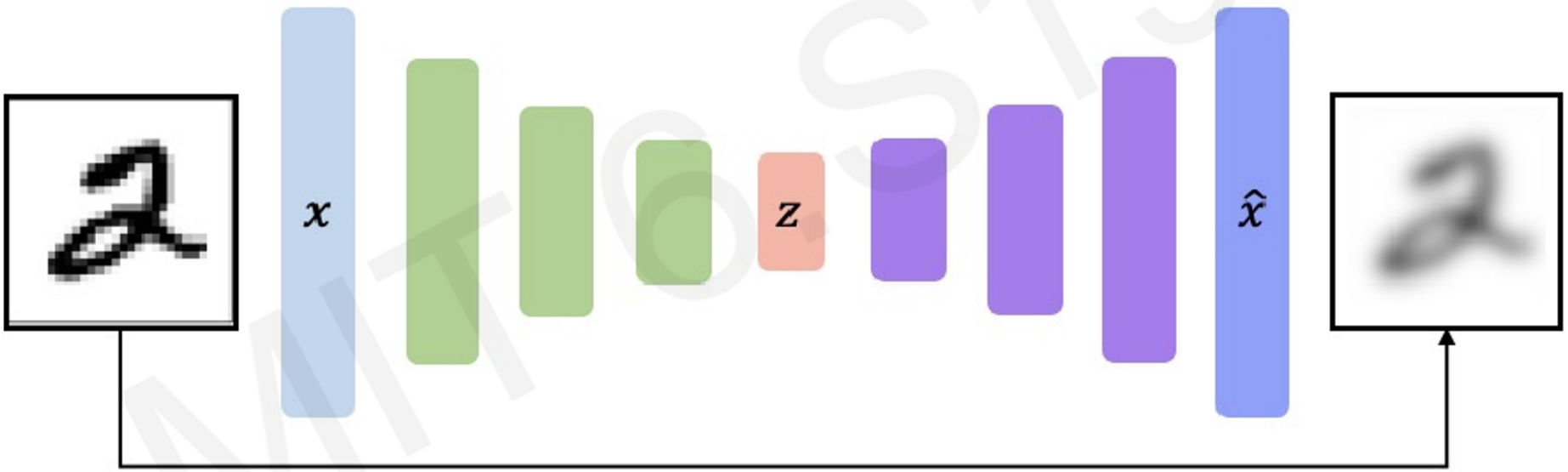
How can we learn this latent space?
Train the model to use these features to **reconstruct the original data**



"Decoder" learns mapping back from latent space, z,
to a reconstructed observation, $\hat{x}$

How can we learn this latent space?

Train the model to use these features to **reconstruct the original data**

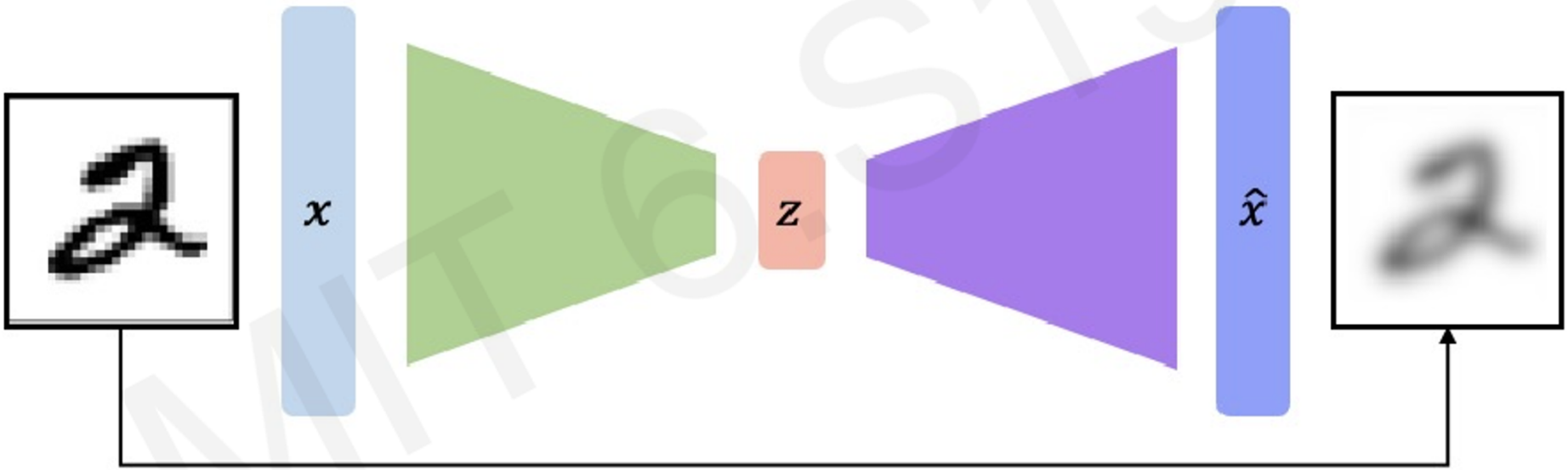


$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2$$

Loss function doesn't use any labels!!

How can we learn this latent space?

Train the model to use these features to **reconstruct the original data**



$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2$$

Loss function doesn't use any labels!!

Dimension of latent space

Autoencoding is a form of compression!
Smaller latent space will force a larger training bottleneck

