HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

NỀN TẢNG AI TẠO SINH

(IT5410 – Foundation of Generative AI)

ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Đánh giá chất lượng

Bài 10, IT5410 - Nền tảng AI tạo sinh

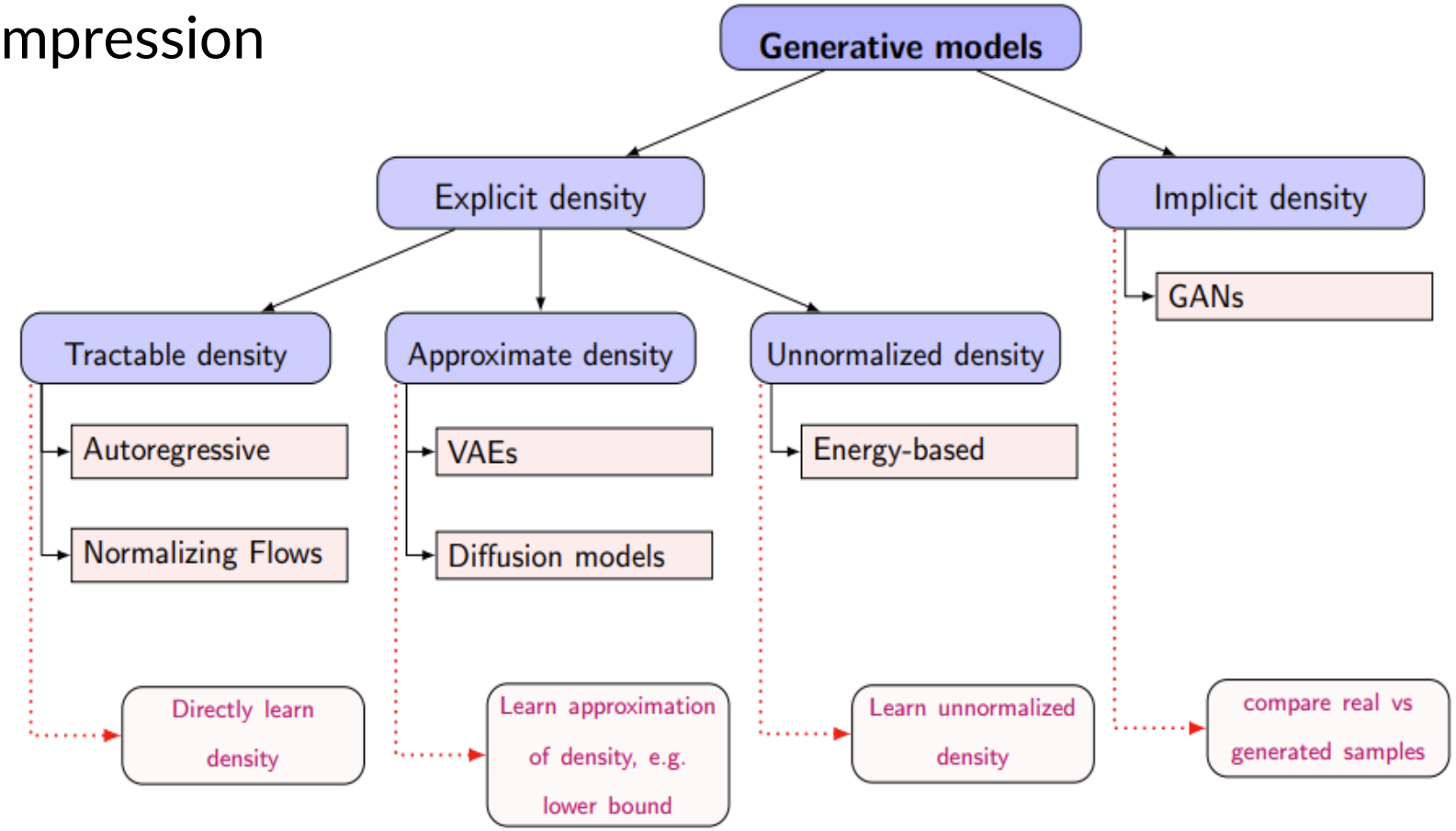
ONE LOVE. ONE FUTURE.

