Today: Outline

- *Recap:* **Generative Models**
- *Recap:* **Autoencoder**
- **Variational Autoencoder**
- *Break-out Session:* **Project Help**

- Reminders:
  - Thu Jun 17: *is a free-choice lecture*

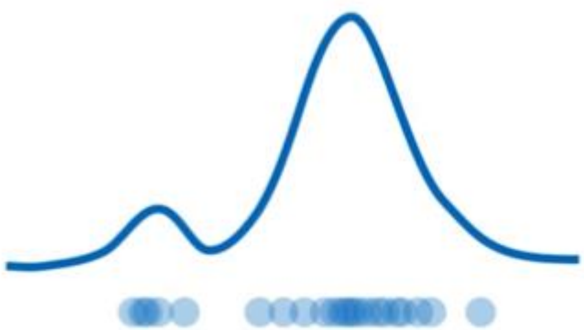
  - Fri Jun 18: *Problem Set 2 due*
  - Mon Jun 21: *Pre-lecture Material 6 due*

  - Tue Jun 22: *Exam during class time*
  (and ~12 hrs before for remote only students)
  - Practice problems available on Resources

Generative modeling

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

Density Estimation

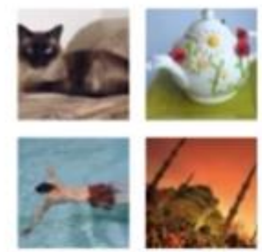


samples

Sample Generation

Input samples

Training data $\sim P_{data}(x)$



Generated samples

Generated $\sim P_{model}(x)$

How can we learn $P_{model}(x)$ similar to $P_{data}(x)$?

Why generative models? Debiasing

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

vs

Homogeneous skin color, pose

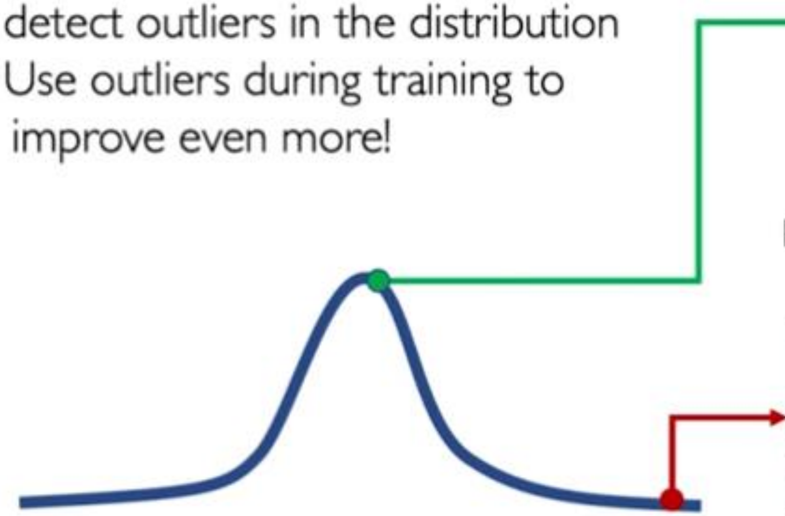
Diverse skin color, pose, illumination

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

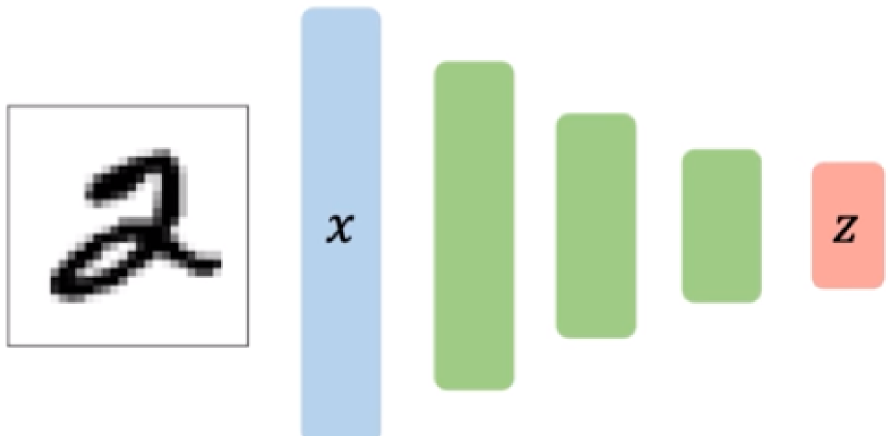
Pedestrians

What is a latent variable?

Myth of the Cave

Autoencoders: background

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



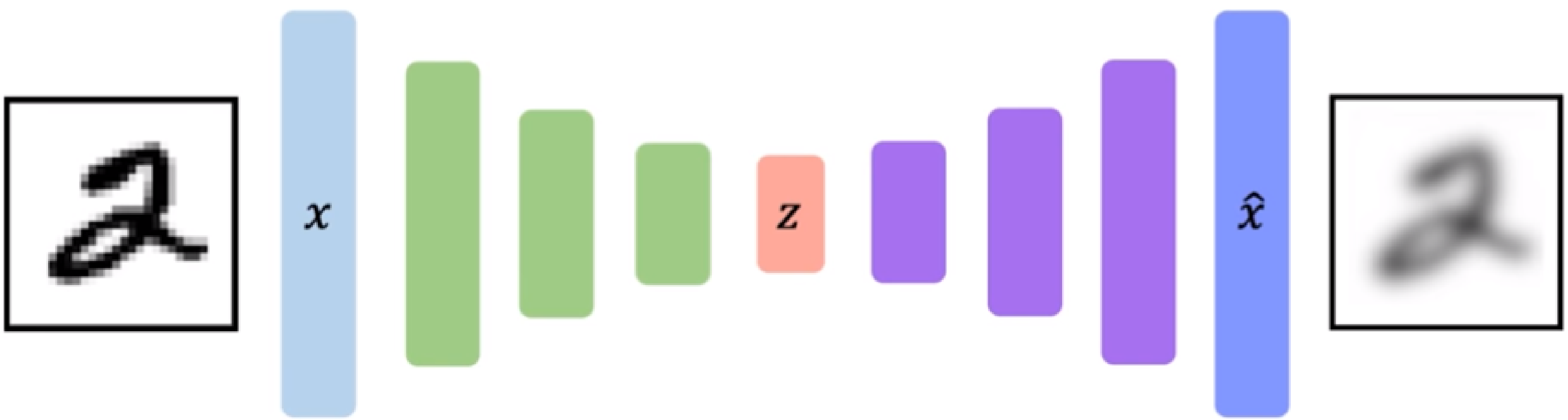
Why do we care about a low-dimensional z?

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

Autoencoders: background

How can we learn this latent space?

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

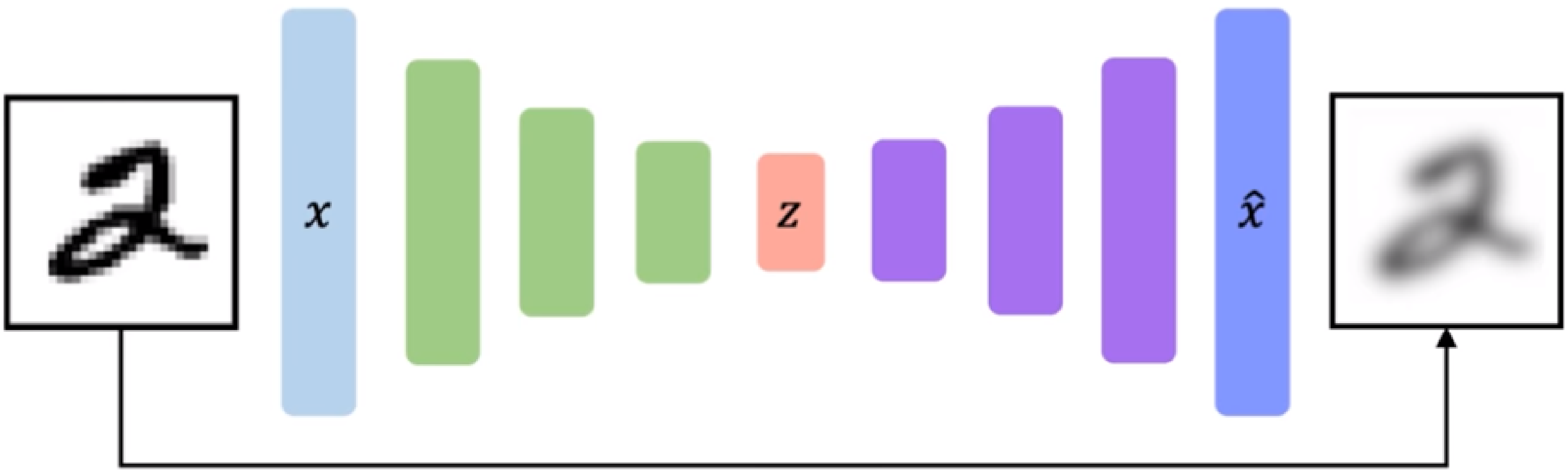


"Decoder" learns mapping back from latent space, z, to a reconstructed observation, $\hat{x}$

Autoencoders: background

How can we learn this latent space?

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

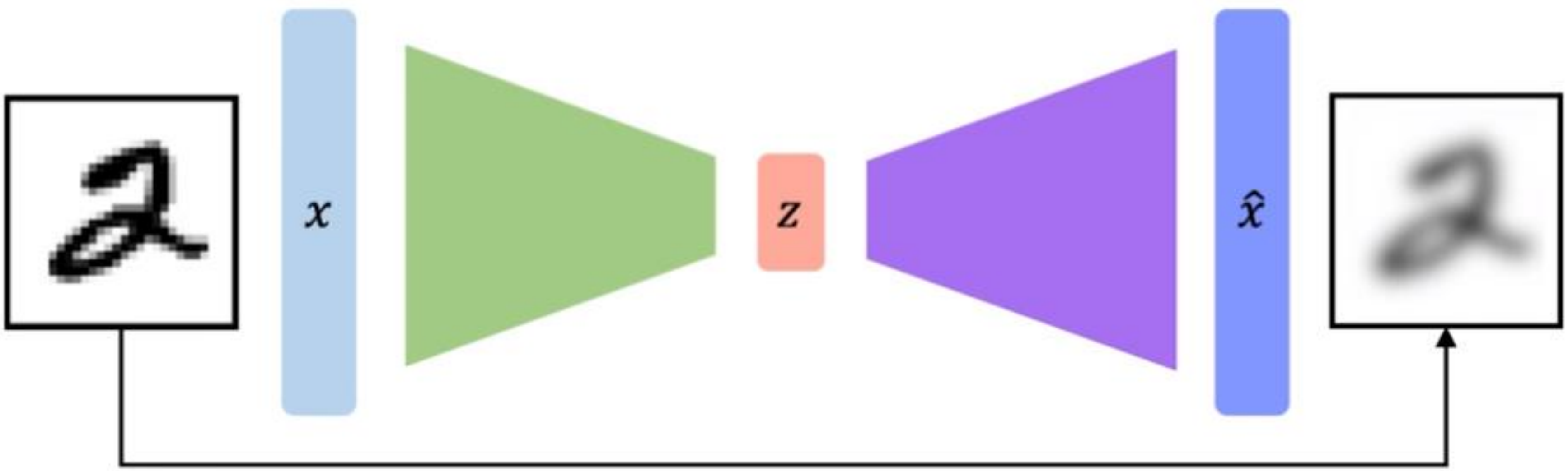


$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2$$

Autoencoders: background

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!!

Dimensionality of latent space → reconstruction quality

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

2D latent space

5D latent space

Ground Truth

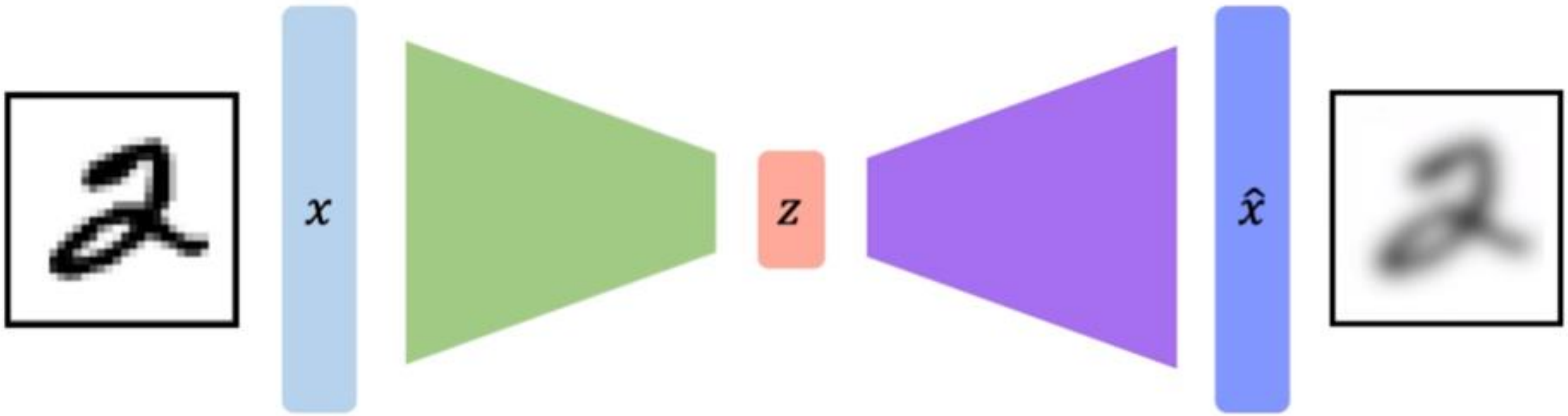
Autoencoders for representation learning

Bottleneck hidden layer forces network to learn a compressed latent representation

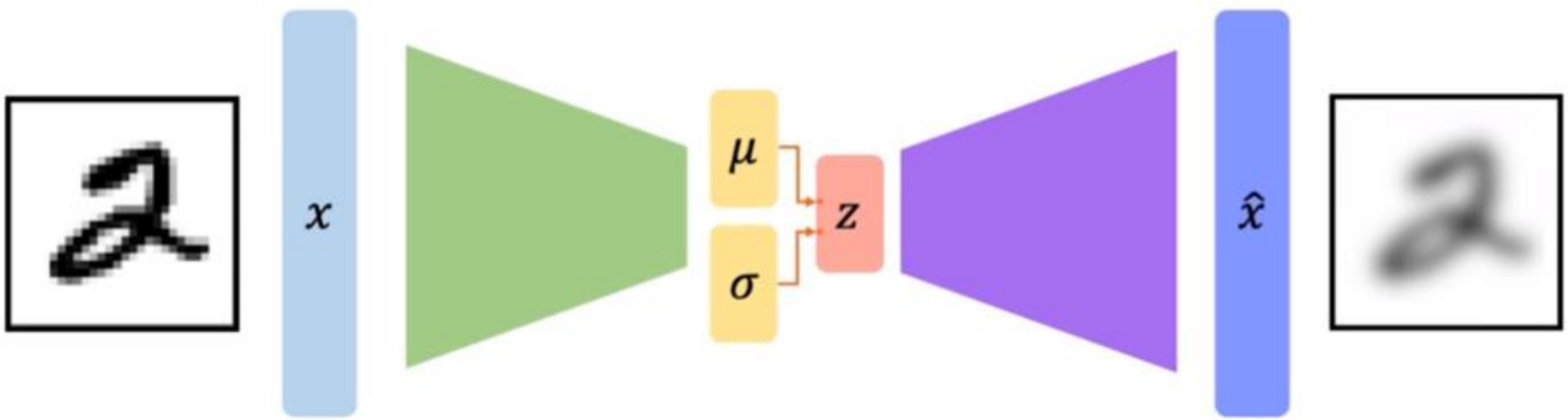
Reconstruction loss forces the latent representation to capture (or encode) as much "information" about the data as possible

Autoencoding = **Auto**matically **encoding** data

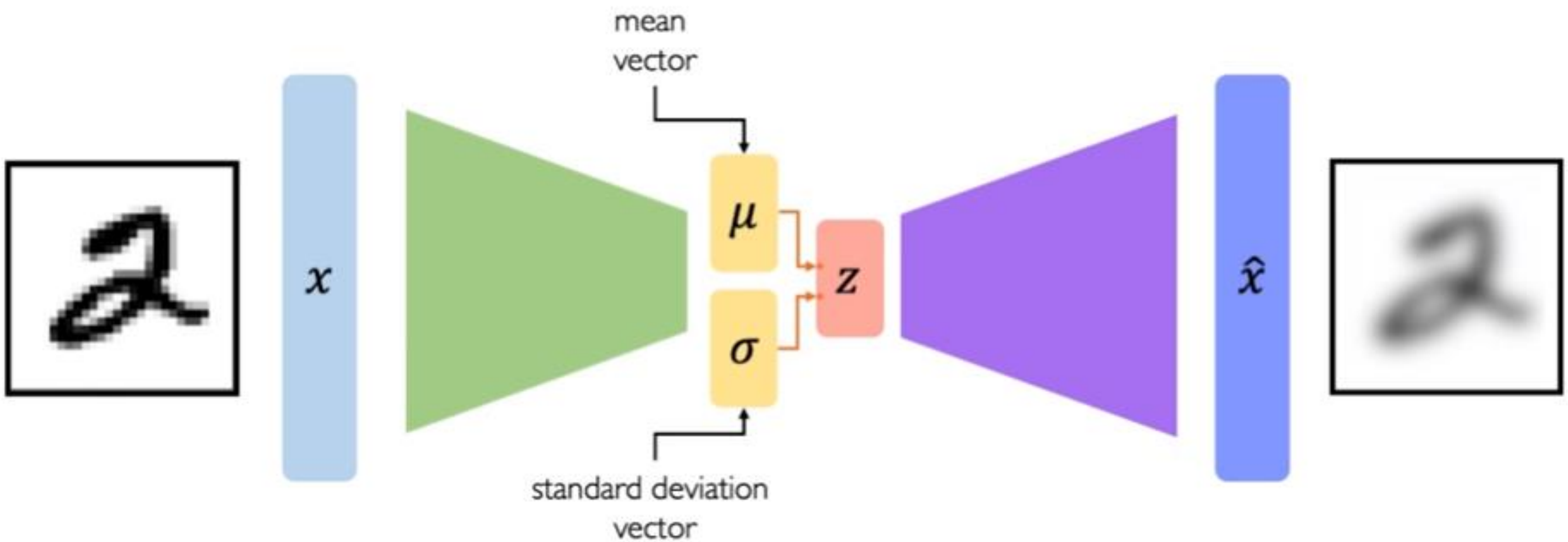
Traditional autoencoders



VAEs: key difference with traditional autoencoder

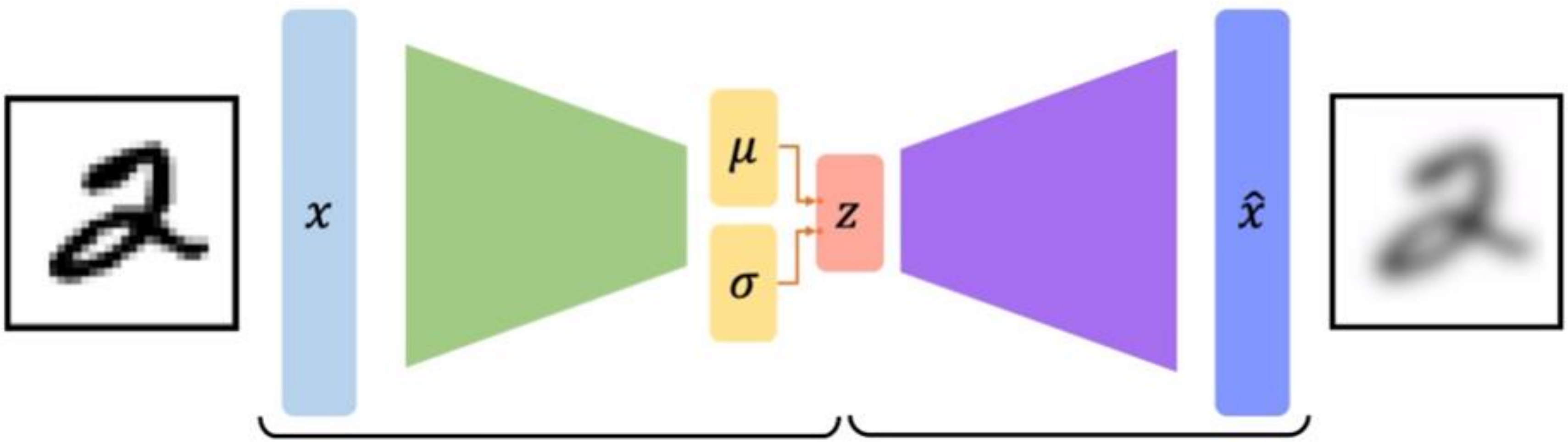


VAEs: key difference with traditional autoencoder



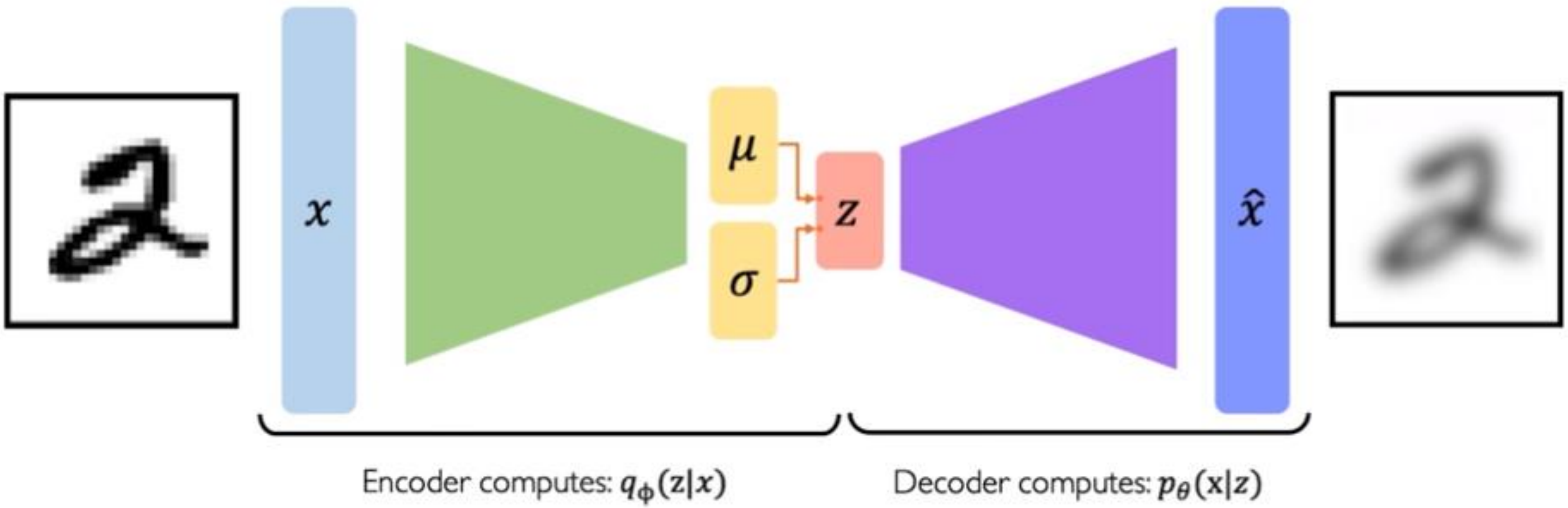
Variational autoencoders are a probabilistic twist on autoencoders!
Sample from the mean and standard deviation to compute latent sample

VAE optimization



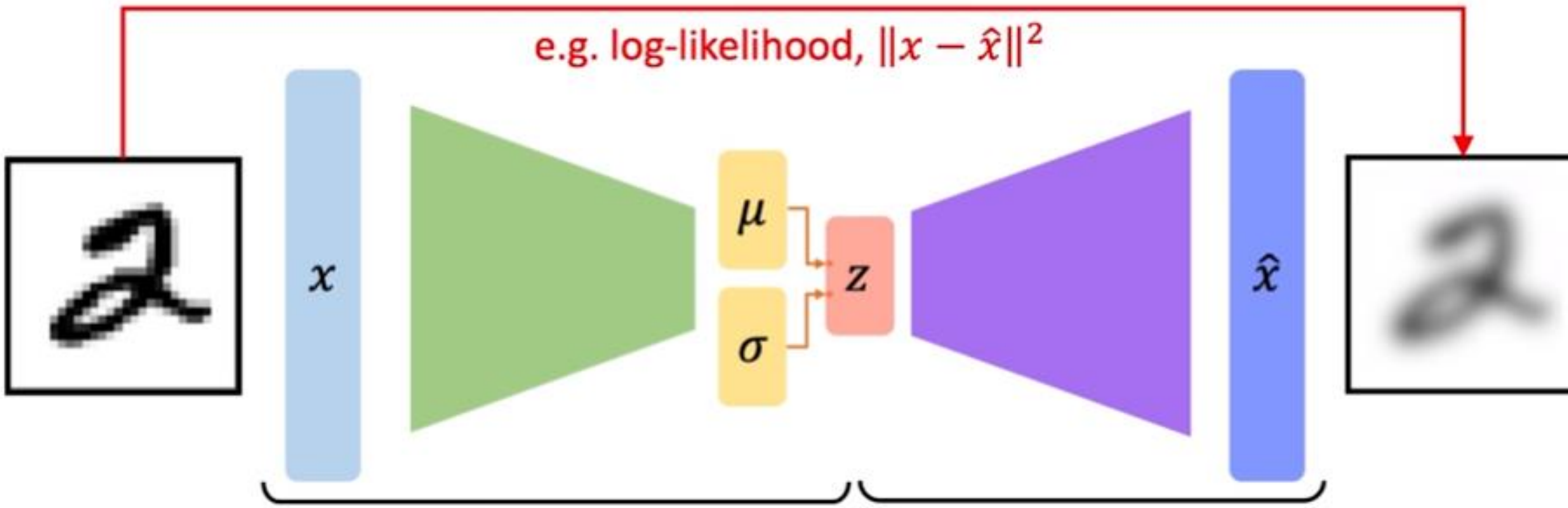
Encoder computes: $q_\phi(z|x)$ Decoder computes: $p_\theta(x|z)$

VAE optimization



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

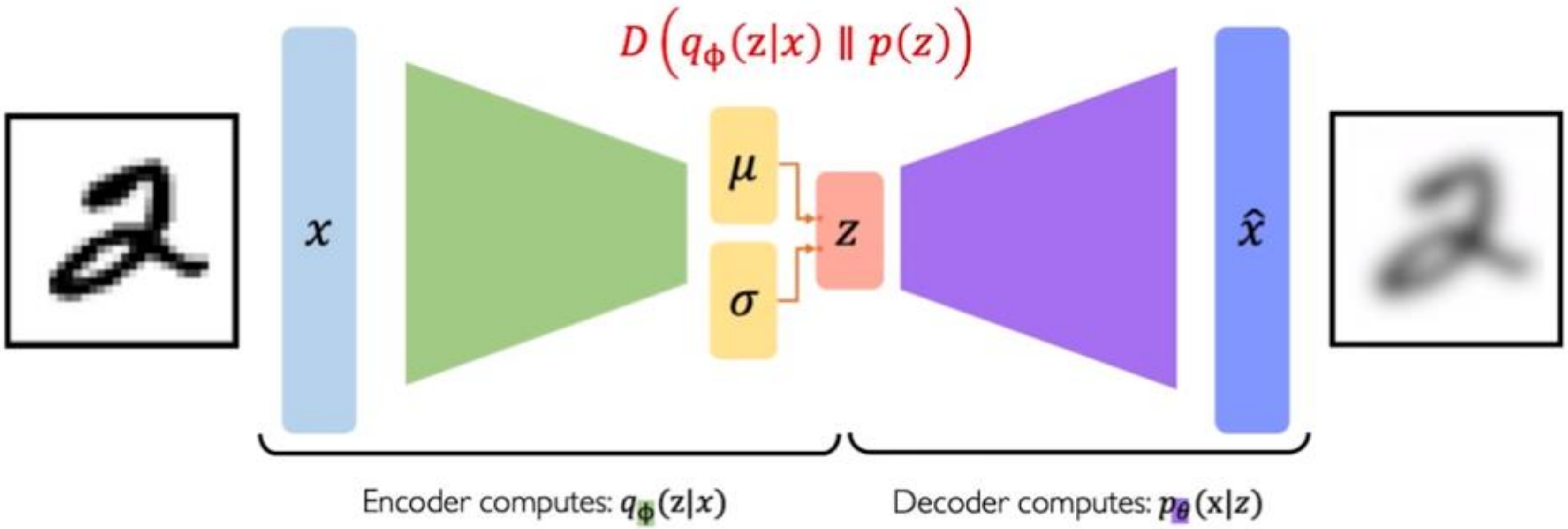
VAE optimization



Encoder computes: $q_{\phi}(z|x)$ Decoder computes: $p_{\theta}(x|z)$

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

VAE optimization



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

Priors on the latent distribution

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

Inferred latent distribution

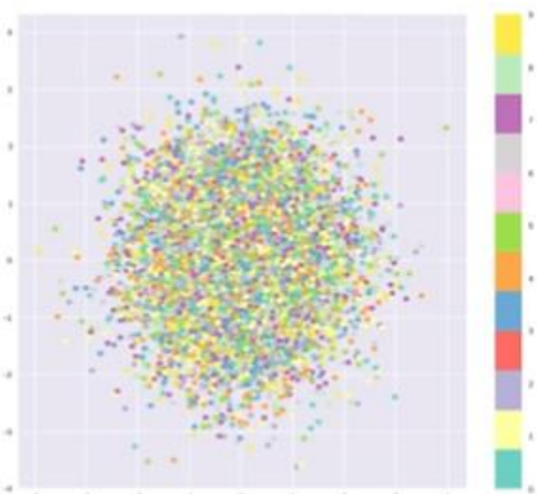
Fixed prior on latent distribution

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:

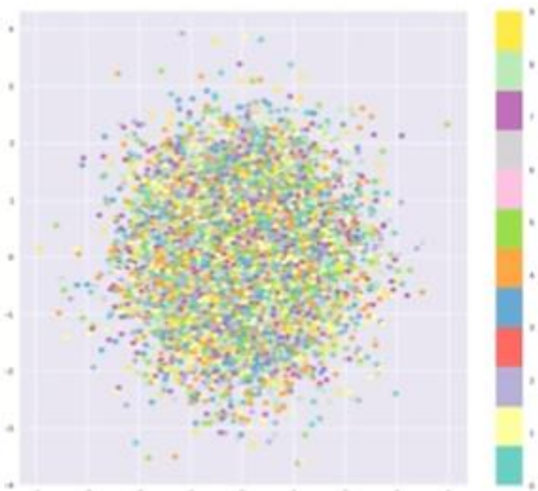
$$p(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

- 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

$$D\left(q_{\phi}(\mathrm{z}|x) \parallel p(z)\right)$$
$$= -\frac{1}{2}\sum_{j=0}^{k-1}\left(\sigma_j + \mu_j^2 - 1 - \log \sigma_j\right)$$

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 the Normal prior

What properties do we want to achieve from regularization? 🤔

1. **Continuity:** points that are close in latent space → similar content after decoding

Intuition on regularization and the Normal prior

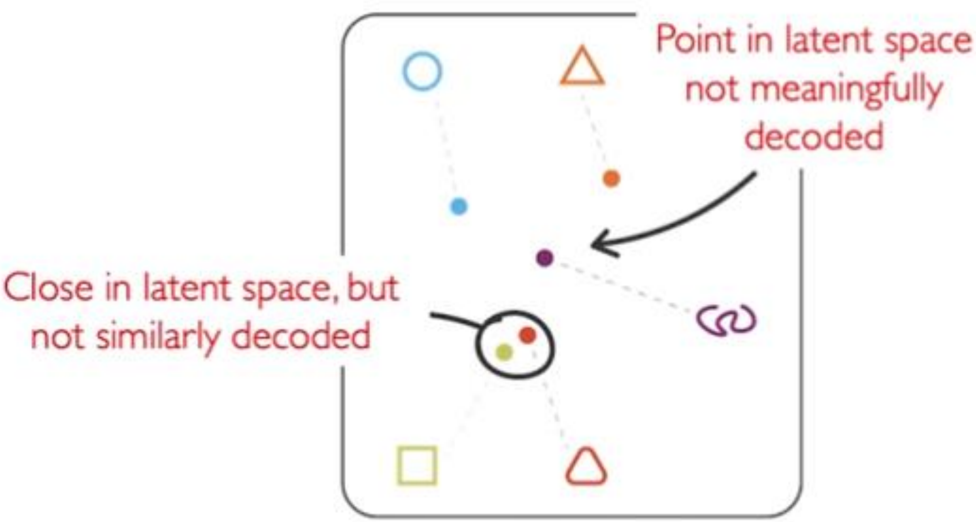
What properties do we want to achieve from regularization? 🤔

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 the 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

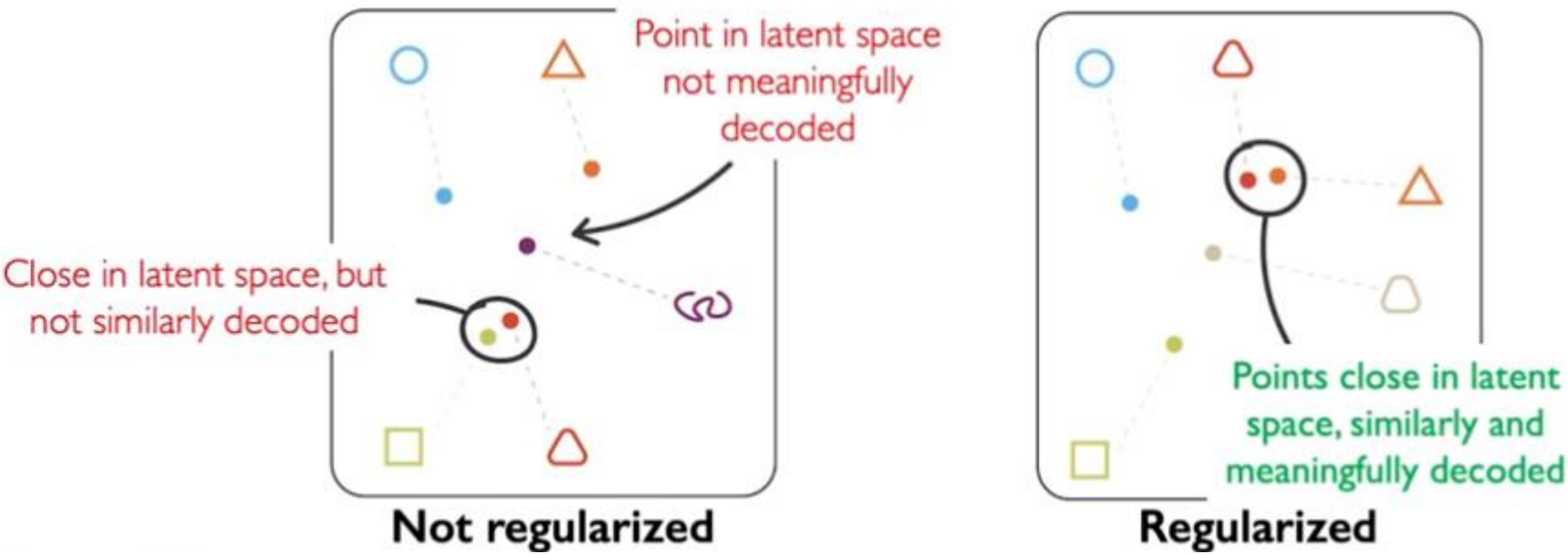


Not regularized

Intuition on regularization and the Normal prior

What properties do we want to achieve from regularization? 🤔

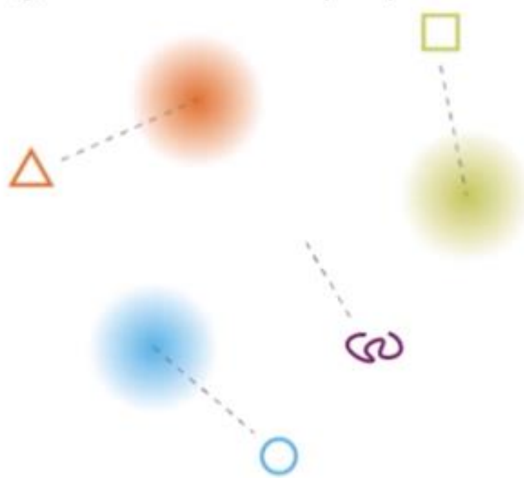
1. **Continuity:** points that are close in latent space → similar content after decoding
2. **Completeness:** sampling from latent space → "meaningful" content after decoding



Intuition on regularization and the 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!



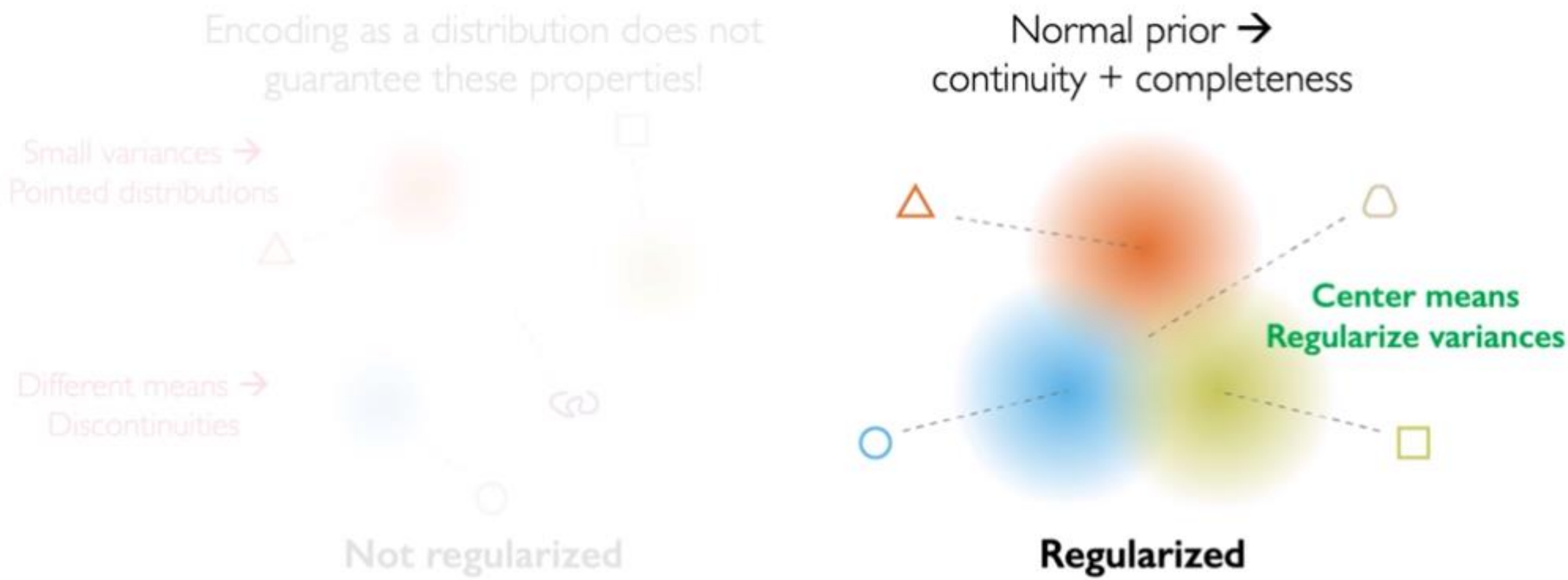
Not regularized

Intuition on regularization and the 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

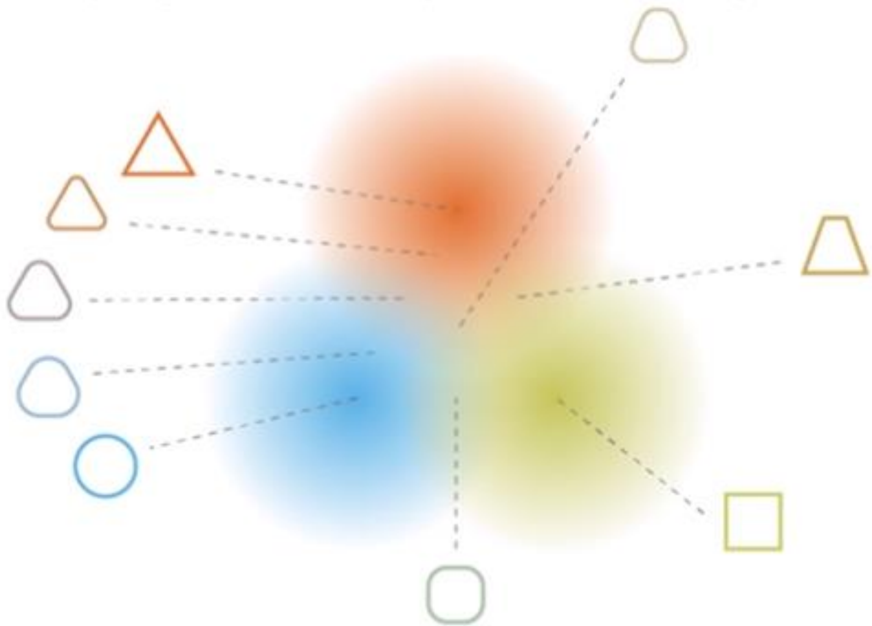
Intuition on regularization and the Normal prior

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



Intuition on regularization and the 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



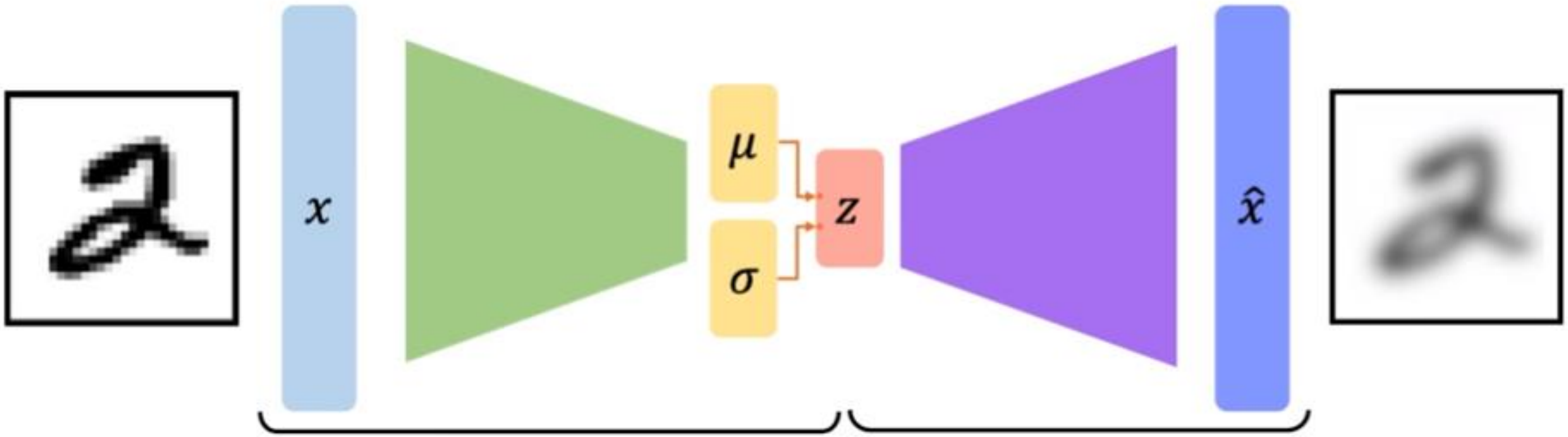
Regularization with Normal prior helps enforce **information gradient** in the latent space.

Tradeoff

- A tradeoff exists!

- The more we regularize
  - The higher the risk of suffering the quality of the reconstruction

VAE computation graph



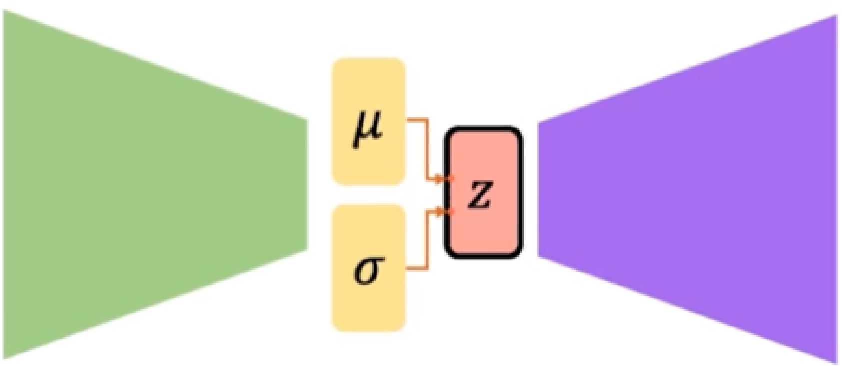
Encoder computes: $q_\phi(\mathrm{z}|x)$ Decoder computes: $p_\theta(\mathrm{x}|z)$

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

Problem

- Not being able to backpropagate through a sampling layer!
- We can only backpropagate using the chain rule if the layers are deterministic.
- In VAEs there is a stochastic layer!

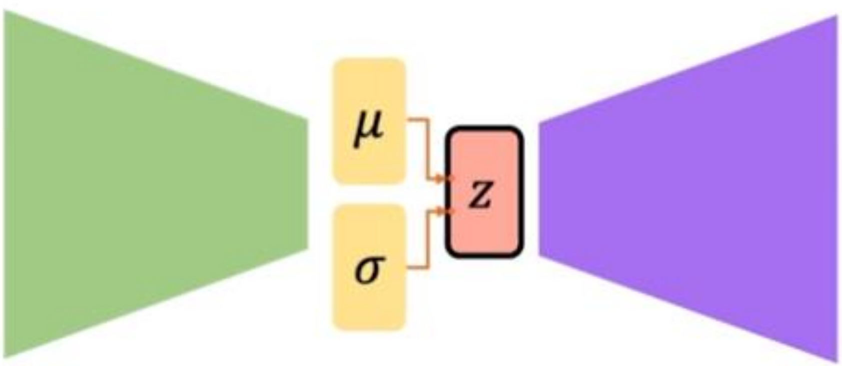
Reparametrizing the sampling layer



Key Idea:

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

Reparametrizing the sampling layer



Key Idea:

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

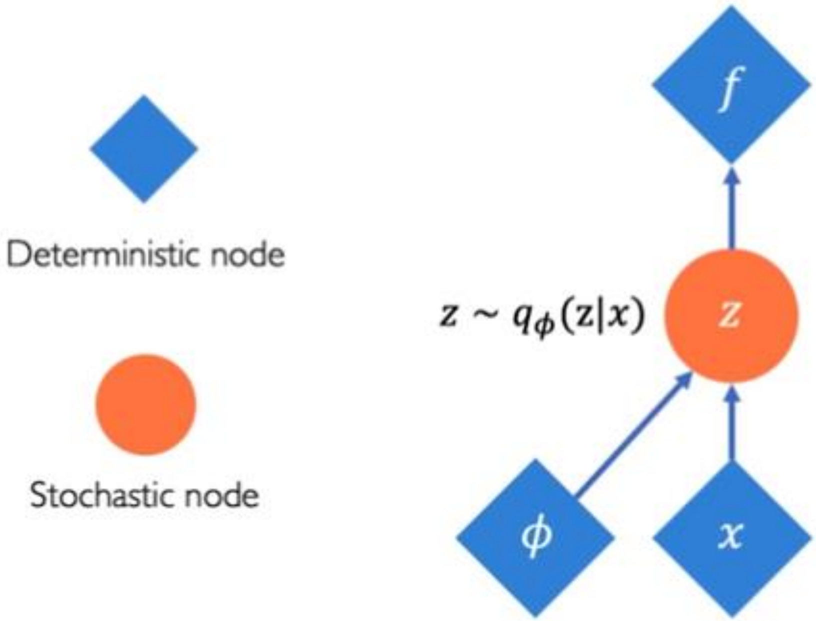
Consider the sampled latent vector $\boldsymbol{z}$ as a sum of

- a fixed $\boldsymbol{\mu}$ vector,
- and fixed $\boldsymbol{\sigma}$ vector, scaled by random constants drawn from the prior distribution

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

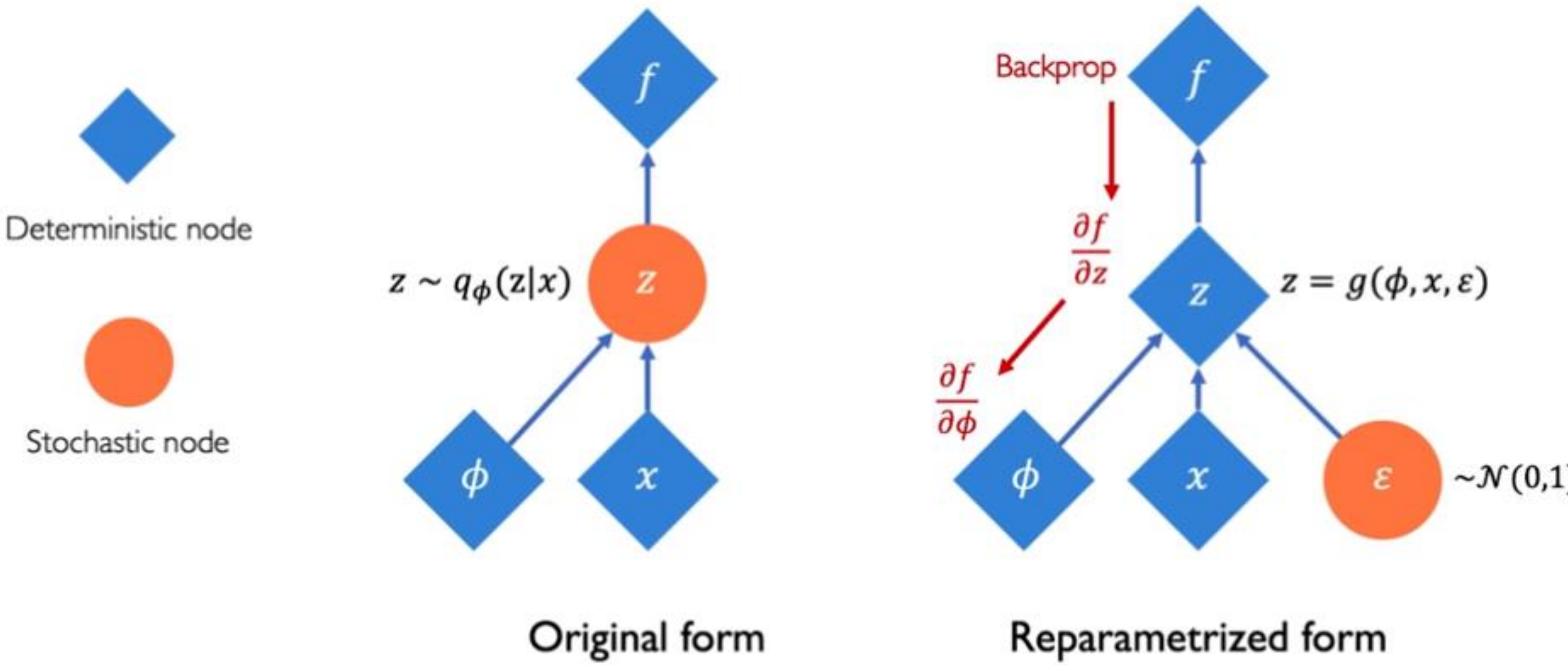
where $\varepsilon \sim \mathcal{N}(0,1)$

Reparametrizing the sampling layer



Original form

Reparametrizing the sampling layer

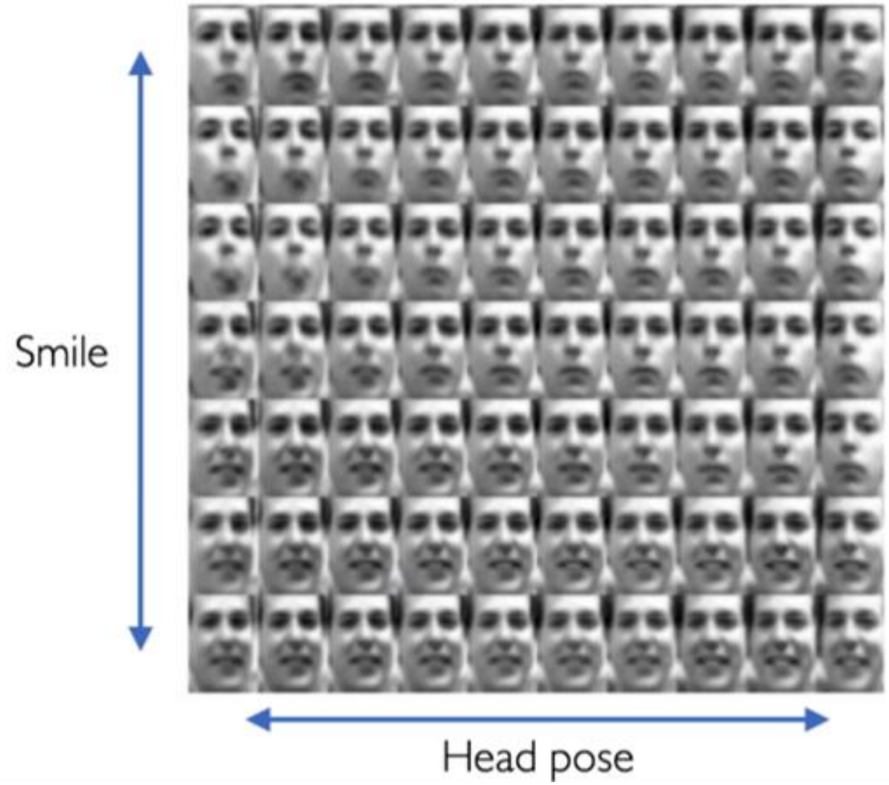


VAEs: Latent perturbation

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

Head pose

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 disentanglement with β-VAEs

β-VAE loss:

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

Latent space disentanglement with β-VAEs

β-VAE loss:

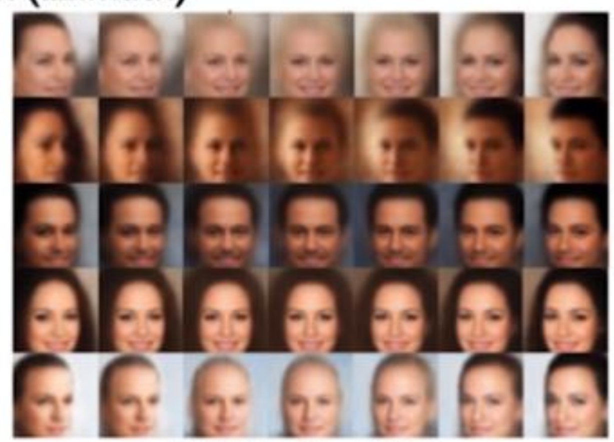
$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\textbf{Reconstruction term}} - \underbrace{\boxed{\beta}\, D_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p(\mathbf{z})\big)}_{\textbf{Regularization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

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

Head rotation (azimuth)

Smile also changing!

Standard VAE (β = 1)

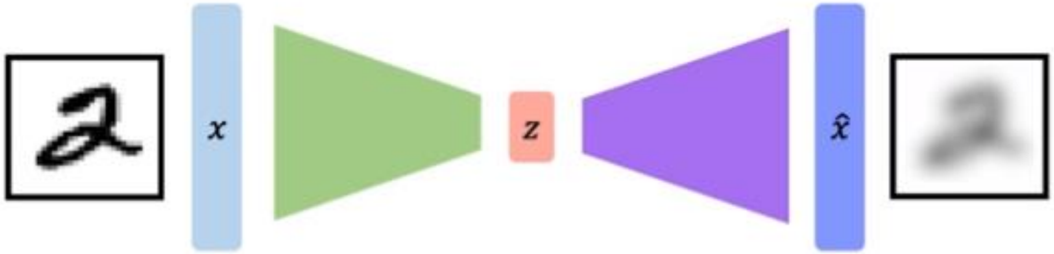


β-VAE (β = 250)

Smile relatively constant!

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. Generating new examples



AEs and VAEs

https://www.youtube.com/watch?v=9zKuYvjFFS8

Thu Jun 17: Free-choice Lecture

- Prof. Brian Kulis

  Associate Professor at ECE, BU
  Amazon Scholar in Alexa

  Audio Lecture

- Andrea Burns

  PhD Student
  IVC Group, CS Dept., BU

  Vision and Language Lecture

- Dr. Mohamed Abdelfattah

  Principal Scientist at Samsung AI Center

  Knowledge Distillation Lecture