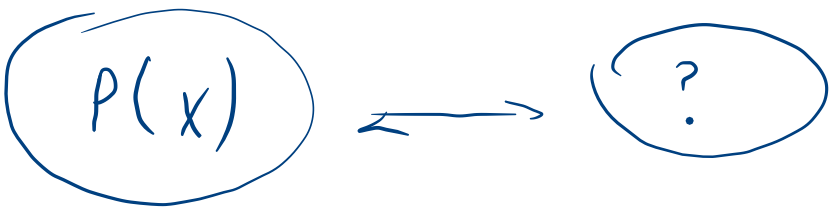
Evaluation

$P(x)$

Diversity

Quality

Speed



Evaluation

- In any research field, evaluation drives progress. How do we evaluate generative models?
- Evaluation of *discriminative* models (e.g., a classifier) is well understood: compare task-specific loss (e.g., top-1 accuracy) on unseen test data
- Evaluating generative models is highly non-trivial.
- **Key question**: What is the task that you care about?
  - Density estimation
  - Compression
  - Sampling/generation
  - Latent representation learning
  - More than one task? Custom downstream task? E.g., Semisupervised learning, image translation, compressive sensing etc. For LLMs: Few shot / zero shot performance through prompting?

Evaluation - Density Estimation or Compression

- Compression is a straightforward for models which have tractable likelihoods

Caveat

Not all models have tractable likelihoods e.g., VAEs, GANs, EBMs.

For VAEs, we can compare evidence lower bounds (ELBO) to log-likelihoods. How about GANs? How to estimate the model likelihood if we only have samples?

In general, unbiased estimation of probability density functions from samples is impossible.

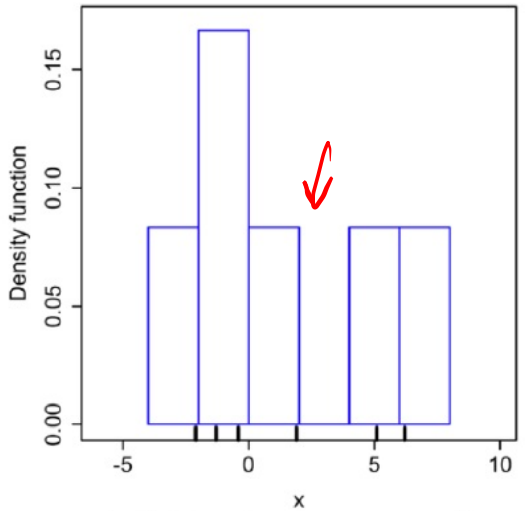
Approximation methods are necessary. We can use kernel density estimates via samples alone.

Kernel Density Estimation

- Given: A model $p_\theta(\mathbf{x})$ with an intractable/ill-defined density
- Let $\mathcal{S} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(6)}\}$ be 6 data points drawn from p_θ.

$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$	$\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$	$\mathbf{x}^{(6)}$
-2.1	-1.3	-0.4	1.9	5.1	6.2

- What is $p_\theta(-0.5)$?
- **Answer 1:** Since $-0.5 \notin \mathcal{S}$, $p_\theta(-0.5) = 0$
- **Answer 2:** Compute a histogram by binning the samples