Nội dung

- Mở đầu
- Một số vấn đề của Học sâu
- Một số kiến trúc mạng nơron
- Mô hình sinh sâu
- **Đánh giá chất lượng**
- Học tăng cường

Evaluating Generative Models (Đánh giá chất lượng)

- Model families
- Evaluation criteria
  - Density Estimation or Compression
  - Sample quality
  - Representation learning
  - Task-based evaluation



Evaluation

- 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.**

Discriminative: determine dining room or bedroom

Generative: generate X

Evaluation: Density Estimation

- **Goal**: Approximate the true data distribution $p_{data}(x)$ with a model distribution $p_\theta(x)$.
- *Likelihood* as a metric for density estimation.
- It gives you how well the model generalizes / compresses.
  - split data into training/validation/testing sets,
  - train a model,
  - tune hyperparameters on validation set,
  - evaluate $\mathbb{E}_{x\sim p_{\text{data}}}[\log p_\theta(x)]$ on test set.

Evaluation: Compression

- **Shannon's source coding theorem** links density estimation to data compression.
- To encode a data point x with an optimal code under model $p_\theta(x)$, the length is approximately:

$$L(x) = \log p_\theta(x) \quad \textit{bits}.$$

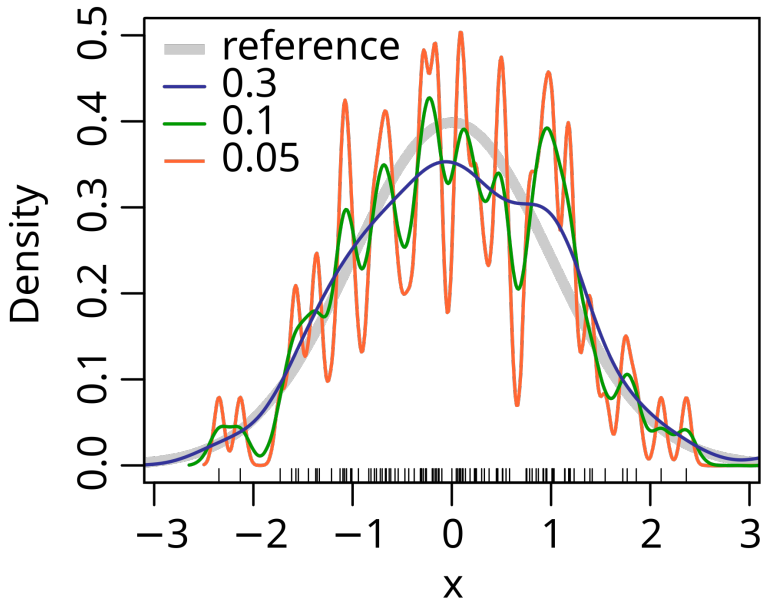
- If the model matches the true distribution, the average code length is close to the entropy of the data.
- Example: A language model with perplexity 50 means on average, each word can be encoded with about $\log 50 \approx 5.64$ bits.

Evaluation: Compression

- **Compression by using Kernel Density Estimation (KDE)**
- KDE is a **non-parametric method** to estimate the probability density function (PDF) of data without assuming a parametric distribution (like Gaussian or exponential).
- Given samples $\{x_1, \dots, x_n\}$, the KDE estimate of the density at a point x is:

$$\hat{p}_h(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right),$$

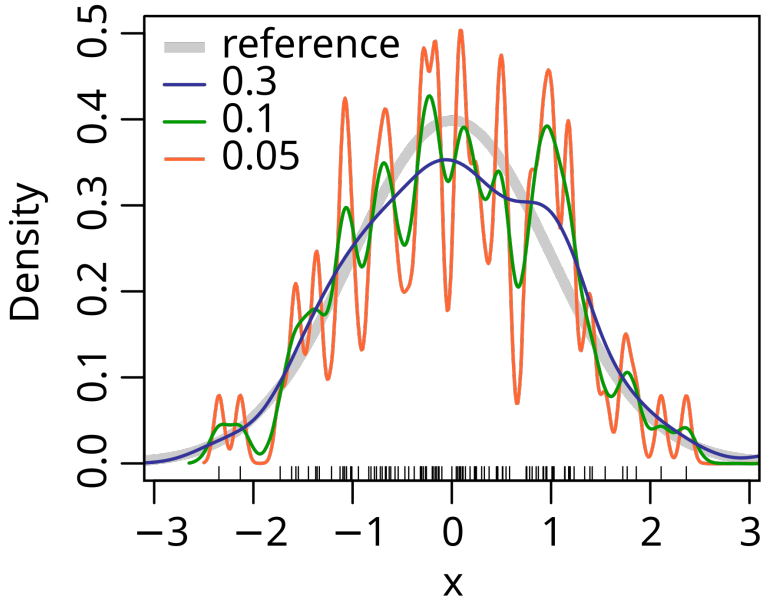
  - n is number of samples,
  - d is data dimensionality,
  - $K(\cdot)$ is kernel function,
  - h (> 0) is bandwidth (smoothing parameter).



