Production Systems and Information Engineering
Volume 12 (1), pp. 64–76.
https://doi.org/10.32968/psaie.2024.1.6

# APPLYING DEEP GENERATIVE MODELS TO PORTRAIT ART GENERATION: A COMPARATIVE STUDY OF GANS AND VAES

ALI JLIDI
University of Miskolc, Hungary
Institute of Information Technology
jlidi.ali@student.uni-miskolc.hu

LÁSZLÓ KOVÁCS
University of Miskolc, Hungary Institute of Information Technology
kovacs@iit.uni-miskolc.hu

**Abstract:** Deep Generative Models (DGMs) have emerged as powerful tools for generating diverse and realistic data across various domains. This paper presents a comprehensive systematic review of existing DGMs, discovering their methodologies, architectures, and applications. We delve into the fundamental concepts of Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), highlighting their respective strengths and weaknesses. Furthermore, we provide a detailed case study focusing on the utilization of GANs and VAEs for generating images of portrait art. By employing a dataset of portrait artworks, we demonstrate the capabilities of these DGMs in capturing the latent representation to generate new art's. Through a comparative analysis of the generated results, we evaluate the likelihood and the inception score achieved by each model. By diving into theoretical insights with practical experimentation, this paper offers valuable insights into DGMs and their potential applications. The findings and discussions presented contribute to a deeper understanding of deep generative modeling techniques and pave the way for future advancements the field.

***Keywords:*** *DGM, GAN, VAE, likelihood, Inception score*

## 1. Introduction

In recent years, the advent of Deep Generative Models (DGMs) [1] has revolutionized the field of artificial intelligence and machine learning. These models, have demonstrated remarkable capabilities in generating data that closely resemble samples from the underlying distribution [2]. This ability holds immense promise across various applications, ranging from image synthesis, text generation and graph generation to drug discovery and data augmentation [3, 4, 5].

The primary objective of this paper is to conduct a review of existing deep generative models, with a particular focus on understanding their methodologies, architectures, and applications.

Key emphasis will be placed on two prominent paradigms in deep generative modeling: Generative Adversarial Networks (GANs) [6] and Variational Autoencoders (VAEs) [7]. These two models are widely used, each offering distinct advantages and trade-offs related to data generation.

Deep generative models represent a class of neural networks comprising numerous hidden layers [8] trained to approximate complex, multidimensional probability distributions [9]. Their fundamental objective is to learn and approximate elusive or computationally infeasible probability distributions from a limited set of independent and identically distributed samples [10]. Upon successful training, these models can assess the likelihood of a given sample and generate new samples resembling those from the underlying distribution [11]. While these challenges have long been central to the realms of probability and statistics, their resolution remains computationally daunting, particularly in high-dimensional spaces [12].

Despite significant progress and notable successes, the field of generative modeling grapples with several unresolved issues. There are mainly three mathematical challenges first is training deep generative models poses an ill-posed problem due to the impossibility of uniquely identifying a probability distribution from a finite sample set [13, 14]. Consequently, the model's performance heavily relies on hyper-parameters such as network design [15], training objectives, regularization techniques, and training algorithms.

Second limitation is evaluating the similarity between samples generated by the model and those from the target distribution necessitates either inverting the generator or comparing the generated sample distribution to the dataset [16]. Both approaches present distinct challenges. Inverting a neural network-based generator is difficult, especially given its inherent nonlinearity. Moreover, quantifying distribution similarity leads to complex two-sample test problems, exacerbated by the absence of prior assumptions on the distributions [17].

Last struggle is that many DGM training methods assume approximating the target distribution by transforming a simpler known distribution (e.g., Gaussian) in a latent space of a defined dimension [18]. However, determining this latent space dimension often proves infeasible, leaving it as a crucial yet challenging hyper-parameter [19]. An erroneous estimate may compromise data approximation or render the generator non-injective, impeding effective training [20].

In essence, this paper endeavors to provide a comprehensive overview of deep generative modeling techniques while offering practical insights into their real-world applications. By synthesizing theoretical knowledge with empirical experimentation, we aim to contribute to the trending field of generative modeling and inspire future research endeavors in this domain.

The forthcoming section focus on the theoretical behind data generation, discovering the principles and frameworks that guide the creation of novel datasets.

Subsequently, the methodology segment will expound on the dataset employed in this study, alongside a detailed examination of the two models deployed for the generation of new portrait artworks. The result section is dedicated to presenting the outcomes of the applied models, displaying the efficacy and potential of the generative techniques. The paper ends with a conclusion that synthesizes the findings, reflects on the implications of the study, and offers avenues for future research in the realm of artistic data generation.

## 2. Literature review

Generative artificial intelligence (AI) first emerged in the 1960s with rudimentary applications like chatbots. However, it was not until 2014, marked by the advent of generative adversarial networks (GANs) [21], a type of machine learning algorithm, that generative AI achieved the capability to produce remarkably authentic images, videos, and audio of real individuals.

This breakthrough has started a wave of opportunities, including enhanced movie dubbing and the creation of immersive educational content. Yet, it has also raised significant concerns, particularly surrounding the proliferation of deepfakes—digitally manipulated images or videos—and the potential for malicious cyberattacks on businesses. These attacks might involve deceptively realistic requests impersonating an employee’s superior.

In addition to GANs, two recent advancements have been instrumental in propelling generative AI into the mainstream: transformers and the groundbreaking language models they facilitated. Transformers represent a novel approach to machine learning [22], enabling researchers to train increasingly large models without the need to pre-label all data. Consequently, these models can be trained on vast amounts of text data, yielding responses with greater complexity and nuance [23].

Moreover, transformers introduced the concept of attention, allowing models to comprehend connections between words not only within individual sentences but also across entire documents, including pages, chapters, and books. This capability extends beyond textual data; transformers can also analyze diverse data types such as code, proteins, chemicals, and DNA by leveraging their capacity to track intricate connections [24, 25].

### *2.1. Overview of Generative Models*

Generative modeling aims at capturing the essence of complex, as probability distributions defined over high-dimensional spaces $\mathbb{R}^n$ [26]. These distributions can exhibit intricate twisted structures, including multimodality and disjoint support, presenting significant challenges for conventional statistical techniques. Unlike traditional statistical inference, which seeks explicit mathematical expressions for probabilities, generative modeling leverages a different approach.

$$g : R^q \to \mathrm{R}^n$$

The primary objective is to develop a generator capable of synthesizing data that closely resembles samples drawn from a distribution $Z$ supported in $R^q$ to points in R $^n$ [27]. This process counts on utilizing a finite yet potentially extensive set of independent and identically distributed samples from the target distribution $X$, collectively termed as the training data. These samples serve as the foundation for training the generator, empowering it to capture the underlying characteristics of the distribution and generate novel data points that adhere to its inherent structure [28]. In Figure 1, demonstrate a deep generative model $g_\theta$ trained to map samples from a simple distribution$Z$, which resemble to the real distribution $Z$ finding a function that quantifies the discrepancy between the generated samples and the original examples is the key obstacle to training generative models.

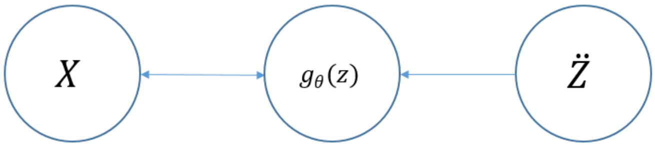


**Figure 1.** Mapping from a simple distribution $\ddot{\boldsymbol{Z}}$ a variable x

A fundamental challenge in training generative models lies in finding an objective function that effectively measures the difference between the generated samples and the original examples [29]. Specifically, when there are no direct correspondences between data samples and latent variables, as is often the case, this task becomes even more daunting. The goal is to train a deep generative model, denoted as $g_\theta$, to transform samples from a simple distribution, $\ddot{Z}$(Right), into a more complex distribution $g_\theta(Z)$ (middle), resembling the true distribution X (left).

Given that the mapping from a latent variable **z** to an observed variable x is often unknown, **z** is commonly termed the latent variable, and the space it inhabits is referred to as the latent space, denoted as $\ddot{\boldsymbol{Z}}$. In our discussion, we adopt the assumption that $\ddot{\boldsymbol{Z}}$ follows a univariate Gaussian distribution in $\mathbb{R}^p$ [30]. This choice have been made for convenience, and does not limit the generality of the approach; any known distribution that allows sampling from $\ddot{\boldsymbol{Z}}$ and, in some cases, computing the probability density function $p_z(\mathbf{z})$can be utilized [31].

It's crucial to acknowledge that the dimensionality of the latent space, denoted as $\boldsymbol{q}$, may differ from that of the data space, denoted as $\boldsymbol{n}$. For instance, in the context of high-resolution images composed of millions of pixels, the inherent structure of the images is not accurately represented in this high-dimensional space. Instead, there exists a concealed manifold of typically unknown dimension where the essential features of the images reside. This adds another layer of complexity to the problem [32]. Assuming that we have knowledge of the generator function $\mathbf{g}$, we can generate new data points by sampling $\boldsymbol{z}$ from $\ddot{\boldsymbol{Z}}$ and computing $g(z)$. in many practical scenarios, such as generating deep fakes or in Bayesian statistics, the sole objective is to produce new samples. Additionally, the generator can be employed to calculate the likelihood or evidence of a specific sample $\boldsymbol{x}$ through the process of marginalization.

### *2.2. Generative Models and Their Application*

Generative models come in various forms, each with its unique approach to understanding and generating data. Here is a more comprehensive list of some of the most prominent types and thier applications:

**Table 1.** Generative models and their applications

| Generative Model | Description | Application |
|---|---|---|
| Bayesian Networks | Graphical models representing probabilistic relationships among variables. | Medical diagnosis, Causal relationship analysis |
| Diffusion Models | Models describing how things spread or evolve over time. | Rumor spreading analysis, Virus spread prediction |
| Generative Adversarial Networks (GANs) | Two neural networks (generator and discriminator) trained together for data generation. | Realistic image synthesis, Style transfer, Data augmentation |
| Variational Autoencoders (VAEs) | Autoencoders that produce and decode compressed representations of data. | Image generation, Anomaly detection, Data compression |
| Restricted Boltzmann Machines (RBMs) | Neural networks learning a probability distribution over inputs. | Collaborative filtering, Feature learning, Dimensionality reduction |
| Pixel Recurrent Neural Networks (PixelRNNs) | Models generating images pixel by pixel. | Text generation, Image completion, Density estimation |
| Markov Chains | Models predicting future states based on the current state. | Text generation |
| Normalizing Flows | Series of invertible transformations applied to probability distributions. | Density estimation, Image synthesis, Anomaly detection |

## 3. Methodology

In this study, we employed two generative models, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), to generate new data from portrait art images. Their adversarial training process characterizes GANs, where a generator network learns to produce realistic samples while a discriminator network distinguishes between real and generated samples. On the other hand, VAEs operate by learning a latent representation of input data and generating new samples by sampling from this learned distribution. The following section dive in the details of the methodology used

### *3.1. Dataset (oil painted portraits)*

As a high-dimensional example in which the data's intrinsic dimensionality is clearly less than n.

We consider the portraits dataset shown in *Figure 2*, the dataset consists of color-valued digital images; each will be resized to 128 × 128 pixels and showing a portrait. The dataset provides 4117 images. To train the generator, we do not require labels; however, the first obstacle to setting up the DGM training is that the intrinsic dimension of the images Dataset is unknown, which renders choosing the dimension of the latent space non-trivial. While each image contains n = 16,384 pixels, the support of $X$ will likely lie in a subset of a much lower dimension. In addition, since the images are grouped into unknown different classes, one can expect the support to be disjoint with a substantial distance between the different clusters.

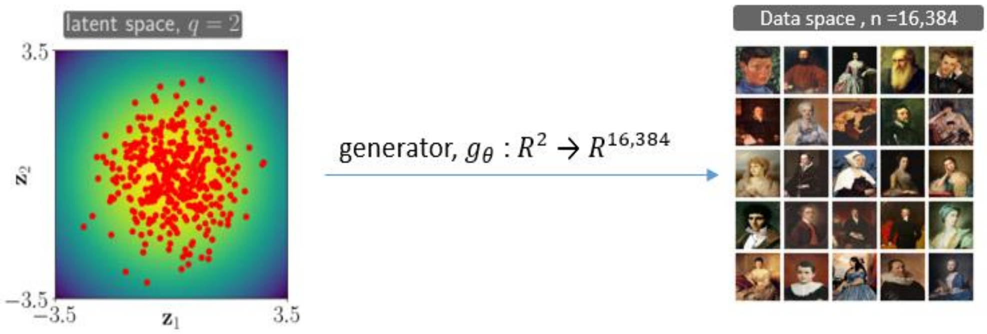


**Figure 2.** From a space distribution to a 3 dimensional tensor

In Figure 2 is the Illustration of the portraits image generation process, the intrinsic dimension of the dataset (right) is unknown but assumed to be much less than the number of pixels per image, n = 16,384. In our example, we define the latent variable to be distributed according to the q = 2-dimensional standard normal distribution (left), the generator cannot be assumed invertible. This complicates the density estimation and the training process.

### *3.2. Variational Autoencoders*

In most practical scenarios, assuming equal dimensions for the latent space and the data space is not feasible. This poses a challenge for directly applying flow models, as the generator lacks inevitability, potentially resulting in unbounded or ill-defined KL divergence. To circumvent this constraint, Variational Autoencoders (VAEs) [33, 34, 35] offer a widely adopted solution. VAEs typically employ a latent space of significantly lower dimensionality compared to the data space, denoted as $q << n$. additionally; VAEs provide enhanced control over the latent space dimensionality, as discussed subsequently. Since the generator $g_\theta$ is not invertible, computing the negative log-likelihood loss directly is unattainable. Recall that the likelihood of a sample **x** drawn from $\ddot{\boldsymbol{X}}$ (also termed its evidence) implied by the generator is denoted as$P_\theta(x)$. It is worth noting that employing Bayes's rule allows for the re-expression of the likelihood in the following way:

$$p_\theta(x) = \frac{p_\theta(x,z)}{p_\theta(z|x)}$$

We train the generator for the art generation using the VAE approach. Recall that we defined the latent space to be two dimensional. Since the image intensities are in [0, 1], we measure the reconstruction quality using the Bernoulli likelihood [x]. We use the same architecture of the neural network used to compute the mean and covariance of the approximate posterior as in the excellent VAE tutorial [36], but note that our generator is different. For a given art image x we use two convolution layers for feature extraction

$$h^{(1)} = \sigma Relu(C^{(1)}_{VAE}x + c^{(1)}_{VAE})$$
$$h^{(2)} = \sigma Relu(C^{(2)}_{VAE}h^{(1)} + c^{(1)}_{VAE})$$

Here, $C^{(1)}_{VAE}$and $C^{(2)}_{VAE}$ are convolution operators with 4 × 4 stencils and strides of two, that is, they reduce the number of pixels by a factor of two in each axis. The first layer has 32 hidden channels and the second layer has 64 hidden channels. The bias vectors $c^{(1)}_{VAE}$ and $C^{(2)}_{VAE}$ apply constant shifts to each channel. Given the feature $h^{(2)}$, we compute the mean and the diagonal of the covariance of the approximate posterior, $e_\psi$(z|x)

### *3.3. Generative Adversarial Networks*

In Generative Adversarial Networks (GANs), we train the model's parameters $(\theta)$ by minimizing a loss function that quantifies the difference between the generated samples $g_\theta(Z)$ and real data (X). Unlike other methods like VAEs, GANs compare distributions directly in data space. They are termed likelihood-free models because they do not rely on sample likelihood or lower bounds, and they do not infer latent variables. GANs' popularity is growing due to promising results [37, 38], and several excellent works that go beyond this work [39, 40, 41]. One challenge in GAN training is defining an effective loss function that measures the difference between generated and real samples without known correspondences. Standard approaches involve using a discriminator network, leading to a challenging optimization problem resembling a two-player game between the generator and discriminator.

We continue the art generation and seek to train the generator along with a discriminator whose architecture is similar to the one used in Deep Convolutional GAN (DCGAN) [42]. To be specific, we define the discriminator using two convolution layers and one fully connected layer: that is, given the input feature $\boldsymbol{x} \in R^{\,n}$, we predict the probability

That x is sampled from the true dataset using

$$\boldsymbol{v}^{(1)} = \sigma Relu(N(C^{(1)}_{GAN}x + c^{(1)}_{GAN}))$$
$$\boldsymbol{v}^{(2)} = \sigma Relu(N(C^{(2)}_{GAN}\mathbf{v}^{(1)} + c^{(2)}_{GAN}))$$

$$d_{\emptyset}(x) = \sigma sigma((d_{GAN})^T \mathbf{v}^{(2)} + \delta_{GAN})$$

Here, $C_{GAN}^{(1)}$ and $C_{GAN}^{(2)}$ are convolution operators, $d_{GAN}$ is a vector, $c_{GAN}^{(1)}$), $c_{GAN}^{(2)}$, $\delta_{GAN}$ are bias terms, $N$ is a batch normalization layer, and σReLU is the leaky ReLU activation.

In training, we perform gradients approximated using minibatches of size 64 and using the ADAM scheme. We use fixed learning rates of 0.0003 and, as proposed in [42], a momentum of 0.5. We observed that the training performance is highly dependent on these parameter choices and that, for instance, changes in the batch size can quickly lead to complete failure of the training. We perform a fixed number of 100 training epochs.

## 4. Results

### *4.1. Visual inspection*

In comparing the outputs of two distinct models, a visual inspection reveals intriguing nuances between the results generated by Variational Autoencoder (VAE) and Generative Adversarial Network (GAN) architectures with the dataset of oil-painted portraits. The VAE output exudes a certain fidelity to the original data, capturing subtle details with a touch of realism, occasionally lacking in boldness and imaginative touch. On the contrary, the GAN's creations boast a striking dynamism, infusing the portraits with vivid colors and expressive strokes, a sometimes at the expense of accuracy in finer details. The VAE's approach leans towards precision, faithfully recreating features, while the GAN's method leans towards creativity, often diffuse the portraits with a captivating sense of emotion and narrative. This visual inspection underscores the diverse artistic interpretations these models offer

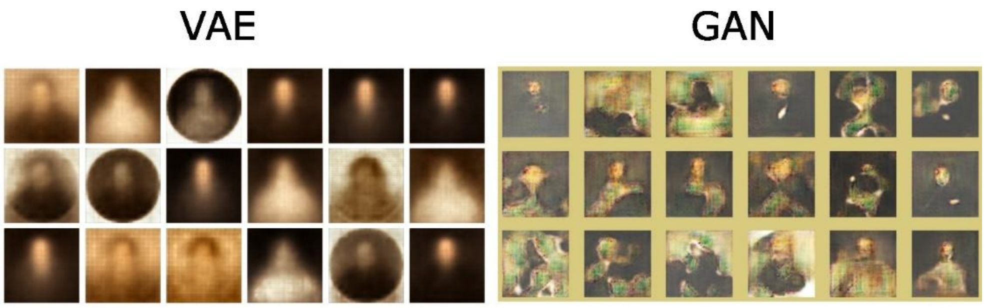


**Figure 1.** 18 Images generated (VAE versus GAN)

### *4.2. Log-Likelihood*

Average log-likelihood is widely considered as the default measure for quantifying generative image modeling performance. Consider images $\boldsymbol{x} \in \{0, \ldots, 255\}^D$ with a discrete probability distribution$P(x)$, uniform noise u $\in [0, 1[^D$, and noisy data **y = x+u**. If p refers to the noisy data density and q refers to the model density.

The following table shows the value of the likelihood of some generated image by the GAN and the VAE

**Table 1.** Log-Likelihood of VAE and GAN for the generated images

| | GAN Model | VAE Model |
|---|---|---|
| Generated Image 1 LL | –5.64 | 0.73 |
| Generated Image 2 LL | –5.15 | 1.09 |
| Generated Image 3 LL | –5.34 | 1.27 |
| Generated Image 4 LL | –5.24 | 0.81 |
| Generated Image 5 LL | –4.68 | 1.50 |
| Generated Image 6 LL | –4.79 | 1.19 |

The log-likelihood values for generated images by both GAN and VAE models serve as measures of how well these models capture the data distribution. Higher values indicate a better match with the true data distribution. Comparing these values allows us to assess the relative performance of the models, aiding in model selection. However, it is crucial to consider other factors like visual quality and computational efficiency alongside log-likelihood values for comprehensive model evaluation and decision-making.

## 5. Conclusion

In conclusion, this study offers a comprehensive exploration of deep generative models, focusing on their methodologies, applications, and challenges. Through a detailed case study on portrait art generation using Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), we have demonstrated their potential in capturing and generating realistic data.

Our investigation not only sheds light on the technical intricacies of these models but also highlights their practical implications in creative endeavors such as art generation. By synthesizing theoretical insights with practical experimentation, this research contributes to the broader understanding of deep generative modeling techniques. The findings presented here not only expand our knowledge base but also inspire future research and innovation in the field of artificial intelligence and machine learning. With further refinement and exploration, these generative models hold promise for transformative applications across various domains, from entertainment and design to healthcare and beyond, ultimately shaping the future of AI-driven creativity and problem solving.

As machine learning, especially deep learning, continues to advance, the training of more sophisticated generative models becomes increasingly feasible. However, numerous unanswered questions and challenges persist, ensuring ongoing research activity in the realm of deep generative modeling. Looking ahead, we aim to identify some avenues for future exploration that extend beyond the scope of our current paper.

Central to deep generative modeling is the task of effectively comparing complex, high-dimensional probability distributions. This challenge has long been a cornerstone of statistical theory, and leveraging recent advancements in generative modeling to address it represents a promising avenue for future investigation. Bridging the gap between theoretical insights and practical implementation is crucial for enhancing the reliability of DGM training and mitigating the substantial computational burdens associated with it. Our paper has highlighted specific challenges such as the sampling issue in Variational Autoencoders (VAEs) and the implementation of the Lipschitz constraint in training Wasserstein Generative Adversarial Networks (WGANs).

While many existing DGM approaches rely on black-box neural networks as generators, there remains a notable absence of models that integrate domain-specific knowledge. This limitation proves particularly significant in scientific contexts, where tailored models could offer substantial benefits. Addressing this gap presents an exciting avenue for future research, one that holds the potential to unlock new possibilities and applications across diverse domains.

## References


[1] Yelmen, B., & Jay, F. (2023). An Overview of Deep Generative Models in Functional and Evolutionary Genomics. *Annual Review of Biomedical Data Science*. https://doi.org/10.1146/annurev-biodatasci-020722

[2] Drummond, A.J., Rambaut, A. BEAST (2007). Bayesian evolutionary analysis by sampling trees. *BMC Evol Biol*, 7, 214 (2007). https://doi.org/10.1186/1471-2148-7-214

[3] Stanislav Frolov, Tobias Hinz, Federico Raue, Jörn Hees, Andreas Dengel (2021). Adversarial text-to-image synthesis: A review. *Neural Networks*, Volume 144. https://doi.org/10.1016/j.neunet.2021.07.019.

[4] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang (2022). A Survey of Knowledge-enhanced Text Generation. *ACM Comput. Surv.*, 54, 11s, Article 227 (January 2022), 38 pages. https://doi.org/10.1145/3512467

[5] Kovács, L., & Jlidi, A. (n.d.). Graphs Unveiled: Graph Neural Networks and Graph Generation. *Production Systems and Information Engineering*, 11 (1), p. 54. https://doi.org/10.32968/psaie.2023.1.5

[6] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65. https://doi.org/10.1109/MSP.2017.2765202

[7] Diederik P. Kingma and Max Welling (2019). An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, Vol. 12, No. 4, pp 307–392. http://dx.doi.org/10.1561/220000005

[8] Kawaji, H., Nakamura, M., Takahashi, Y. et al. (2008). Hidden layers of human small RNAs. *BMC Genomics*, 9, 157. https://doi.org/10.1186/1471-2164-9-157

[9] Ashby, F. Gregory (2014). Multivariate probability distributions. In: *Multidimensional models of perception and cognition*. Psychology Press, pp. 1–34.

[10] Stein, C. M. (1985). On the coverage probability of confidence sets based on a prior distribution. *Banach Center Publications*, 1 (16), pp. 485–514.

[11] Blume, J. D. (2002). Likelihood methods for measuring statistical evidence. *Statistics in medicine*, 21 (17), pp. 2563–2599. https://doi.org/10.1002/sim.1216

[12] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13 (7), pp. 1443–1471. https://doi.org/10.1162/089976601750264965

[13] Manski, C. F. (2003). *Partial identification of probability distributions*. Springer Science & Business Media.

[14] Cella, L. (2023). Finite sample valid probabilistic inference on quantile regression. In: *International Symposium on Imprecise Probability.Theories and Applications*. 2023, July, pp. 109–118, PMLR.

[15] Smithson, S. C., Yang, G., Gross, W. J., and Meyer, B. H. (2016). Neural networks designing neural networks: Multi-objective hyper-parameter optimization. *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Austin, TX, USA, pp. 1–8. https://doi.org/10.1145/2966986.2967058

[16] Choi, Seoung-Ho et Jung, Sung Hoon (2019). Similarity analysis of actual fake fingerprints and generated fake fingerprints by dcgan. *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 19, no 1, pp. 40–47. https://doi.org/10.5391/IJFIS.2019.19.1.40

[17] Eigenschink, P., Reutterer, T., Vamosi, S., Vamosi, R., Sun, C., and Kalcher, K. (2023). Deep Generative Models for Synthetic Data: A Survey. *IEEE Access*, vol. 11, pp. 47304–47320. https://doi.org/10.1109/ACCESS.2023.3275134

[18] Smith, Michael Stanley, Rubén Loaiza-Maya, and David J. Nott (2020). High-Dimensional Copula Variational Approximation Through Transformation. *Journal of Computational and Graphical Statistics*, 29, no. 4, pp. 729–43. https://doi.org/10.1080/10618600.2020.1740097

[19] Vahdat, A., Kreis, K., & Kautz, J. (n.d.). *Score-based Generative Modeling in Latent Space*. https://github.com/NVlabs/LSGM.

[20] Sajjadi, Mehdi S. M, Scholkopf, B., Hirsch, M. (2017). Enhancenet: Single image super-resolution through automated texture synthesis. In: *Proceedings of the IEEE international conference on computer vision*. pp. 4491–4500.

[21] Jones, E. A., Wang, F., and Ozpineci, B. (2014). Application-based review of GaN HFETs. *2014 IEEE Workshop on Wide Bandgap Power Devices and Applications*, Knoxville, TN, USA. https://doi.org/10.1109/WiPDA.2014.6964617

[22] Tianyang Lin, Yuxin Wang, Xiangyang Liu, Xipeng Qiu (2022). A survey of transformers. *AI Open*, Volume 3. https://doi.org/10.1016/j.aiopen.2022.10.001

[23] Minaee, Shervin, Mikolov, Tomas, Nikzad, Narjes, et al. (2024). *Large language models: A survey*. arXiv preprint arXiv:2402.06196. https://doi.org/10.48550/arXiv.2402.06196

[24] Yenduri, Gokul, Srivastava, Gautam, Maddikunta, Praveen Kumar Reddy, et al. (2023). *Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions*. arXiv preprint arXiv:2305.10435.

[25] Rosário, Albérico Travassos (2024). Generative AI and Generative Pre-Trained Transformer Applications: Challenges and Opportunities. *Making Art With Generative AI Tools*. https://doi.org/10.4018/979-8-3693-1950-5.ch00

[26] Deneu, Benjamin, Servajean, Maximilien, Bonnet, Pierre, et al. (2021). Convolutional neural networks improve species distribution modelling by capturing the spatial structure of the environment. *PLoS computational biology*, vol. 17, no 4, p. e1008856. https://doi.org/10.1371/journal.pcbi.1008856

[27] Figueira, Alvaro, Vaz, Bruno (2022). Survey on synthetic data generation, evaluation methods and GANs. *Mathematics*, vol. 10, no 15, p. 2733. https://doi.org/10.3390/math10152733

[28] Baum, Eric, Wilczek, Frank (1987). Supervised learning of probability distributions by neural networks. Neural information processing systems.

[29] Pilar, Philipp, Wahlström, Niklas (2023). *Probabilistic matching of real and generated data statistics in generative adversarial networks*. arXiv preprint arXiv:2306.10943. https://doi.org/10.48550/arXiv.2306.10943

[30] Ahrendt, P. (2005). The Multivariate Gaussian Probability Distribution. www.imm.dtu.dk/~pa

[31] Diebolt, Jean et Robert, Christian P. (1994). Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 56, no 2, pp. 363–375.

[32] Kumar, Brajesh, Dikshit, Onkar, Gupta, Ashwani, et al. (2020). Feature extraction for hyperspectral image classification: A review. *International Journal of Remote Sensing*, vol. 41, no 16, pp. 6248–6287. https://doi.org/10.1080/01431161.2020.1736732

[33] Kingma, D. P., and Welling, M. (2013). *Auto-Encoding Variational Bayes*. Dec. 2013 arXiv:1312.6114.

[34] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*. Jan. 2014. arXiv:1401.4082.

[35] Kingma, D. P., Welling, M. et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12 (4), pp. 307–392. http://dx.doi.org/10.1561/2200000056

[36] *SmartGeometry at UCL*. CreativeAI: Deep Learning for Graphics Tutorial Code, 2020-12-3. https://github.com/smartgeometry-ucl/dl4g

[37] Brock, A., Donahue, J., and Simonyan, K. (2018). *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. Sept. 2018. arXiv:1809.11096

[38] Karras, T., Laine, S., and Aila? T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. *CVPR*, pp. 4401–4410.

[39] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). *Generative adversarial nets*. Volume 27, pp. 2672–2680.

[40] Goodfellow, I. (2016). *NIPS 2016 Tutorial: Generative Adversarial Networks*. Dec, arXiv:1701.00160. https://doi.org/10.48550/arXiv.1701.00160

[41] Arjovsky, M., and Bottou, L. (2017). *Towards Principled Methods for Training Generative Adversarial Networks*. Jan., arXiv:1701.04862. https://doi.org/10.48550/arXiv.1701.04862

[42] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). *Improved techniques for training gans*. pp. 2234–2242.