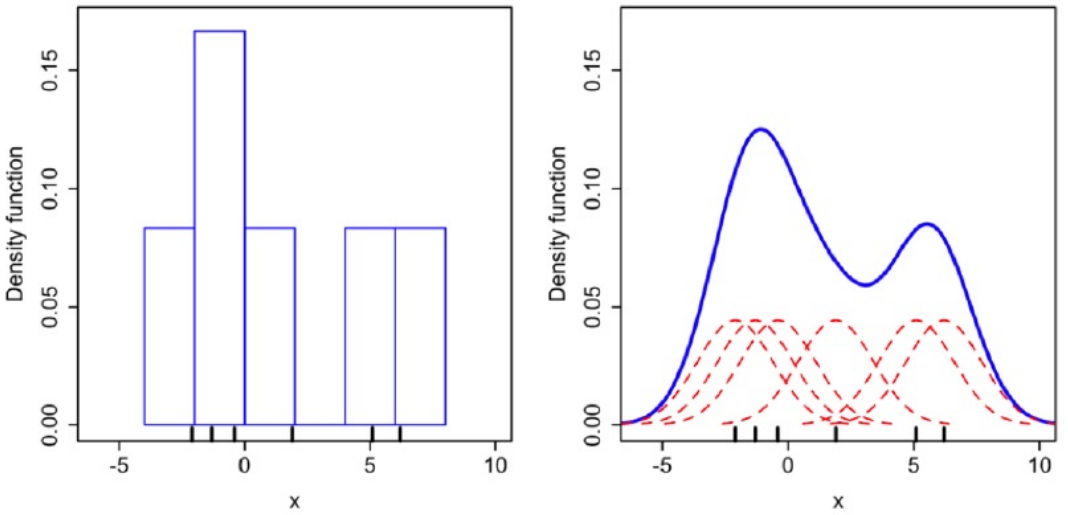
- Bin width= 2, min height= 1/12 (area under histogram should equal 1). What is $p_\theta(-0.5)$? 1/6 $p_\theta(-1.99)$? 1/6 $p_\theta(-2.01)$? 1/12

Kernel Density Estimation (KDE)

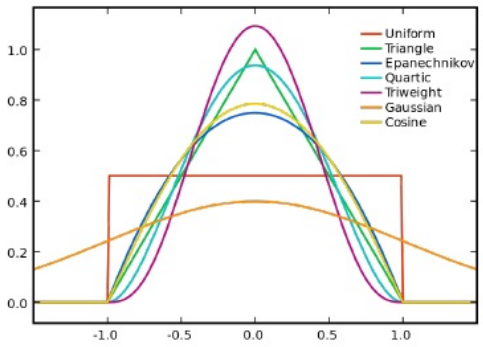
- **Answer 3:** Compute kernel density estimate (KDE) over $\mathcal{S}$

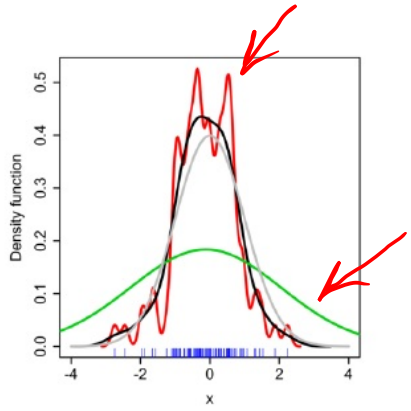
$$\hat{p}(\mathbf{x}) = \frac{1}{n} \sum_{\mathbf{x}^{(i)} \in \mathcal{S}} K\left(\frac{\mathbf{x} - \mathbf{x}^{(i)}}{\sigma}\right)$$

  where σ is called the bandwidth parameter and K is called the kernel function, n is the number of samples in $\mathcal{S}$.
- Example: Gaussian kernel, $K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right)$
- Histogram density estimate vs. KDE estimate with Gaussian kernel