Source: wikipedia

Evaluation: Compression

- **Intuition**
  - Imagine you place a **bump (kernel)**, like a Gaussian bell, centered at each data point.
  - Add all bumps together → smoothed approximation of the underlying density.
- The **bandwidth** h controls smoothness:
  - Small h: spiky estimate (overfitting);
  - Large h: overly smooth, missing structure.



Evaluation - Sample quality

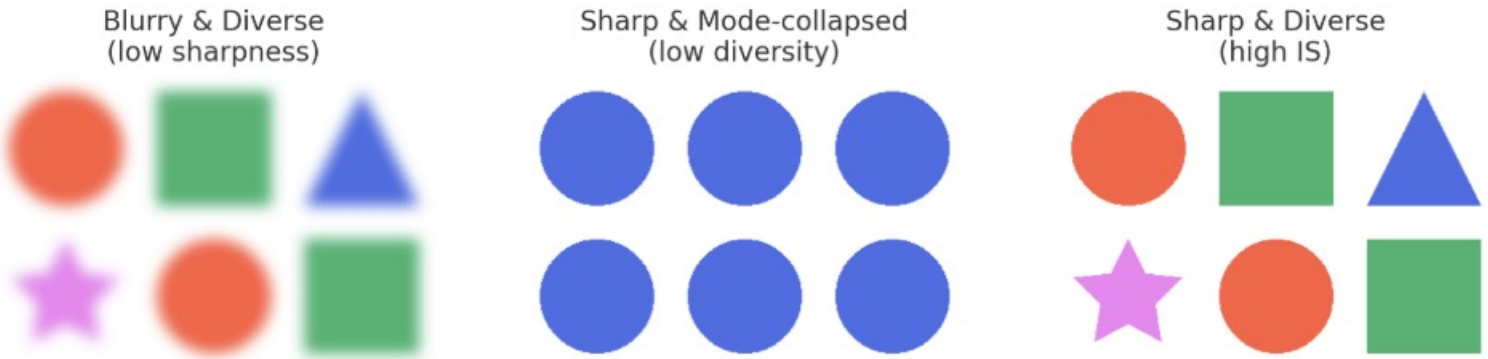
- Compare two sets of generated samples
  - Human evaluations are expensive, biased, hard to reproduce
  - Generalization is hard to define and assess: memorizing the training set would give excellent samples but clearly undesirable
- Quantitative evaluation of a qualitative task can have many answers
- Popular metrics:
  - Inception Scores (IS)
  - Frechet Inception Distance (FID)
  - Kernel Inception Distance (KID)

Evaluation - Sample quality

- 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
- Source: https://stanfordhci.github.io/gen-eval/
- ***Gold standard: ask humans if samples look real or not.***
- Examples:
  - Visual Turing Test: humans guess if an image is real or generated.
  - HYPE benchmark: controlled large-scale human perceptual evaluation.
- Advantages: measures true perceptual quality.
- Disadvantages: expensive, slow, not reproducible, limited scalability.

Evaluation: Inception Scores (IS)

- A good probabilistic classifier c(y|x) for predicting the label y for any point x
- A good generative model to satisfy two criteria: sharpness and diversity
- Illustrative image showing three cases that affect Inception Score:



- **Blurry & Diverse (low sharpness)**: varied objects but blurred (classifier uncertain → low p(y|x))
- **Sharp & Mode-collapsed (low diversity)**: high-quality images but all same class (low marginal entropy p(y))
- **Sharp & Diverse (high IS)**: high-quality images across many classes

Evaluation: Inception Scores (IS)

- Sharpness (S)

