CENG796 - Topic Summary:
Evaluation Metrics For Generative Models

Meriç Karadayı - 2448553, İbrahim Ersel Yigit - 2449072

April 2024

Contents

1 Introduction

Evaluation drives the progress of generative models, as it does in any research field. So we should evaluate the generative models properly. But it is not a trivial task. The technique we should use to evaluate the model highly depends on what we care.

While developing generative models, in different task we may care about obtaining a good density estimation, good sampling, or a good latent representation learning. Some task even may require multiple of them. In this document, we will explain the methods and elaborate them for each, respectively. Furthermore, we will mention evaluation methods for text domain generative models.

2 Density Estimation Metrics

Evaluating the density estimation of generative models involves assessing how well these models can estimate the probability distribution of the data they are trained on. It may require different approach for different architectures.

2.1 Models with tractable likelihoods

Some of generative model architectures like auto-regressive models or gaussian mixture models, have tractable likelihood, which makes density estimation straightforward. It can be done by following steps:

1. First, split the dataset into train, validation, and test sets
2. Evaluate gradients based on train set
3. Tune hyperparameters by using validation set
4. Compute the log-likelihood of the test data under the model. (In language models, perplexity may be used instead of log-likelihood)

2.2 Kernel Density Estimation

Unfortunately, not all generative models have tractable likelihood. Therefore, we need a more general method to evaluate the density estimation. Kernel Density Estimation provides us this.

Kernel density estimation is a technique for estimation of probability density function that is a must-have enabling the user to better analyse the studied probability distribution than when using a traditional histogram, as it explained in [1]. Intuitively, a kernel is measure of similarity between pairs of points. Which means the kernel function should return higher when two points are closer to each other.

Computing the kernel density estimation over S can be done by following:

$$\widehat{p}(x) = \tfrac{1}{n} \textstyle\sum_{x^(i) \in S} K(\tfrac{x-x^{(i)}}{\sigma}) \text{ where;}$$

K is the kernel function and σ is the bandwidth

The Kernel Function need to be a non-negative function that satisfy the following properties:

- Normalization: $\int_{-\infty}^{\infty} K(u)du = 1$
- Symmetric: $K(u) = K(-u)$ for all u

The badtwidth σ is a hyperparameter that controls the smoothness that brings more smoother kernel function with higher value. This parameter may and should be tuned with cross-validation.

It should be noted that Kernel Density Estimation method gets more unreliable as the number of dimension gets higher.

2.3 Importance Sampling

Importance sampling is introduced in [2] and it is a Monte Carlo method used to evaluate properties of one distribution by using samples generated from a different distribution. Importance sampling has plenty of application related with statistics and it can also be used while evaluating the density estimation. Which also called Annealed Importance Sampling (AIS) [3]

Annealed Importance Sampling can be perform by following the steps;

1. Initialize: Start with samples from an easy-to-sample initial distribution
2. Annealing Schedule: Define a sequence of intermediate distributions that progressively transition from the initial distribution to the target distribution.
3. Importance Weights: Calculate weights for each sample based on the ratios of probabilities between consecutive distributions.
4. Combining Weights: Accumulate the weights across all transitions to estimate the overall importance weight for each sample.

Note that this method provides unbiased estimates of likelihoods but biased estimates of log-likelihood.

2.4 Laplace Approximation

In the above sections we have mentioned that we cannot straightforwardly estimate the density

$$p_X(x) = \int p_{x|z}(x|z)p_z(z)dz \tag{1}$$

when we do not know the $p_Z(z)$ which is the case for many generative model architectures like VAEs and GANs.

[4] proposes that even though we cannot estimate the (1), we still can approximate it by Laplace's Method.

$$p_X(x) \approx (\frac{1}{\sqrt{2\pi}})^n \sigma^{-n} \sqrt{det(\Sigma)} e^{-\frac{c(x)}{2}}$$

2.5 Parzen Window Density Estimation

When the tractable likelihoods of the model are not available, another common alternative is Parzen window estimation [5]. In this method we do the followings in order;

1. Generate samples from the model
2. Use the samples to construct a tractable model (typically a kernel density estimator with Gaussian kernel)
3. Evaluate likelihoods under this tractable model
4. Used them as proxy for the true model likelihoods

It need to be always considered while using Parzen window estimation, the method generally could not brings likelihoods similar to likelihoods of the true model, when the number of data dimension gets higher, even the large number of sample generated.

3 Sampling/Generation Metrics

Evaluating the quality of sampling or generation can be basically considered as the generated samples how looks good. The diversity of the generated samples should be keep in mind in order to separate the memorizing the dataset and a good generation which is not a trivial task.

Quantitative evaluation of qualitative task may have different answer with different methods, in this document we will explain some of popular methods and metrics.

3.1 Inception Score (IS)

Inception Score is used to evaluate both image quality and output diversity of the model. While calculating the Inception Score we need to make following 2 assumptions;

- We are evaluating sample quality for generative models that trained on labelled datasets.
- We have a good probabilistic classifier $c(y|x)$ where y is the predicted label and x is the data point. (Typically pre-trained inception classifier is used as $c(y|x)$)

Inception Score method declares that samples from a good generative model should satisfy two criteria; Sharpness and Diversity.

3.1.1 Sharpness

The sharpness score S is computed as:

$$S = exp(E_{x\sim p}[\int c(y|x) log c(y|x) dy])$$

and high sharpness score implies better image quality.

3.1.2 Diversity

The diversity score D is computed as:

$$D = exp(-E_{x\sim p}[\int c(y|x) log c(y) dy])$$

and high diversity score implies better generalization.

Inception Score combines the sharpness score and diversity score, and calculated as following; $IS = S \times D$. Therefore, obviously higher Inception Score (IS) indicates the model has better generation and sampling.

Inception Score can also be interpreted as;

$$IS = exp(\mathbb{E}_{x\sim p_g} D_{KL}(p(y|x)||p(y)))$$

which indicates, exponential of the Kullback-Leibler (KL) divergence between the conditional label distribution (given a data point) and the marginal label distribution (overall distribution across all generated samples)

3.2 Frechet Inception Distance (FID)

Inception Score considers image quality and diversity of generated samples, but it does not explicitly consider the training data distribution. On the other hand, Frechet Inception Distance measures the similarity between generated data distribution and the real data distribution, by using their feature representation.

In order to calculate FID, lets denote G the generated sample distribution, T the real data distribution, and F a feature representation extractor (typically prefinal layer of Inception Net)

Then follow these steps in order;

- Compute F_G and F_T which represents the feature representation of G and T respectively.
- Fit two multivariate Gaussian distribution for F_G and F_T. Lets denote them (μ_G, Σ_G) and (μ_T, Σ_T) respectively. Note that μ represents mean and Σ represents covariance of corresponding multivariate Gaussian distribution.
- Finally compute the FID score as:

  $$FID = \|\mu_T - \mu_G\|^2 - Tr(\Sigma_T + \Sigma_G - 2(\Sigma_T \Sigma_G)^{1/2})$$

  where; $Tr(M)$ means Trace of matrix M

Even though it can be inferred by the formula, it should be noted that, lower Frechet Inception Distance (FID) indicates a model with better sampling/generation.

3.3 Maximum Mean Discrepancy (MMD)

In general, Maximum Mean Discrepancy is a statistic that describe difference between two distributions p and q by using their moments which obtained by a kernel [6]. For example, obtaining moments mean and variance via Gaussian as kernel.

Maximum Mean Discrepancy (MMD) between distributions p and q is calculated as;

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

where; K is stands for the kernel.

Note that, $MMD(p, q)$ measures the similarity between distributions p and q, and lower MMD values indicates closer distributions p and q. Reasonably it is equal to 0 if and only if $p = q$ [6].

More specifically, in order to use Maximum Mean Discrepancy for evaluation of sampling/generation quality of a model, we simply choose p as the real data distribution and q as generated data distribution. In that way, we basically measure the similarity of real data distribution and generated data distribution like we did while computing FID.

3.4 Kernel Inception Distance (KID)

Kernel Inception Distance takes MMD one step forward. KID is calculated by again computing the Maximum Mean Discrepancy, but instead of using their moment, now we compute the MMD in the feature space of a classifier (typically a neural network like Inception).

3.4.1 FID vs KID

There is a trade-off between using FID or KID for evaluating generative models. KID has two main advantage over FID;

- Unlike the FID, the KID has a simple unbiased estimator [8].
- While FID fits a Gaussian distribution, KID does not require any specific distribution [8].

On the other hand, FID has a computational advantage over KID. While FID can be computed in $O(N)$, KID evaluation requires $O(N^2)$.

3.5 Feature Likelihood Divergence (FLD)

Even though, all IS, FID, KID metrics that are explained above are commonly used to evaluation of sampling quality of generative models, recent research [9] states that they have considerable limitations such as;

- being insensitive to over-fitting
- could not generalizing beyond the training dataset.

In order to overcome such issues a new metric called Feature Likelihood Divergence (FLD) is proposed. a novel sample-based metric that captures sample fidelity, diversity, and novelty. FLD enjoys the same scalability as popular sample-based metrics such as FID and IS but crucially also assesses sample novelty, over-fitting, and memorization [9].

Feature Likelihood Divergence between real data distribution (D_T) and generated data distribution (D_G) can be computed as;

$$FLD(D_T, D_G) = -\tfrac{100}{d} logp_{\hat{\sigma}}(D_T|D_G) - C$$

where;

- $logp_{\hat{\sigma}}(D)$ is a Mixture of Gaussian density estimator
- d is the dimension of feature space
- C is a dataset dependent constant

Note that, higher FLD values indicates problems in one or more areas (fidelity, diversity, novelty) evaluated by FLD [9].

3.6 Novel Representation of Precision/Recall

Sampling/generation quality of generative models can be evaluating by redefining the precision and recall metrics which are already commonly used in discriminative tasks.

[10] points out the deficiency of commonly used metrics IS and FID which is being unable to distinguish between different failure cases since they only yield one-dimensional scores. Therefore they propose a new definition for precision and recall such that they will be applicable for probability distributions.

The formal definition made in [10] is given as following;

For $\alpha, \beta \in (0, 1]$ the probability distribution Q has precision α at recall β with respect to another probability distribution P if there exist distributions μ, υ_P, υ_Q such that;

$$P = \beta\mu + (1 - \beta)\upsilon_P \text{ and } Q = \alpha\mu + (1 - \alpha)\upsilon_Q$$

where;

- υ_P stands for the part of P that is "missed" by Q
- υ_Q stands for the noise part of Q

The behaviour of the newly defined for generative model evaluation precision and recall are quite similar to traditional precision and recall concepts. Intuitively;

- precision measures, how much of Q can be generated by a "part" of P
- recall measures, how much of P can be generated by a "part" of Q

when we embrace P as the reference distribution.

4 Latent Representation Metrics

Evaluating generative models using latent representations involves assessing how well the model captures and utilizes the underlying structure of the data in its latent space. Latent space is where the data is encoded into a lower-dimensional representation, capturing the essential features and variations. Latent representations can be evaluated using relevant performance metrics, such as accuracy for semi-supervised learning and reconstruction quality for denoising tasks. For unsupervised tasks, no single metric applies universally. Instead, three commonly used approaches for evaluating unsupervised latent representations are clustering, compression, and disentanglement.

4.1 Clustering

Clusters can be obtained by applying k-means or other clustering algorithms within the latent space of generative models. This approach helps to evaluate how well the generative model organizes the data into meaningful groupings. For labeled datasets, numerous quantitative evaluation metrics can be employed to assess the quality of these clusters. Examples include **completeness score**, **homogeneity score**, and **V-measure score**[7], which provide insights into how accurately and coherently the clusters reflect the underlying data distribution. It is important to note that these labels are used exclusively for evaluation purposes and do not influence the clustering process itself. This ensures that the generative model's ability to learn and represent the data structure remains unbiased and unsupervised, while still allowing for a rigorous assessment of its clustering performance.

4.1.1 Homogeneity

In order to satisfy homogeneity criteria a, a clustering must assign only the data points that are members of a single class to a single cluster. The class distribution within each cluster should be skewed to a single class, that is, zero entropy. It determines how close a given clustering is to this ideal by examining the conditional entropy of the class distribution given the proposed clustering.

$$h = \begin{cases} 1, & \text{if } H(C,K) = 0 \\ \frac{1-H(C|K)}{H(C)}, & \text{else} \end{cases}$$

Where H is the entropy.

$H(C|K)$ is maximal (and equal to $H(C)$) when the class distribution within each cluster is equal to the overall class distribution. $H(C|K)$is 0 when each cluster contains only members of a single class, a perfectly homogenous clustering.

h is maximized when all of its clusters contain only data points that are members of a single class

4.1.2 Completeness

Completeness is symmetrical to homogeneity. In order to satisfy the completeness criteria, a clustering must assign all of those data points that are members of a single class to a single cluster. To evaluate the completeness, the distribution of cluster assignments within each class is examined. In a perfectly complete clustering solution, each of these distributions will be completely skewed to a single cluster

$$c = \begin{cases} 1, & \text{if } H(K,C) = 0 \\ \frac{1-H(K|C)}{H(K)}, & \text{else} \end{cases}$$

In the perfectly complete case, $H(K|C) = 0$. However, in the worst-case scenario, each class is represented by every cluster with a distribution equal to the distribution of cluster sizes, $H(K|C)$ is maximal and equals H(K). Finally, in the degenerate case where $H(K) = 0$, when there is a single cluster, we define completeness to be 1.

c is maximized when all the data points that are members of a given class are elements of the same cluster.

4.1.3 V measure Score

V measure Score is also called normalized mutual information. It is an entropy-based measure that explicitly measures how successfully the criteria of homogeneity and completeness have been satisfied. V measure is computed as the harmonic mean of distinct homogeneity and completeness scores, just as precision and recall are commonly combined into F-measure.

$$V_\beta = \frac{(1 + \beta) * h * c}{(\beta * h) + c}$$

if β is greater than 1 completeness is weighted more strongly in the calculation, if β is less than 1, homogeneity is weighted more strongly

Computations of homogeneity, completeness, and V measure are completely independent of the number of classes, the number of clusters, the size of the data set, and the clustering algorithm used. Thus these measures can be applied to and compared across any clustering solution, regardless of the number of data points (n-invariance), the number of classes, or the number of clusters.

4.2 Compression

Latent representations can be evaluated by assessing their compression capabilities while maintaining high fidelity in reconstruction accuracy. This evaluation often employs conventional metrics such as Mean Squared Error (MSE), Peak Signal Noise Ratio (PSNR), and Structural Similarity Index (SSIM) to indicate the quality of reconstructed data.

4.2.1 Mean Squared Error (MSE)

Mean Squared Error (MSE) is a common metric used to measure the average squared difference between the actual and predicted values in a dataset.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where:

- n is the number of samples in the dataset.
- y_i represents the actual value of the i-th sample.
- $\hat{y}_i$ represents the predicted value of the i-th sample.

4.2.2 Peak Signal to Noise Ratio(PSNR)

Peak Signal-to-Noise Ratio (PSNR) is a metric used to evaluate the quality of a reconstructed or compressed image. It measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. In the context of image processing, PSNR is typically expressed in decibels (dB). The higher the value of PSNR, the better will be the quality of the output image.

$$\begin{aligned} \text{PSNR} &= 10 \cdot \log\left(\frac{\text{MAX}^2}{\text{MSE}}\right) \\ &= 20 \cdot log(MAX) - 10 \cdot log(MSE) \end{aligned}$$

where:

- MAX is the maximum possible pixel value of the image.
- MSE is the Mean Squared Error, computed as the average of the squared differences between the original and the reconstructed/compressed image.

4.2.3 Structure Similarity Index (SSIM)

The Structural Similarity Index (SSIM) is a metric used to quantify the similarity between two images. Unlike traditional metrics such as Mean Squared Error (MSE), SSIM takes into account the perceived changes in **structural information, luminance, and contrast**, which are more aligned with human perception of image quality.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where:

- x and y are the two compared images.
- μ_x and μ_y are the means of x and y, respectively.
- σ_x and σ_y are the standard deviations of x and y, respectively.
- σ_{xy} is the covariance of x and y.
- C_1 and C_2 are constants added to avoid instability when the denominator approaches zero.

4.3 Disentanglement

Disentanglement in generative models refers to the ability of the model to separate and represent different attributes **independently** and in a way that is easy to understand. For example, in an image of a face, factors like pose, identity, expression, and lighting can vary. A generative model with disentanglement would be able to manipulate each of these factors separately without affecting the others. This allows for more precise control over the generated data and a better understanding of its underlying structure. Most of the suggested metrics for measuring disentanglement rely on an encoder network to translate input images into latent codes. However, two new methods for quantifying disentanglement have been introduced with StyleGAN[11]: Perceptual Path Length, Linear separability. These methods do not need an encoder or known factors of variation, making them applicable to any image dataset and generator.

4.3.1 Perceptual Path Length (PPL)

The Perceptual Path Length (PPL)[11] metric evaluates the smoothness of the latent space in generative models by quantifying the perceptual changes in images during interpolation of latent-space vectors. Non-linear changes in interpolated images suggest entangled latent spaces where factors of variation are not properly separated. To measure this, a perceptually-based pairwise image distance metric is employed, calculated as the weighted difference between VGG16 embeddings, reflecting human perceptual judgments. By segmenting the interpolation path into linear segments, the total perceptual length is computed as the sum of perceptual differences over each segment. Although the natural definition involves infinitely fine subdivision, practical approximation is achieved using a small subdivision. The average perceptual path length is then computed over all possible endpoints. The metric considers spherical interpolation for appropriate interpolation in the normalized input latent space and focuses on facial features by cropping images accordingly. The expectation is computed by sampling and dividing by the square of a small subdivision value.

$$I_Z = E[\frac{1}{\epsilon^2} d(G(slerp(z_1, z_2; t)), G(slerp(z_1, z_2; t+\epsilon)))]$$

where I_z is the average perceptual path length in latent space Z, over all possible endpoints. Slerp denotes spherical interpolation, which is one way of interpolating in our normalized input latent space. To focus on facial features rather than the background, the generated images are cropped to include only the face before evaluating the pairwise image metric. Given that the metric d is quadratic, it is divided by the square of epsilon.

$$I_W = E[\frac{1}{\epsilon^2} d(g(lerp(f(z_1), f(z_2); t)), g(lerp(f(z_1), f(z_2); t+\epsilon)))]$$

where the only difference is that interpolation happens in W space. As vectors in W are devoid of any normalization, linear interpolation (lerp) is employed.

4.3.2 Linear separability

In a sufficiently disentangled latent space, directional vectors signify specific image features. We measure this with linear separability, which measures how easily we can divide latent-space points into two groups based on image attributes. A classifier is utilized to label generated images, with linear SVMs predicting labels and conditional entropy ensuring consistency. To assess separability, samples are sorted by classifier confidence, keeping the most confident half. Then, for each attribute, linear SVMs are fitted to predict labels based on latent-space points, and conditional entropy is calculated to gauge the additional information needed to determine a sample's true class. Finally, the separability score is computed.

$$exp(\sum_i H(Y_i|X_i))$$

where i is the enumeration of the attributes.

4.3.3 Factor-VAE

FactorVAE[12], quantifies disentanglement by evaluating how well latent variables capture variations when a specific generative factor is fixed. The process involves selecting a generative factor, generating a batch of vectors with this factor held constant, and encoding these vectors into latent codes. These codes are then normalized by their empirical standard deviation, and the variance of each dimension is computed. The dimension with the lowest variance, along with the fixed factor, provides a training point for a classifier. FactorVAE is the accuracy of this classifier in predicting the fixed generative factor, thereby assessing the model's disentanglement capability.

4.3.4 Mutual Information Gap (MIG)

The Mutual Information Gap (MIG)[13] is a metric for assessing disentanglement by measuring the informativeness between generative factors and latent variables using mutual information. Mutual information I(c;z) quantifies how much knowing one variable reduces uncertainty about another, being nonnegative and zero only if the variables are independent. To compute the MIG, a matrix of mutual information values between each generative factor and latent variable is created. For each generative factor, the difference between the highest and second-highest mutual information values is normalized by the entropy of the generative factor. The MIG score is the average of these normalized differences, indicating how well each generative factor is distinctly represented by the latent variables.

$$I(c; z) = H(z) - H(z|c)$$

where H is the entropy.

$$MIG(c,z) = \frac{1}{K} \sum \frac{I_{i_k,k} - max_{l \neq i_k} I_{l,k}}{H(z_k)}$$

where $i_k = argmax_i I_{i,k}$

5 Text Based Generative Model Evaluation Metrics

5.1 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

ROGUE is a set of metrics used to evaluate the quality of text summaries by comparing them to reference summaries or human-generated summaries. ROGUE metrics assess how well the information in the generated summary overlaps with the information in the reference summary, focusing on recall rather than precision. ROGUE includes several variants, such as **ROGUE-N**, and **ROGUE-L**. These metrics are commonly used in natural language processing tasks such as text summarization and machine translation to quantitatively assess the quality of generated text summaries.

5.1.1 ROGUE-N

ROGUE-N is used for measuring overlapping n-grams.

$$R = \frac{\text{Number of common n-grams}}{\text{Total number of n-grams in reference summary}}$$

This formula gives the recall score for ROGUE-N, which measures how much of the information in the reference summary is captured by the generated summary in terms of n-gram overlap.

5.1.2 ROGUE-L

ROGUE-L is used for measuring the longest common subsequence of words.

$$R = \frac{\text{Length of LCS}}{\text{Total number of words in reference summary}}$$

This formula calculates the recall score for ROGUE-L, which measures how much of the information in the reference summary is captured by the generated summary based on the longest common subsequence of words.

5.2 Bilingual Evaluation Understudy (BLEU)

BLEU is a metric commonly used to evaluate the quality of machine-generated translations by comparing them to human-generated translations or reference translations. It measures the similarity between the generated translation and one or more reference translations based on n-gram overlap and brevity penalty. BLEU considers precision, which measures how many of the generated n-grams match the reference n-grams, and brevity penalty, which penalizes translations that are shorter than the reference translations.

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} \frac{1}{N} \times \log\left(\text{precision}_n\right)\right)$$

where BP is the brevity penalty, which penalizes translations that are shorter than the reference translations.

$$\text{BP} = \begin{cases} 1 & \text{if}\, c > r \\ e^{(1-\frac{r}{c})} & \text{if}\, c \leq r \end{cases}$$

Where c is the length of the generated translation and r is the effective reference length (the length of the reference translation(s) closest to c).

6 Conclusion

The evaluation of generative models is a complex process that requires careful consideration of a range of metrics, depending on the specific goals of the model and the tasks it is designed to perform. This document discusses a wide range of evaluation metrics for generative models, categorised into density estimation, sampling/generation, and latent representation metrics.

Density estimation metrics, such as log-likelihood, kernel density estimation, importance sampling, Laplace approximation, and Parzen window density estimation, provide insights into how well generative models capture the underlying data distribution. Sampling/generation metrics, such as Inception Score, Frechet Inception Distance, Maximum Mean Discrepancy, Kernel Inception Distance, and Feature Likelihood Divergence, assess the quality and diversity of generated samples. Latent representation metrics, including clustering, compression, and disentanglement, evaluate how well generative models capture and utilize the underlying structure of the data in their latent spaces.

Furthermore, evaluation metrics tailored for text-based generative models are utilized. ROUGE and BLEU assess the quality of generated text summaries and translations by comparing them to human-generated or reference summaries and translations.

7 References

[1] Weglarczyk, S. (2018). Kernel density estimation and its application. In ITM web of conferences (Vol. 23, p. 00037). EDP Sciences.

[2] Kloek, T.; van Dijk, H. K. (1978). ”Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo” (PDF). Econometrica. 46 (1): 1–19. doi:10.2307/1913641. JSTOR 1913641

[3] Neal, R. M. (2001). Annealed importance sampling. Statistics and computing, 11, 125-139.

[4] Liu, Q., Xu, J., Jiang, R., Wong, W. H. (2021). Density estimation using deep generative neural networks. Proceedings of the National Academy of Sciences, 118(15), e2101344118.

[5] Theis, L., Oord, A. V. D., Bethge, M. (2015). A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844.

[6] Gretton, A., Borgwardt, K. M., Rasch, M. J., Scholkopf, B., and Smola, A. J. A kernel two-sample test. Journal of Machine Learning Research, 13:723–773, 2012a.

[7] Rosenberg, Andrew, Hirschberg, Julia. (2007). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure.. 410-420.

[8] Bińkowski, M., Sutherland, D. J., Arbel, M., Gretton, A. (2018). Demystifying mmd gans. arXiv preprint arXiv:1801.01401.

[9] Jiralerspong, M., Bose, J., Gemp, I., Qin, C., Bachrach, Y., Gidel, G. (2024). Feature likelihood score: Evaluating the generalization of generative models using samples. Advances in Neural Information Processing Systems, 36.

[10] Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., Gelly, S. (2018). Assessing generative models via precision and recall. Advances in neural information processing systems, 31.

[11] Karras, T., Laine, S., & Aila, T. (2018). A Style-Based generator architecture for generative adversarial networks. *arXiv (Cornell University).* https://doi.org/10.48550/arxiv.1812.04948

[12] Kim, H. and Mnih, A. Disentangling by factorising. arXiv preprint arXiv:1802.05983, 2018.

[13] Chen, T. Q., Li, X., Grosse, R., and Duvenaud, D. Isolating sources of disentanglement in variational autoencoders. arXiv preprint arXiv:1802.04942, 2018.