2D latent space

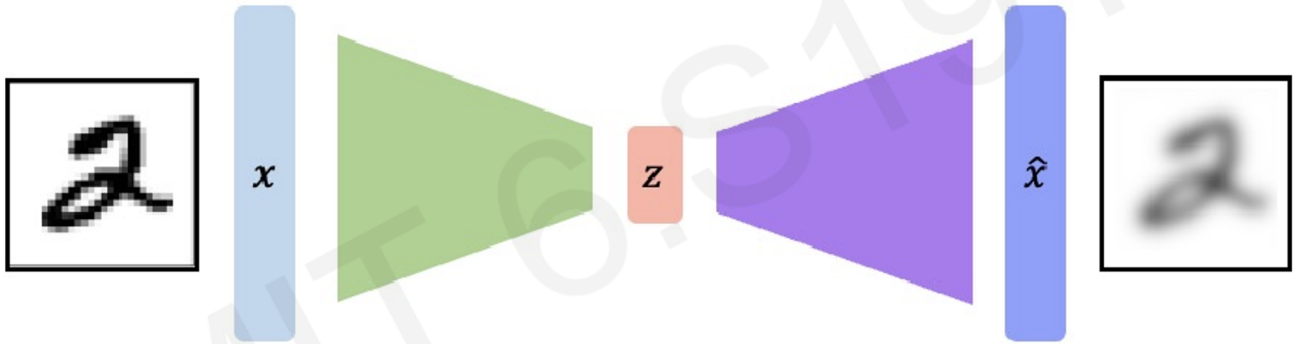
5D latent space

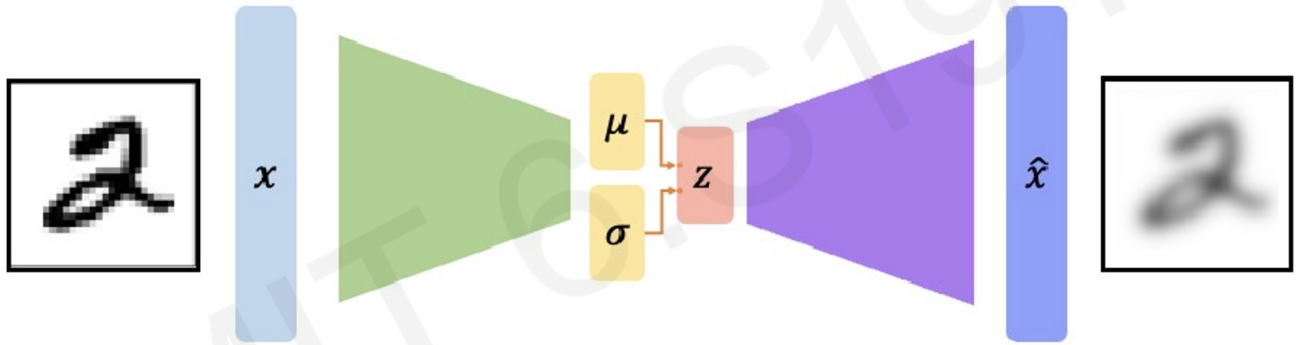
Ground Truth

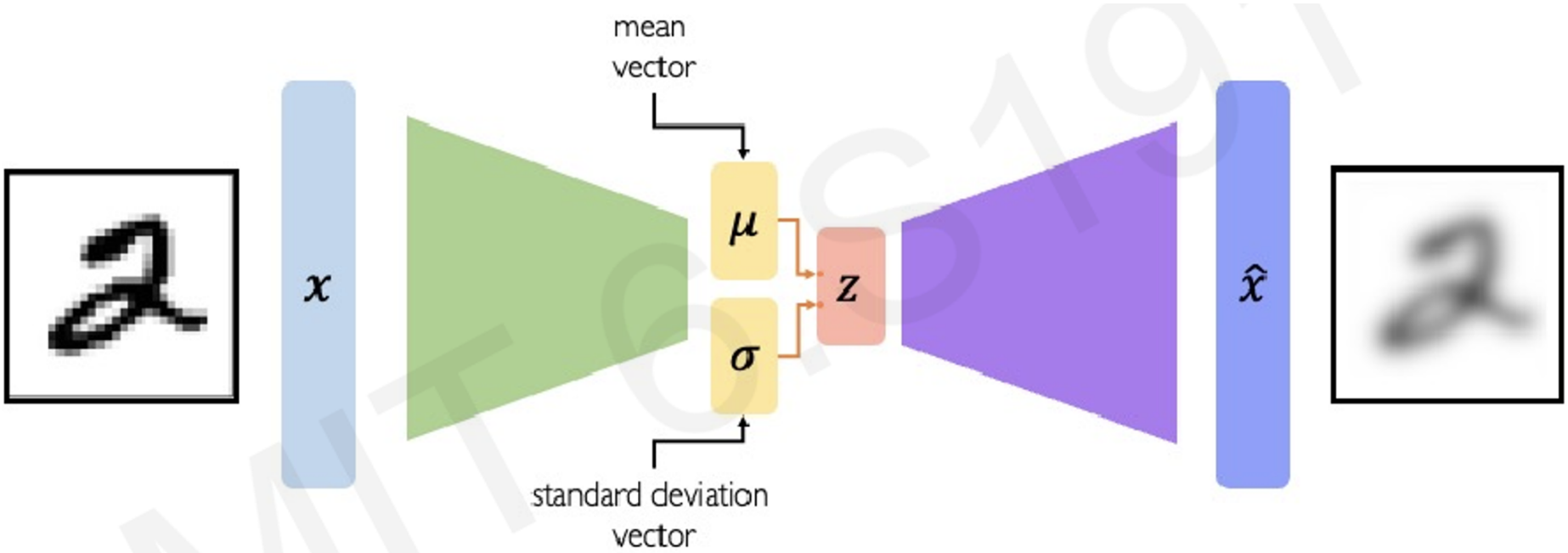
Autoencoders for representation learning

- **Bottleneck hidden layer** forces network to learn a compressed latent space representation
- **Reconstruction loss** forces the latent representation to capture (or encode) as much *information* about the data as possible
- **Autoencoding** = **Auto**matically **encoding** data; *Auto* = **self**-encoding