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

- High sharpness implies classifier is confident in making predictions for generated images so that classifier's predictive distribution c(y|x) has low entropy

- *Source*: Stefano Ermon (AI Lab)

Evaluation: Inception Scores (IS)

- Diversity (D)

  - 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×S
  - Higher IS corresponds to better quality.
- If classifier is not available, train a classifier on a large dataset. e.g., InceptionNet trained on the ImageNet dataset

Evaluation: Fréchet Inception Distance (FID)

- Fréchet distance is to compare two distributions
- For two Gaussians $\mathrm{p}_1 = \mathcal{N}(\mu_1, \Sigma_1)$, $\mathrm{p}_2 = \mathcal{N}(\mu_2, \Sigma_2)$, the distance is $d(p_1, p_2)$

$$d(p_1, p_2)^2 = \|\mu_1 - \mu_2\|^2 - tr(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{0.5})$$

- In practice: compare two sets of samples
  - Set of real samples from p_{data}
  - Set of generated samples from p_θ
- How to compute the Fréchet distance of those distributions?
  - They are not Gaussian ➔ Intractable to compute
- Make them Gaussian?

Evaluation: FID

- FID:
  - **Input**:
    - a function $f: X \longrightarrow \mathbb{R}^n$
    - Two datasets $S_1, S_2 \subseteq X$
  - **Compute** $f(S_1), f(S_2)$, the new representations of the samples
  - **Fit** Gaussian $\mathcal{N}(\mu_1, \Sigma_1)$ from $f(S_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$ from $f(S_2)$
  - **Return** $d(p_1, p_2)^2 = \|\mu_1 - \mu_2\|^2 - tr(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{0.5})$
- Where does f come from?
- For images:
  - f is the **Inception v3** model trained on the ImageNet, but without its final classification layer
  - *(we implicitly assume that **Inception v3** is good enough to map from a data distribution to a Gaussian)*

Evaluation: Maximum Mean Discrepancy (MMD)

- We want to test whether **two probability distributions** are the same:
  - Real data distribution $p(x)$.
  - Generated data distribution $q(x)$.
- MMD compares the two distributions via their embeddings in a **Reproducing Kernel Hilbert Space (RKHS)**.
- **MMD can be used for Generative Models**
  - **Training**: MMD-GANs use MMD as a loss function instead of adversarial training.
  - **Evaluation**: MMD is used to check whether generated data matches real data distribution.

Evaluation: MMD

Maximum Mean Discrepancy (MMD)

Given two distributions $p$and q, the **MMD** is defined as:

$$\mathrm{MMD}[\mathcal{F}, p, q] = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{x \sim p}[f(x)] - \mathbb{E}_{y \sim q}[f(y)]\right)$$

where:

- $\mathcal{F}$ is a class of functions induced by a kernel $k(\cdot,\cdot)$.
- The idea: If $p = q$, expectations under p and q should be equal for all f.

By using RKHS properties, this supremum has a **closed form**.

Key idea: Use a suitable kernel e.g., Gaussian to measure similarity between points

Kernel Trick Formulation

With kernel $k(x, y)$, we can write:

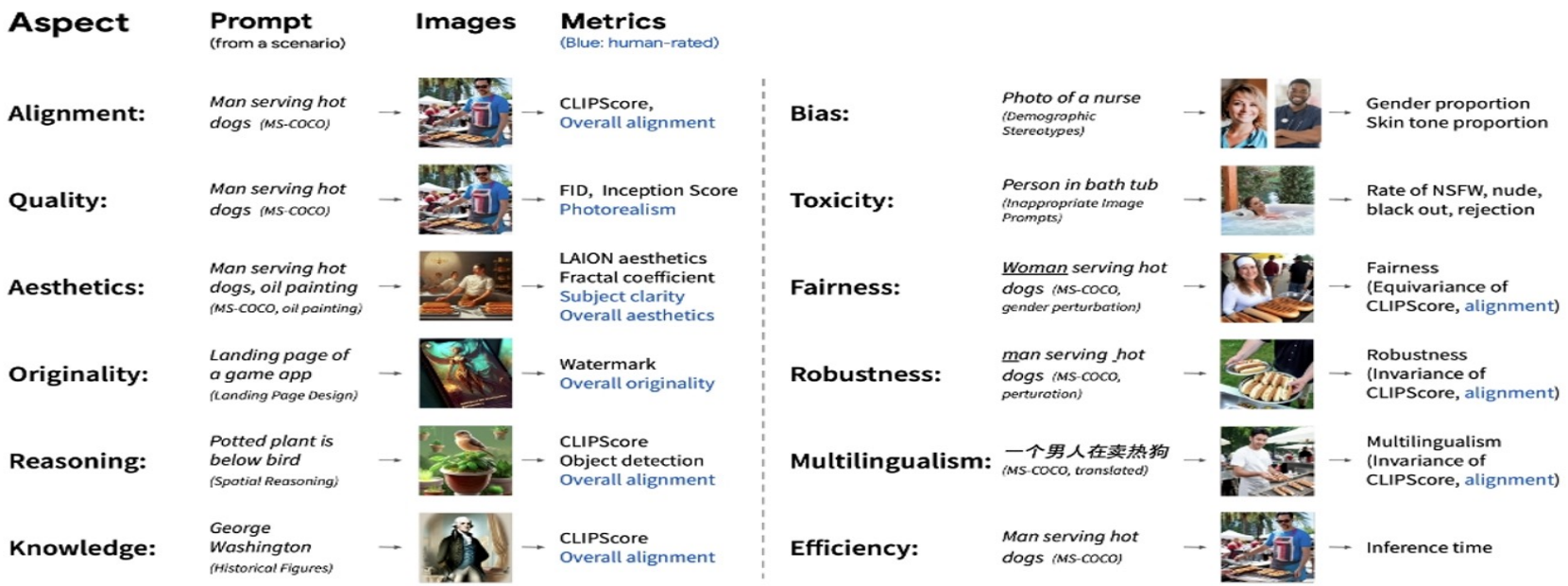
$$\mathrm{MMD}^2(p, q) = \mathbb{E}_{x,x' \sim p}[k(x, x')] + \mathbb{E}_{y,y' \sim q}[k(y, y')] - 2\mathbb{E}_{x \sim p, y \sim q}[k(x, y)]$$

- If $p = q$, then MMD = 0.
- If $p \neq q$, MMD > 0 (the bigger, the more different).
- Choice of kernel (e.g., Gaussian RBF) determines sensitivity.

Intuitively, MMD is comparing the "similarity" between samples within p and q individually to the samples from the mixture of p and q

Evaluation: Kernel Inception Distance (KID)

- Compute the MMD in the feature space of a classifier (e.g., InceptionNet)
- 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



Source: Stefano Ermon (AI Lab)

Evaluation: Kernel Inception Distance (KID)

- Many metrics to consider when evaluating text2image models: quality (FID, Inception, KID, etc), alignment with provided caption (CLIP score), biases, etc.
- HEIM: Holistic Evaluation of Text2Image models 26 models, 29 scenarios, 33 metrics (automated and human) https://crfm.stanford.edu/heim/latest/
- Frechet Inception Distance (FID):
  - Compares real vs generated distributions in feature space.
  - Lower = better.
  - Limitation: no text grounding, only image realism.
- Kernel Inception Distance (KID), MMD-based metrics:
  - Similar to FID but unbiased.

Evaluation - Latent representations

Evaluating latent representations

When we train a generative model (GAN, Diffusion, etc.), the **latent space** encodes compressed information about the data. Evaluating how good this latent space is matters because it determines:

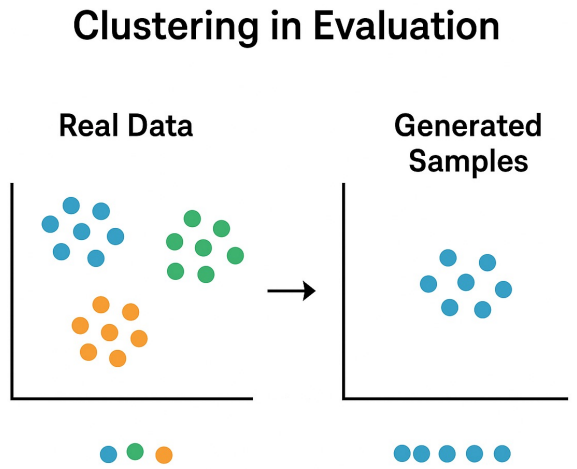
- **Quality of generation** (sample good outputs).
- **Structure & disentanglement** (do latent dimensions).
- **Downstream usability** (latents help classification, clustering, retrieval).