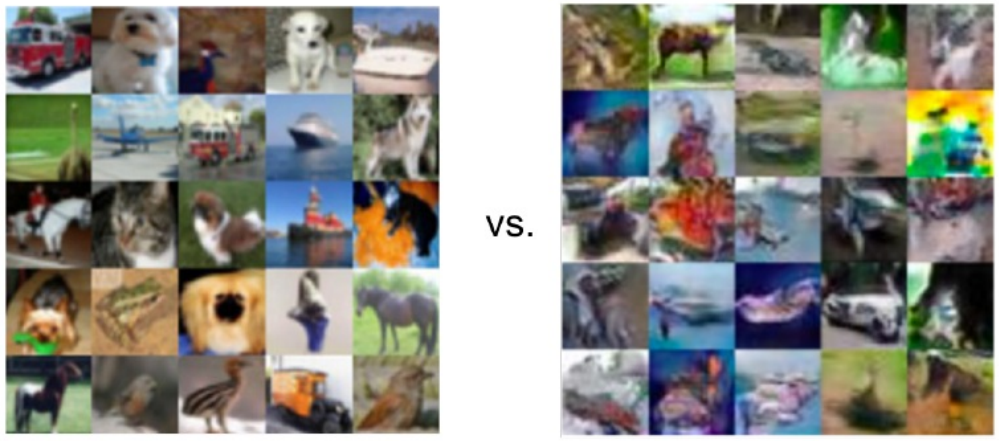
Kernel Density Estimation





- A **kernel** K is any non-negative function satisfying two properties
  - Normalization: $\int_{-\infty}^{\infty} K(u)\mathrm{d}u = 1$ (ensures KDE is also normalized)
  - Symmetric: $K(u) = K(-u)$ for all u
- Intuitively, a kernel is a measure of similarity between pairs of points (function is higher when the difference in points is close to 0)
- **Bandwidth** σ controls the smoothness (see right figure above)
  - Optimal sigma (black) is such that KDE is close to true density (grey)
  - Low sigma (red curve): undersmoothed
  - High sigma (green curve): oversmoothed
  - Tuned via crossvalidation
- **Con**: KDE is very unreliable in higher dimensions

Evaluation - Sample quality



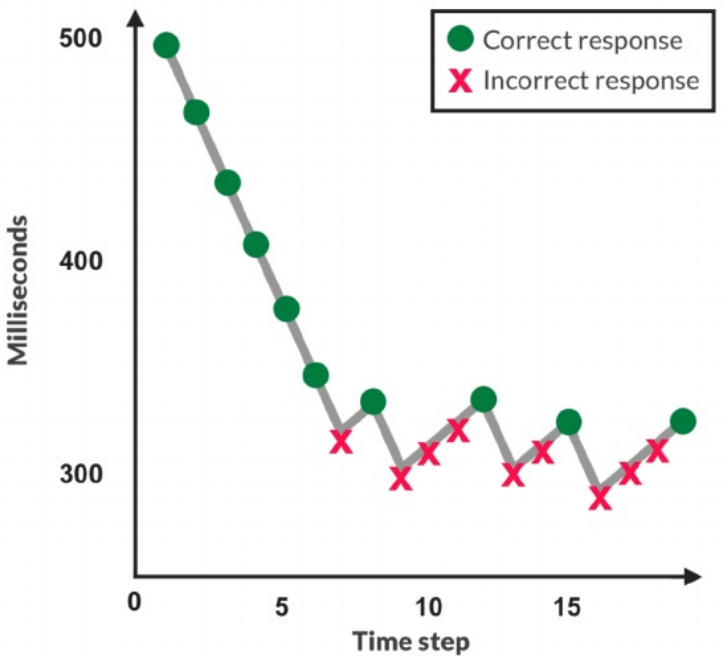
$S_1 = \{\mathbf{x} \sim P\}$ $S_2 = \{\mathbf{x} \sim Q\}$

- Which of these two sets of generated samples "look" better?
- Human evaluations (e.g., Mechanical Turk) are the gold standard.

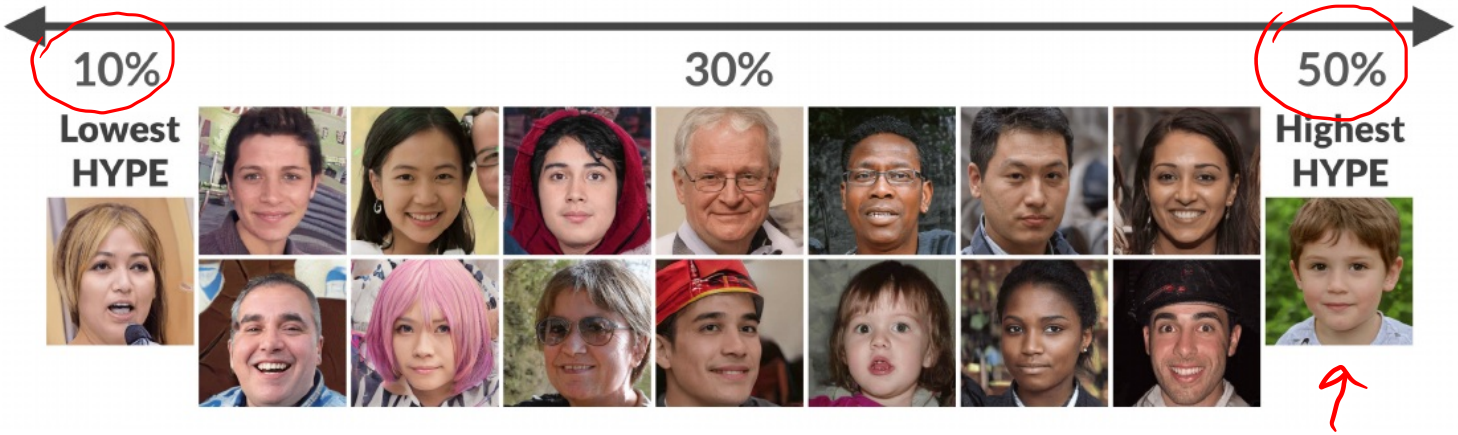
HYPE: Human eYe Perceptual Evaluation (Zhou et al., 2019)

- $\text{HYPE}_{\text{time}}$: the minimum time people needed to make accurate classifications. The larger, the better.
- HYPE_{∞}: The percentage of samples that deceive people under unlimited time. The larger, the better.
- https://stanfordhci.github.io/gen-eval/

Evaluation - Sample quality



The process of determining $\text{HYPE}_{\text{time}}$ scores.



→ HYPE_{∞} scores for samples generated from a StyleGAN.

Evaluation - Sample quality

vs.

$S_1 = \{\mathbf{x} \sim P\}$ $S_2 = \{\mathbf{x} \sim Q\}$

- Which of these two sets of generated samples "look" better?
- Human evaluations (e.g., Mechanical Turk) are expensive, biased, hard to reproduce
- Generalization is hard to define and assess: memorizing the training set would give excellent samples but clearly undesirable
- Quantitative evaluation of a qualitative task can have many answers
- Popular metrics: Inception Scores, Frechet Inception Distance, Kernel Inception Distance

Inception Scores (IS)

- **Assumption 1:** We are evaluating sample quality for generative models trained on labelled datasets
- **Assumption 2:** We have a good probabilistic classifier $c(y|\mathbf{x})$ for predicting the label y for any point $\mathbf{x}$
- We want samples from a good generative model to satisfy two criteria: sharpness and diversity
- **Sharpness (S)**

Low sharpness

High sharpness

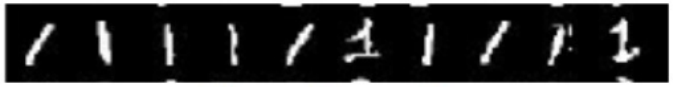
$$S = \exp\left(E_{\mathbf{x}\sim p}\left[\int c(y|\mathbf{x}) \log c(y|\mathbf{x}) \mathrm{d}y\right]\right)$$

$-H(y|X)$ ↓

- High sharpness implies classifier is confident in making predictions for generated images
- That is, classifier's predictive distribution $c(y|\mathbf{x})$ has low entropy

Inception Scores

- **Diversity (D)**



Low diversity

High diversity

$$D = \exp\left(-E_{\mathbf{x}\sim p}\left[\int c(y|\mathbf{x}) \log c(y) \mathrm{d}y\right]\right)$$

where $c(y) = E_{\mathbf{x}\sim p}[c(y|\mathbf{x})]$ is the classifier's marginal predictive distribution

- High diversity implies $c(y)$ has high entropy
- Inception scores (IS) combine the two criteria of sharpness and diversity into a simple metric

$$IS = D \times S$$

- Higher IS corresponds to better quality.
- If classifier is not available, a classifier trained on a large dataset, e.g., Inception Net trained on the ImageNet dataset

$$IS = e^{E\left[KL(c(y|x) \,\|\, c(y))\right]}$$

Inception net

$$KL(c(y|x) \,\|\, c(y)) = \sum_y c(y|x) \log \frac{c(y|x)}{c(y)}$$

$$E_x\left[p(y|x)\right] = p(y)$$

$$= \underbrace{\sum_y c(y|x) \log c(y|x)}_{-H(y|x)} - \sum_y c(y|x) \log c(y)$$

$$E_x\left[KL(c(y|x) \,\|\, c(y))\right] = -H(y|x) - \sum_y \underbrace{E_x\left[c(y|x)\right]}_{c(y)} \log c(y)$$

$$= -H(y|x) - \underbrace{\sum_y c(y) \log c(y)}$$

$$= -H(y|x) + H(y) = I(x;y)$$

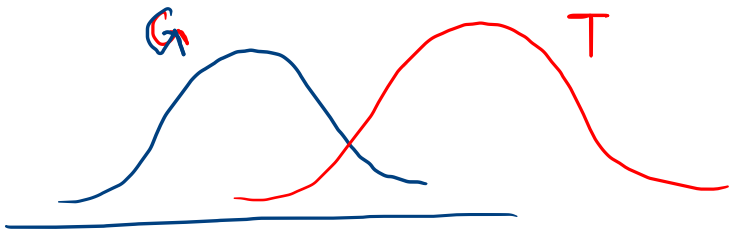
$$IS = \underbrace{e^{-H(y|x)}}_{S\uparrow} \underbrace{e^{H(y)}}_{D\uparrow} = e^{I(x;y)}$$

Frechet Inception Distance

- Inception Scores only require samples from p_θ and do not take into account the desired data distribution p_{data} directly (only implicitly via a classifier)
- **Frechet Inception Distance (FID)** measures similarities in the feature representations (e.g., those learned by a pretrained classifier) for datapoints sampled from p_θ and the test dataset
- Computing FID:
  - Let $\mathcal{G}$ denote the generated samples and $\mathcal{T}$ denote the test dataset
  - Compute feature representations $F_\mathcal{G}$ and $F_\mathcal{T}$ for $\mathcal{G}$ and $\mathcal{T}$ respectively (e.g., prefinal layer of Inception Net)
  - Fit a multivariate Gaussian to each of $F_\mathcal{G}$ and $F_\mathcal{T}$. Let $(\mu_\mathcal{G}, \Sigma_\mathcal{G})$ and $(\mu_\mathcal{T}, \Sigma_\mathcal{T})$ denote the mean and covariances of the two Gaussians
  - FID is defined as the Wasserstein-2 distance between these two Gaussians:

$$\text{FID} = \|\mu_\mathcal{T} - \mu_\mathcal{G}\|^2 + \text{Tr}(\Sigma_\mathcal{T} + \Sigma_\mathcal{G} - 2(\Sigma_\mathcal{T}\Sigma_\mathcal{G})^{1/2})$$

- Lower FID implies better sample quality



FID < 4

$$FID = \|\mu_G - \mu_T\|_2^2 + \|\Sigma_G^{1/2} - \Sigma_T^{1/2}\|_F^2$$

$$\|A\|_F^2 = Tr(A^T A)$$

$$\|x\|^2 = x^T x$$

$$Tr\left((\Sigma_G^{1/2} - \Sigma_T^{1/2})^T (\Sigma_G^{1/2} - \Sigma_T^{1/2})\right)$$

$$= Tr(\Sigma_G) + Tr(\Sigma_T) - 2\,Tr(\Sigma_G^{1/2}\Sigma_T^{1/2})$$

Kernel Inception Distance

- **Maximum Mean Discrepancy (MMD)** is a two-sample test statistic that compares samples from two distributions p and q by computing differences in their moments (mean, variances etc.)
- Key idea: Use a suitable kernel e.g., Gaussian to measure similarity between points

$$MMD(p, q) = E_{\mathbf{x},\mathbf{x}' \sim p}[K(\mathbf{x}, \mathbf{x}')] + E_{\mathbf{x},\mathbf{x}' \sim q}[K(\mathbf{x}, \mathbf{x}')] - 2E_{\mathbf{x} \sim p, \mathbf{x}' \sim q}[K(\mathbf{x}, \mathbf{x}')]$$

- Intuitively, MMD is comparing the "similarity" between samples within p and q individually to the samples from the mixture of p and q
- **Kernel Inception Distance (KID):** compute the MMD in the feature space of a classifier (e.g., Inception Network)
- FID vs. KID
  - FID is biased (can only be positive), KID is unbiased
  - FID can be evaluated in $O(n)$ time, KID evaluation requires $O(n^2)$ time

$$MMD^2(p,q) = \|\mu_p - \mu_q\|_2^2 = \|E[\phi(x)] - E[\phi(y)]\|_2^2$$

$x \sim p$ $\quad x \longrightarrow \phi(x)$

$y \sim q$ $\quad y \longrightarrow \phi(y)$

$$K(x,y) = \phi(x)^T \phi(y)$$

$$\|a\|_2^2 = a^T a$$

$$\|\mu_p - \mu_q\|_2^2 = (\mu_p - \mu_q)^T(\mu_p - \mu_q) = \mu_p^T\mu_p + \mu_q^T\mu_q - 2\mu_p^T\mu_q$$

$$E[K(x,x')] \quad x,x' \sim p \qquad E[K(x,x')] \quad x,x' \sim q \qquad E[K(x,x')] \quad x \sim p,\ x' \sim q$$

$$\mu_p^T\mu_p = E[\phi(x)]^T E[\phi(x')] = E[\phi(x)^T\phi(x')] = E[K(x,x')]$$

$$E[x]\,E[x] = E[x]^2$$

$$\|E[x]\|^2 = \underbrace{E[x]^T E[x]} = E[x^T]\,E[x'] = E[x^T x']$$

$$x \sim p(x)$$

$$x' \sim p(x)$$

$O(n)$ ← FID

$T = \{ \dots \}$ $G = \{ \dots \}$

KID

$T = \{x_1, \dots, x_n\}$ $G = \{y_1, \dots, y_n\}$

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T \rightarrow O(nd^2) \leftarrow$$

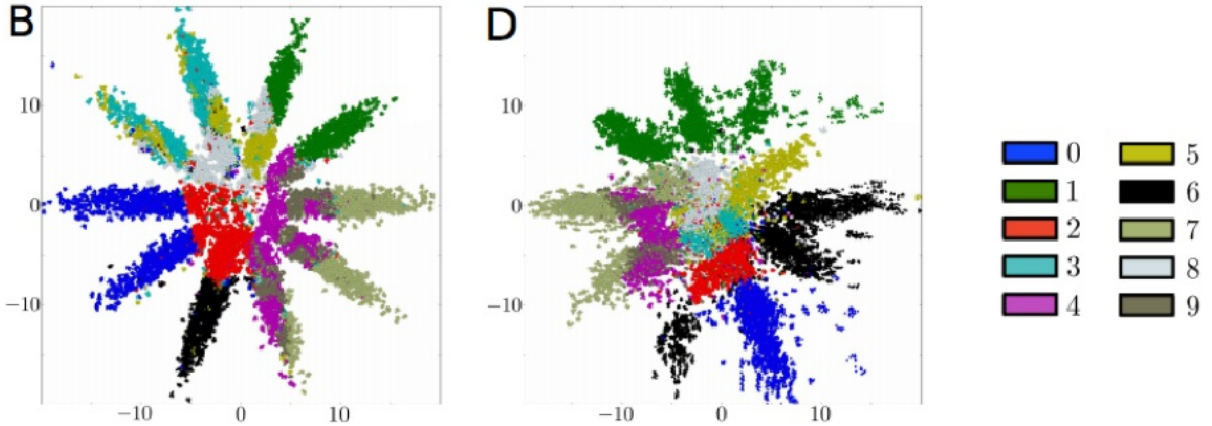
KID: $O(n^2)$ ←

Evaluating latent representations

- What does it mean to learn "good" latent representations?
- For a downstream task, the representations can be evaluated based on the corresponding performance metrics e.g., accuracy for semi-supervised learning, reconstruction quality for denoising
- For unsupervised tasks, there is no one-size-fits-all
- Three commonly used notions for evaluating unsupervised latent representations
  - Clustering
  - Compression
  - Disentanglement

Clustering

- Representations that can group together points based on some semantic attribute are potentially useful (e.g., semi-supervised classification)
- Clusters can be obtained by applying k-means or any other algorithm in the latent space of generative model



Source: Makhzani et al., 2018

- 2D representations learned by two generative models for MNIST digits with colors denoting true labels. Which is better? B or D?

Lossy Compression or Reconstruction

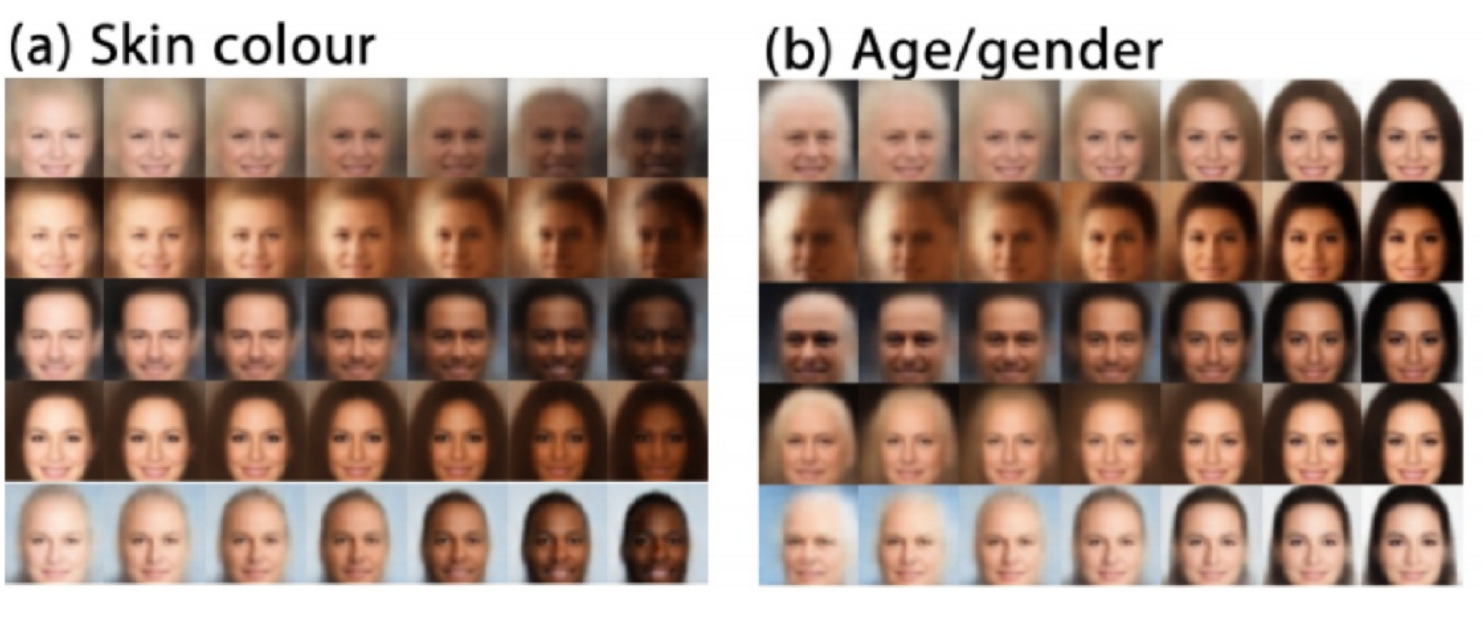
- Latent representations can be evaluated based on the maximum compression they can achieve without significant loss in reconstruction accuracy

Source: Santurkar et al., 2018

- Standard metrics such as Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR), Structure Similarity Index (SSIM)

Disentanglement

- Intuitively, we want representations that disentangle **independent and interpretable** attributes of the observed data



Source: Higgins et al., 2018

- Provide user control over the attributes of the generated data
  - When Z_1 is fixed, size of the generated object never changes
  - When Z_1 is changed, the change is restricted to the size of the generated object

Disentanglement

- Many quantitative evaluation metrics
  - Beta-VAE metric (Higgins et al., 2017): Accuracy of a linear classifier that predicts a fixed factor of variation
  - Many other metrics: Factor-VAE metric, Mutual Information Gap, SAP score, DCI disentanglement, Modularity
  - Check `disentanglement_lib` for implementations of these metrics
- Disentangling generative factors using only unlabeled data is theoretically impossible without additional assumptions