key difference between traditional autoencoders and variational autoencoders



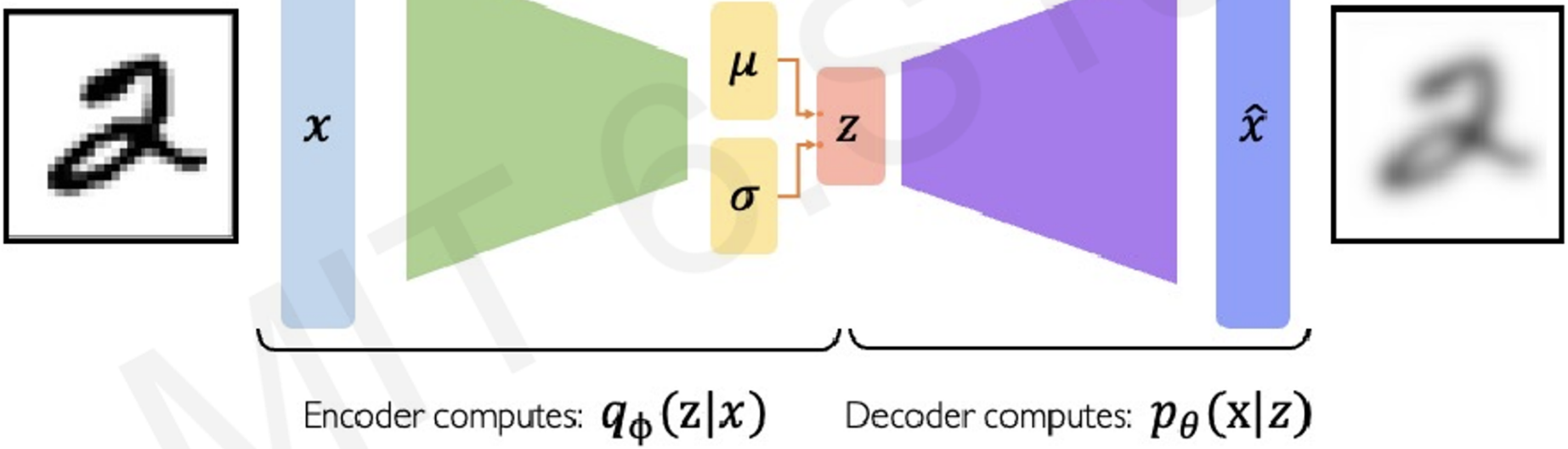




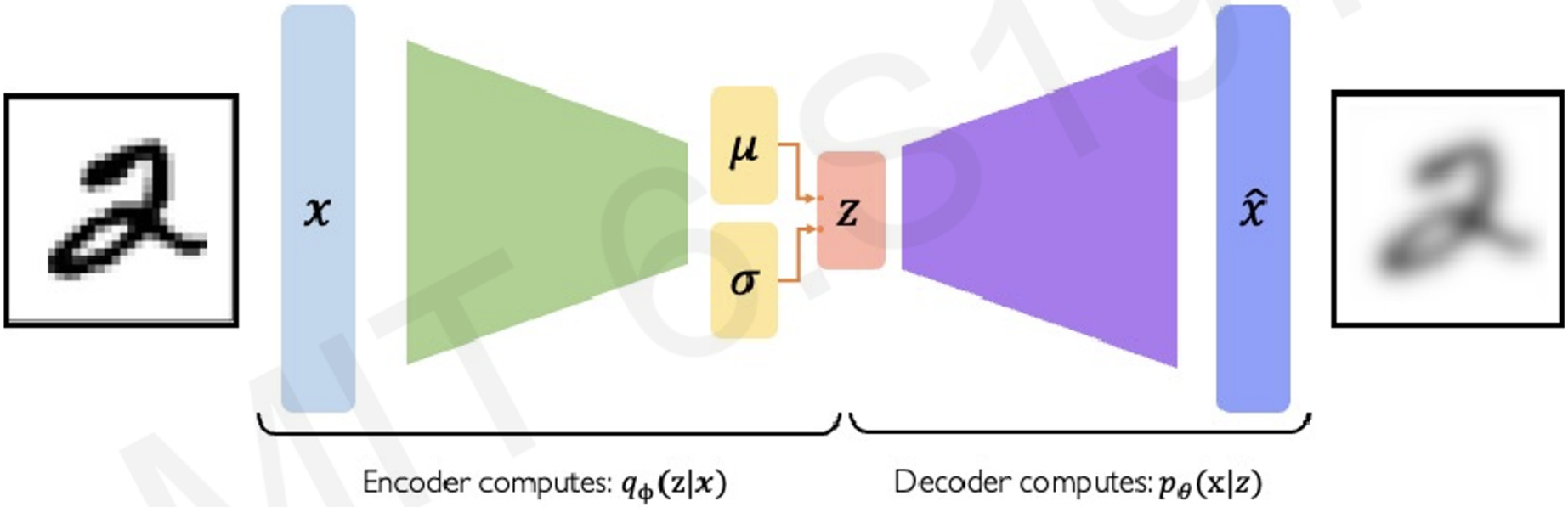
Variational autoencoders are a probabilistic twist on autoencoders!

Sample from the mean and standard deviation to compute latent sample

VAE Optimization

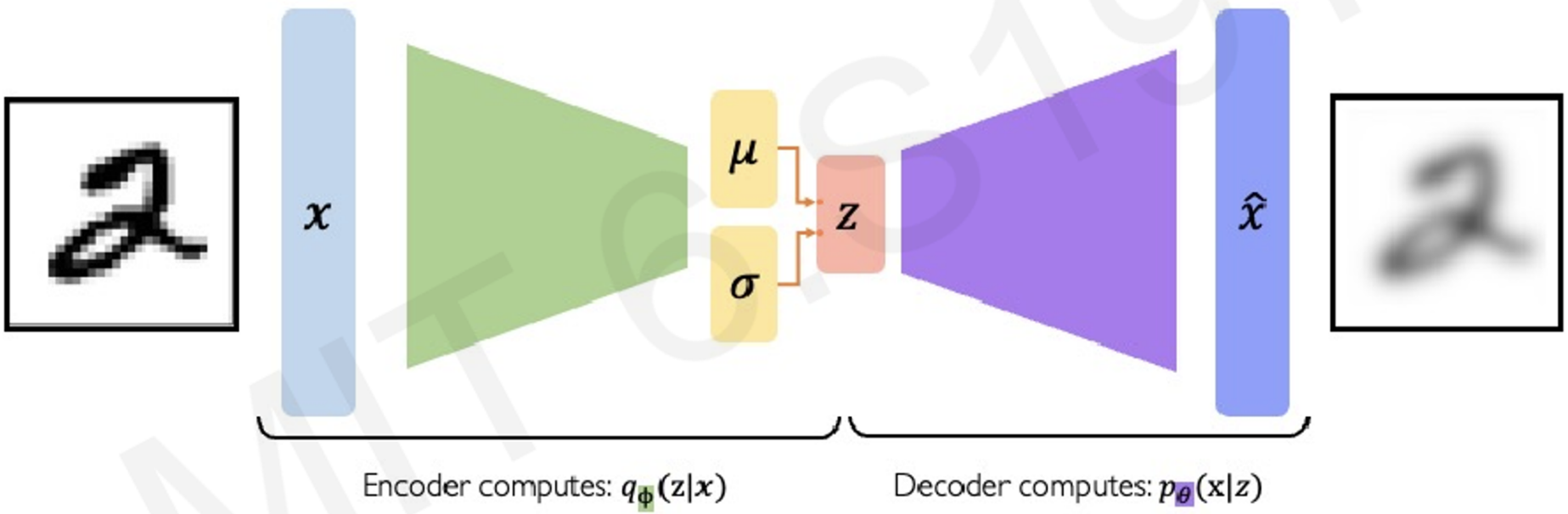


VAE Optimization



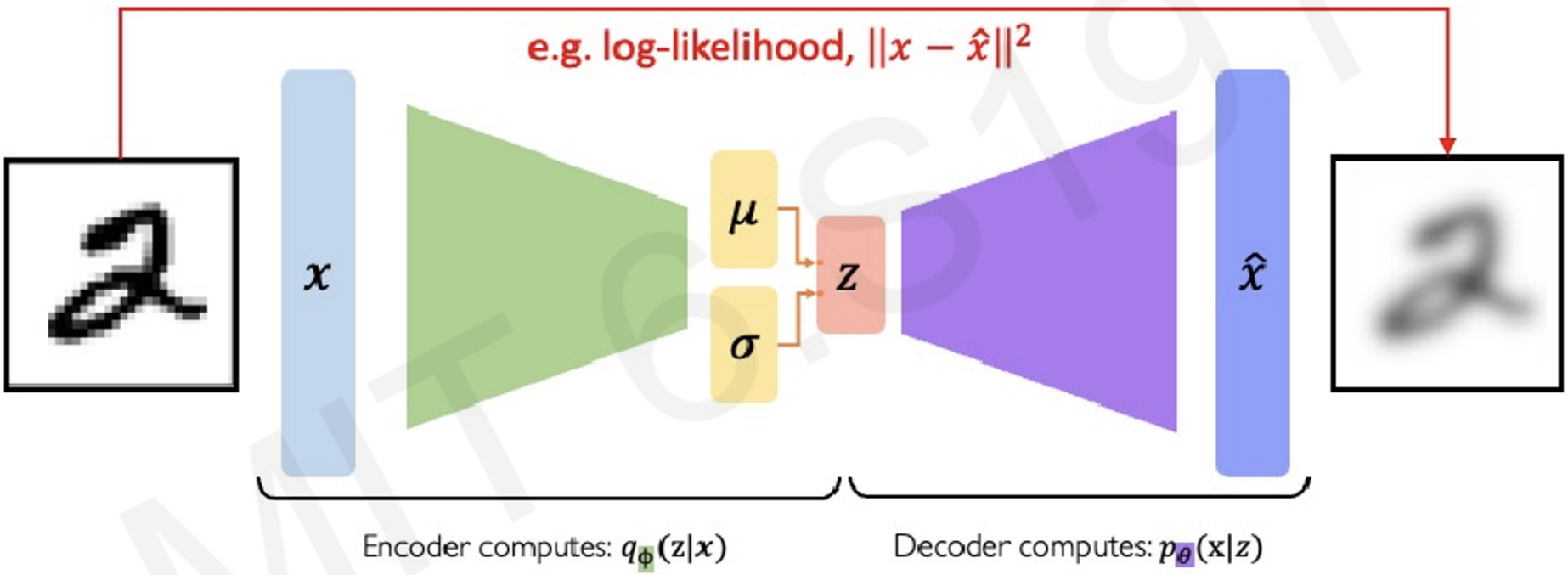
$$\mathcal{L}(\phi, \theta, x) = \text{(reconstruction loss)} + \text{(regularization term)}$$

VAE Optimization



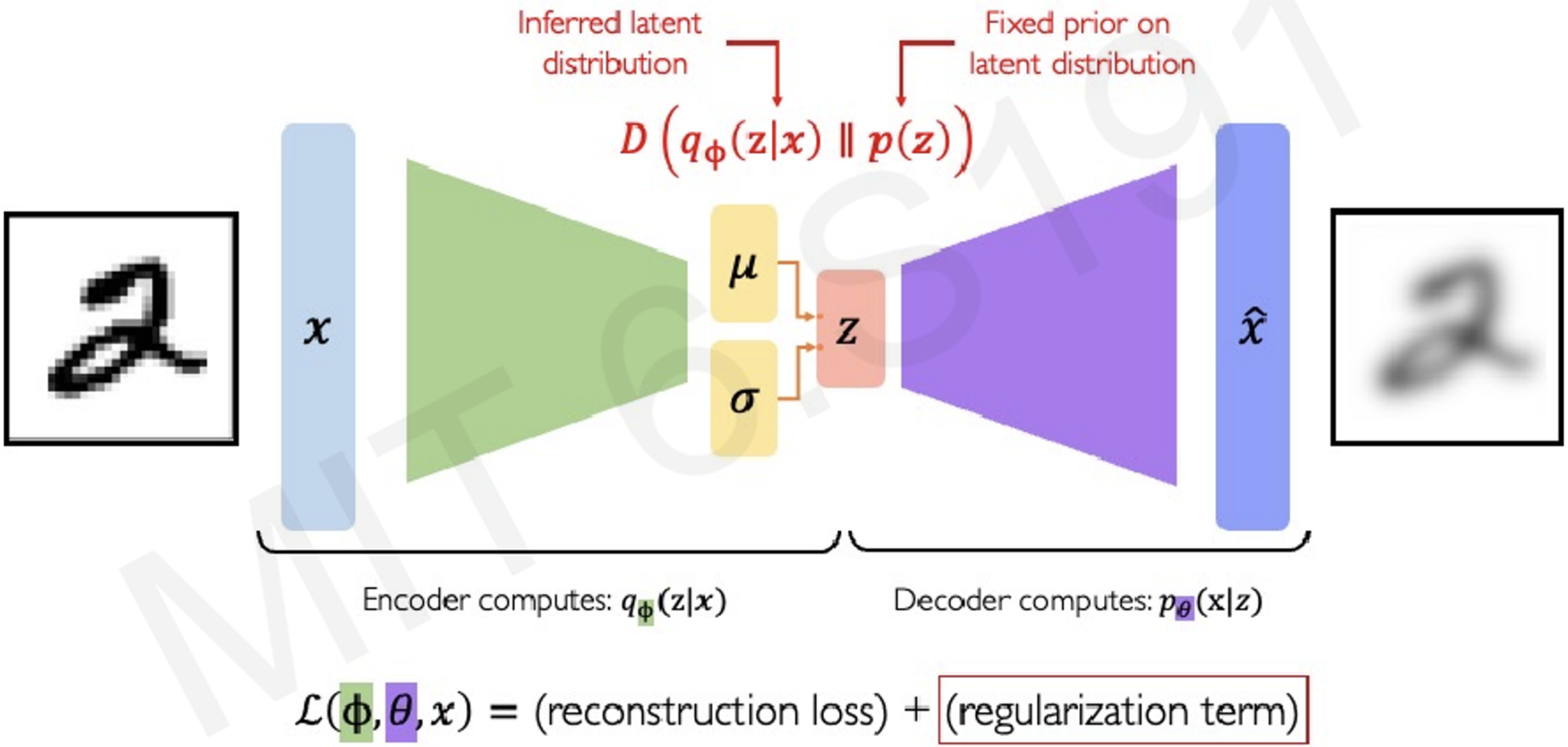
$$\mathcal{L}(\phi, \theta, x) = \text{(reconstruction loss)} + \text{(regularization term)}$$

VAE Optimization



$$\mathcal{L}(\phi, \theta, x) = \boxed{\text{(reconstruction loss)}} + \text{(regularization term)}$$

VAE Optimization



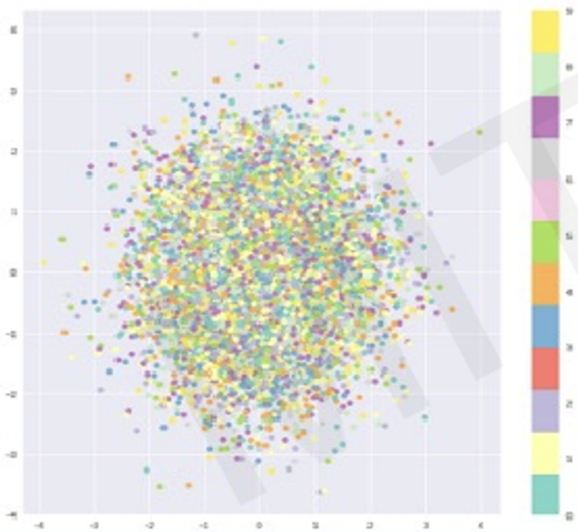
$$\mathcal{L}(\phi, \theta, x) = \text{(reconstruction loss)} + \boxed{\text{(regularization term)}}$$

Priors on the latent distribution

$$D\left(q_\phi(z|x) \parallel p(z)\right)$$

Inferred latent distribution

Fixed prior on latent distribution



Common choice of prior – Normal Gaussian:

$$p(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

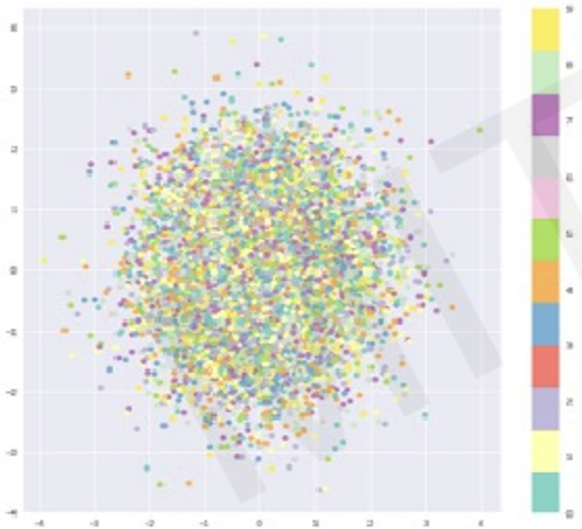
- Encourages encodings to distribute encodings evenly around the center of the latent space
- Penalize the network when it tries to "cheat" by clustering points in specific regions (i.e., by memorizing the data)

Priors on the latent distribution

$$D\left(q_\phi(z|x) \parallel p(z)\right)$$

$$= -\frac{1}{2}\sum_{j=0}^{k-1}\left(\sigma_j + \mu_j^2 - 1 - \log \sigma_j\right)$$

KL-divergence between the two distributions



Common choice of prior – Normal Gaussian:

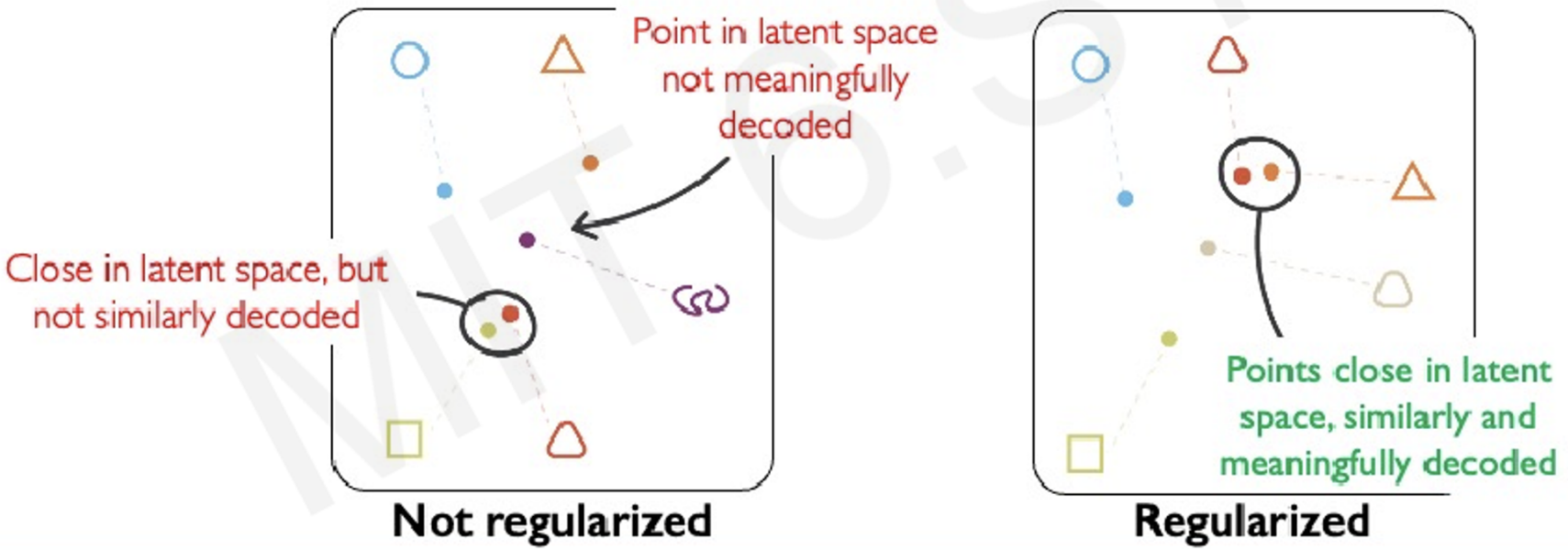
$$p(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

- Encourages encodings to distribute encodings evenly around the center of the latent space
- Penalize the network when it tries to "cheat" by clustering points in specific regions (i.e., by memorizing the data)

Intuition on regularization and Normal prior

What properties do we want to achieve from regularization?

1. **Continuity:** points that are close in latent space → similar content after decoding
2. **Completeness:** sampling from latent space → "meaningful" content after decoding



Intuition on regularization and Normal prior

1. **Continuity**: points that are close in latent space → similar content after decoding
2. **Completeness:** sampling from latent space → "meaningful" content after decoding

Encoding as a distribution does not guarantee these properties!

Small variances → Pointed distributions

Different means → Discontinuities

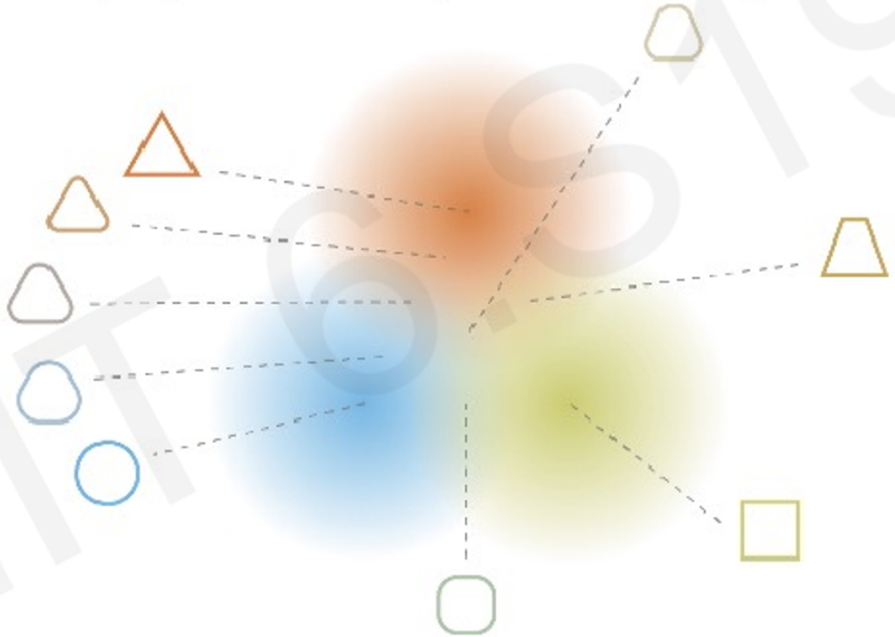
Not regularized

Normal prior → continuity + completeness

Regularized

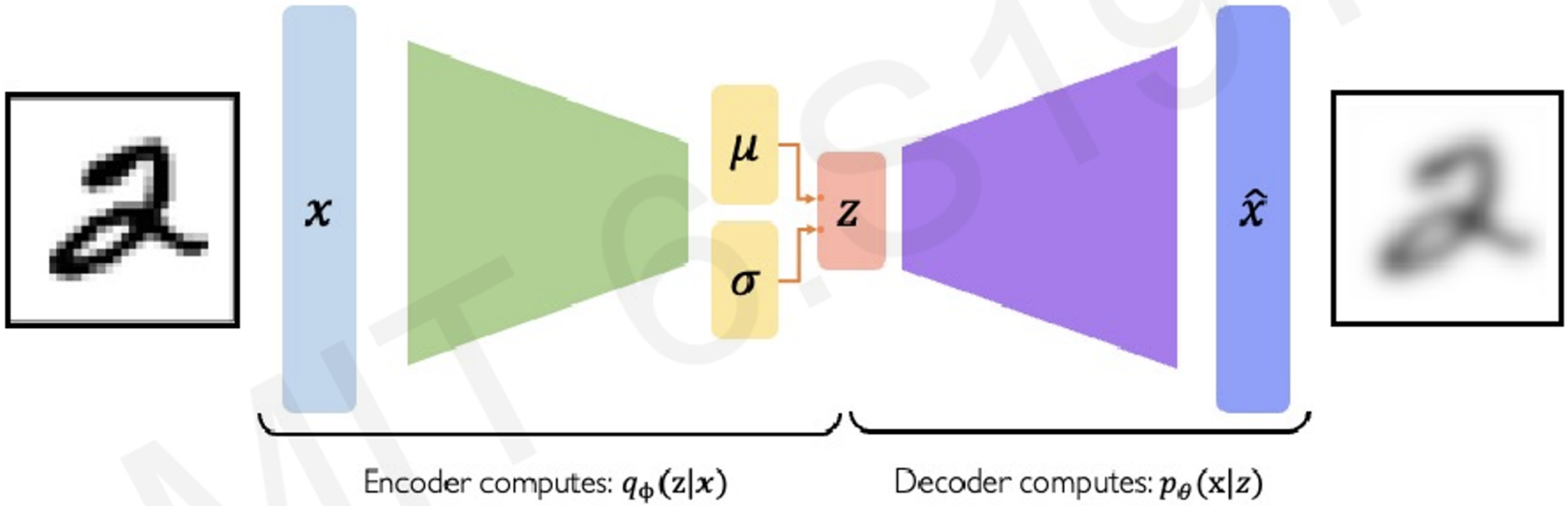
Intuition on regularization and Normal prior

1. **Continuity**: points that are close in latent space → similar content after decoding
2. **Completeness:** sampling from latent space → "meaningful" content after decoding



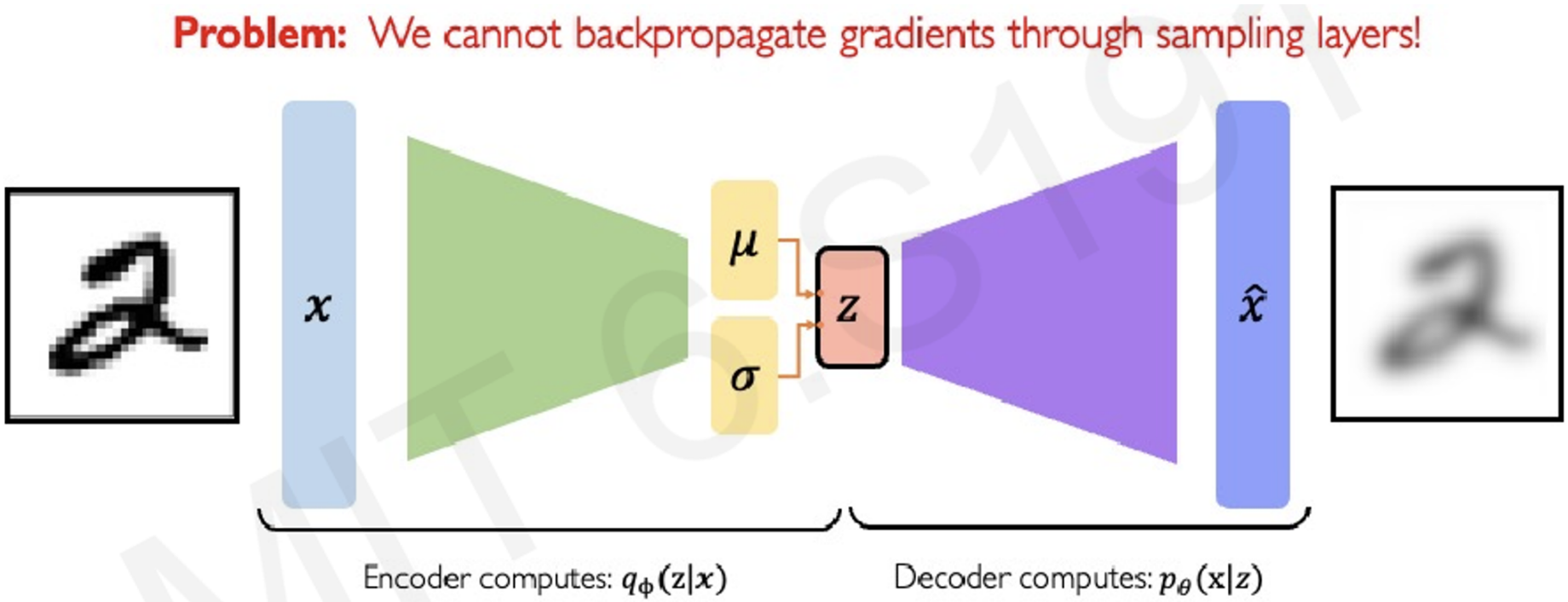
Regularization with Normal prior helps enforce **information gradient** in the latent space.

VAE computation graph



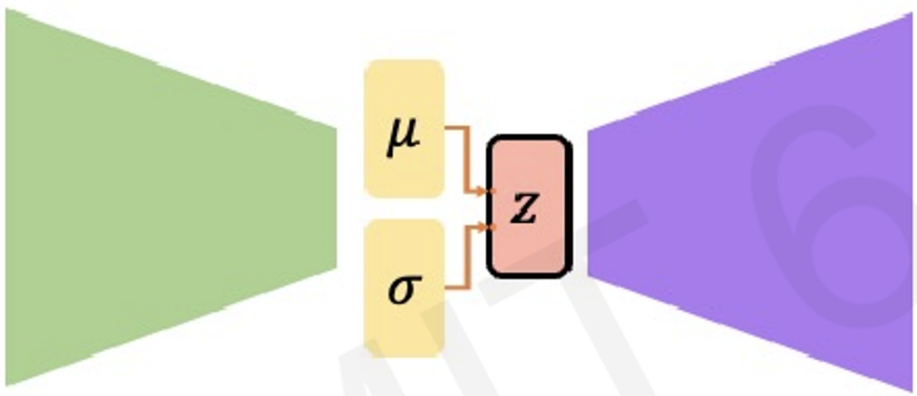
$$\mathcal{L}(\phi, \theta, x) = \text{(reconstruction loss)} + \text{(regularization term)}$$

VAE computation graph



$$\mathcal{L}(\phi, \theta, x) = \text{(reconstruction loss)} + \text{(regularization term)}$$

Reparametrizing the sampling layer



Key Idea:

~~$z \sim \mathcal{N}(\mu, \sigma^2)$~~

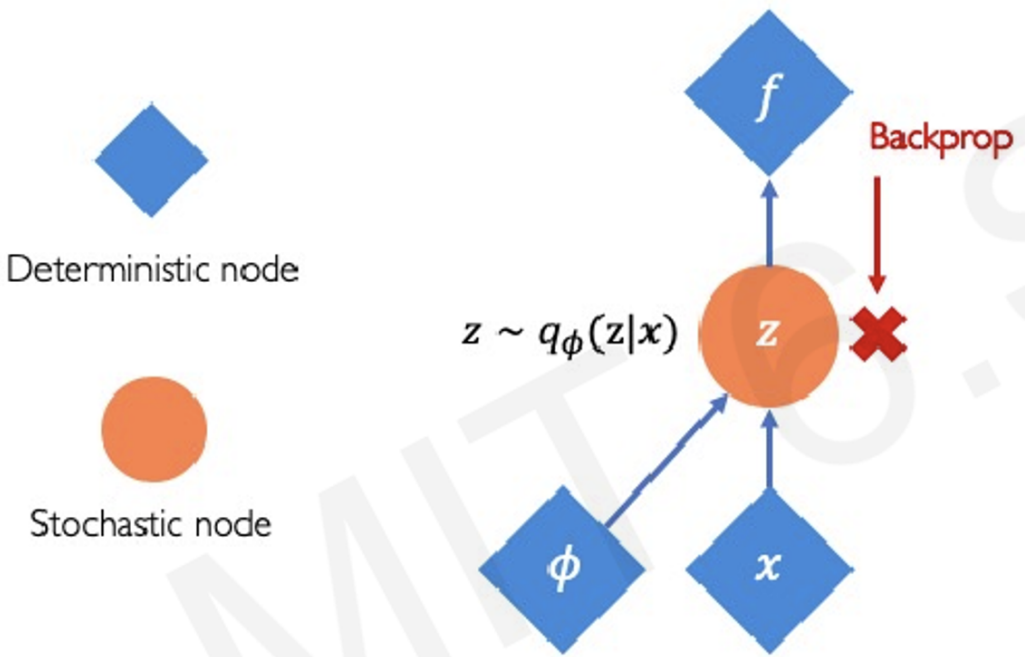
Consider the sampled latent vector z as a sum of

- a fixed μ vector,
- and fixed σ vector, scaled by random constants drawn from the prior distribution

$$\Rightarrow z = \mu + \sigma \odot \varepsilon$$

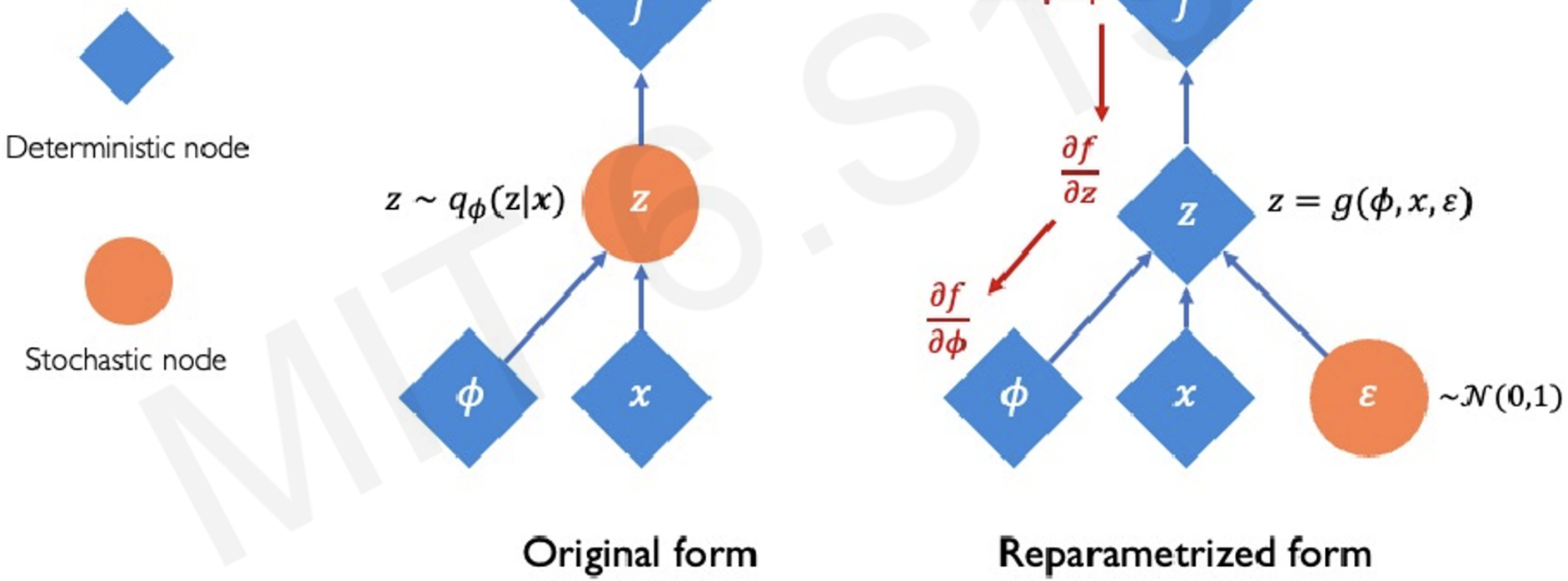
where $\varepsilon \sim \mathcal{N}(0,1)$

Reparametrizing the sampling layer



Original form

Reparametrizing the sampling layer



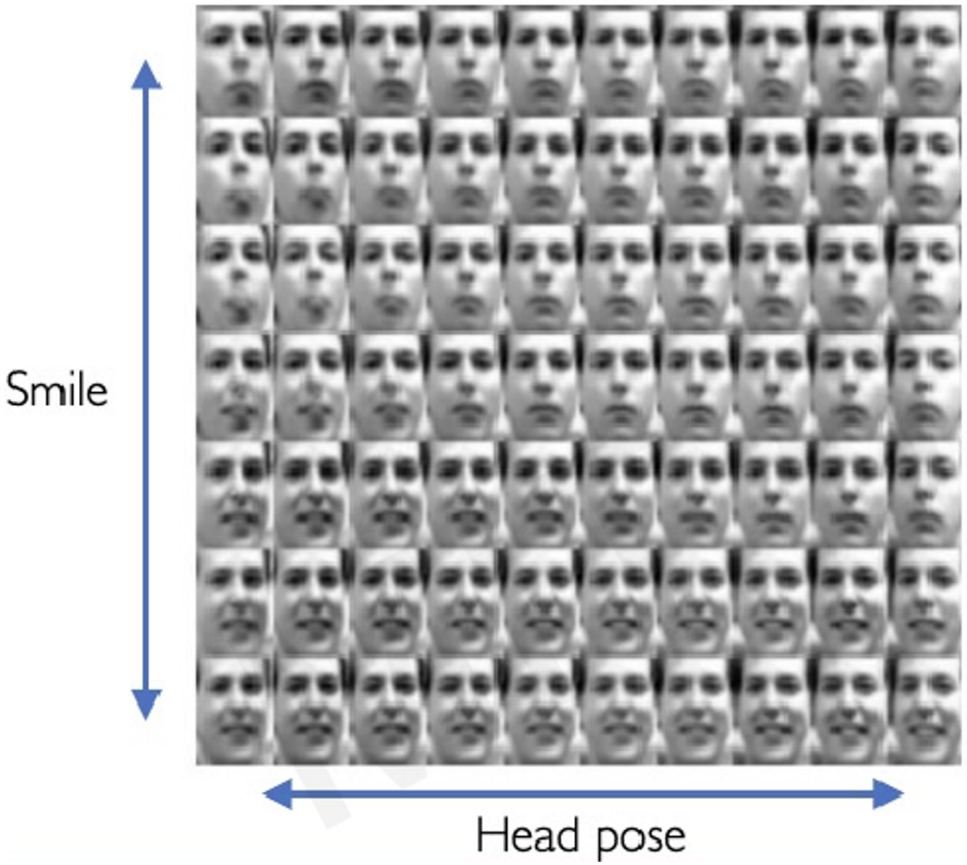
VAEs: Latent perturbation

Slowly increase or decrease a **single latent variable**
Keep all other variables fixed

Head pose

Different dimensions of z encodes **different interpretable latent features**

VAEs: Latent perturbation



Ideally, we want latent variables that are uncorrelated with each other

Enforce diagonal prior on the latent variables to encourage independence

Disentanglement

Latent space distenglement with $\beta - VAEs$

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

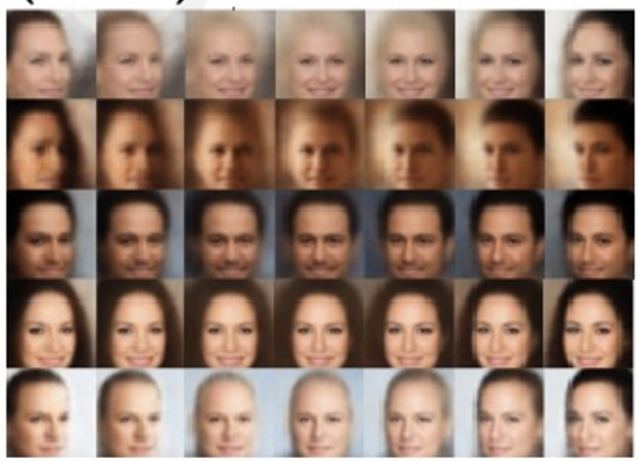
Reconstruction term **Regularization term**

β > 1: constrain latent bottleneck, encourage efficient latent encoding → disentanglement

Head rotation (azimuth)

Smile also changing!

Standard VAE (β = 1)



β-VAE (β = 250)

Smile relatively constant!

Why latent variable models? Debiasing

Capable of uncovering **underlying features** in a dataset

Homogeneous skin color, pose

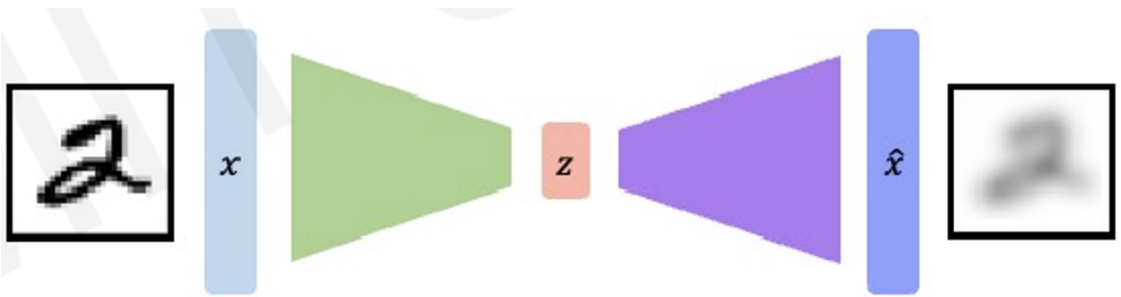
VS

Diverse skin color, pose, illumination

How can we use this information to create fair and representative datasets?

VAE summary

1. Compress representation of world to something we can use to learn
2. Reconstruction allows for unsupervised learning (no labels!)
3. Reparameterization trick to train end-to-end
4. Interpret hidden latent variables using perturbation
5. Generation of new samples



Trilemma for VAE

- **Low-fidelity samples** : due to the overlap of several distributions in the latent space
- **High diversity samples** : Likelihood maximization forces to cover all modes
- **Fast Sampling** : require just one pass of the decoder network to turn a latent representations into a realistic data sample