- 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

Evaluation - Latent representations

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
- Note labels are only used for evaluation, not obtaining clusters itself (i.e., clustering is unsupervised)



Evaluation - Latent representations

Clustering

- from sklearn.metrics.cluster import completeness homogeneity score, v measure score.
- **Completeness score (between [0, 1]):** maximized when all the data points that are members of a given class are elements of the same cluster completeness score(labels true=[0, 0, 1, 1], labels pred=[0, 1, 0, 1]) %0
- **Homogeneity score (between [0, 1]):** maximized when all of its clusters contain only data points which are members of a single class homogeneity score(labels measure true=[0, 0, 1, 1], labels pred=[1, 1, 0, 0]) %1
- **V measure score** (also called normalized mutual information, between [0, 1]): harmonic mean of completeness and homogeneity score v score(labels 0, 0])
- true=[0, 0, 1, 1], labels Deep Generative Models pred=[1, 1,0, 0] %1

Evaluation - Latent representations

Lossy Compression or Reconstruction

In generative modeling, **lossy compression** is closely tied to **reconstruction**:

- Input data $x \rightarrow$ encoded into latent code z.
- Decoder reconstructs $\hat{x}$ from z.
- The difference between x and $\hat{x}$ measures **reconstruction quality**.

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

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

Evaluation - Latent representations

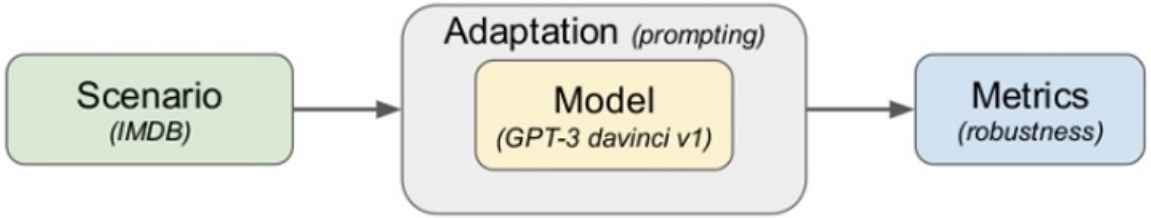
Disentanglement

- A disentangled representation means that individual latent dimensions correspond to independent and interpretable factors of variation in the data.
- Example: in face generation, one latent variable might control pose, another lighting, another hair color.
- Disentanglement is especially important in VAEs, β-VAEs, InfoGANs, FactorVAEs, etc.

- 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

Evaluation - Latent representations

Solving tasks through prompting



- A language model is a generative model of text, e.g. p_θ(next word | sentence)
- A language model can be used to directly solve tasks without extracting representations by specifying tasks in natural language
- For example, in sentiment classification, given a text (e.g., movie review), the goal is to p
- Use p_θ(next word | sentence) to predict the next word ( ). Is it ”Positive” or ”Negative”? Many prompting strategies are possible (Prompt Engineering)
- Adaptation by finetuning the model is also widely used to predict the sentiment (positive or negative).

Summary

- Quantitative evaluation of generative models is a challenging task. These methods calculate some numerical scores based on some criteria.
- Qualitative methods: These methods inspect the generated data visually or auditorily.
- For downstream applications, one can rely on application-specific metrics .
- For unsupervised evaluation, metrics can significantly vary based on end goal: density estimation, sampling, latent representations

References

- Cheema, Fasil and Ruth Urner (2023). “Precision Recall Cover: A Method For Assessing Generative Models”. In *International Conference on Artificial Intelligence and Statistics*, pp. 6571–6594.
- Murphy, Kevin P. (2023). Probabilistic Machine Learning: Advanced Topics. The MIT Press. Sajjadi, Mehdi S. M. et al. (2018). “Assessing Generative Models via Precision and Recall”. In *Advances in Neural Information Processing Systems*, pp. 5234–5243.
- Salimans, Tim et al. (2016). “Improved Techniques for Training GANs”. In *Advances in Neural Information Processing Systems*, pp. 2226–2234.
- Thanh-Tung, Hoang and Truyen Tran (2020). “Toward a Generalization Metric for Deep Generative Models”. In: arXiv abs/2011.00754

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !