IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 16, Issue 4, Ser. V (Jul. – Aug. 2020), PP 59-66
www.iosrjournals.org

Applications of Probability Theory in Generative AI Models

Jag Pratap Singh
Professor, Govt. Degree College Nadha Bhoor Sahaswan, Badaun

Abstract
In recent years, generative artificial intelligence (AI) models have revolutionised tasks in image synthesis, text generation, and multimodal content creation. The foundational capacity of these models to learn, model and sample from complex data distributions is fundamentally rooted in probability theory. This paper presents a mathematical perspective on how probability theory underlies and enables modern generative AI: from latent-variable models, variational and flow-based methods, to adversarial and diffusion-based generative architectures. We formalise how models define and optimize probability distributions $p_\theta(x)$, latent–variable priors $p(z)$, and conditional densities $p(x \mid z)$, and how divergence minimisation (e.g., KL-divergence, Jensen-Shannon divergence) and likelihood maximisation are employed during training. Key equations such as$p_\theta(x) \approx \int p_\theta(x \mid z)\, p(z)\, dzand\min_\theta D_{KL}\,(q_\phi(z \mid x) \parallel p(z)) - \mathbb{E}_{q_\phi(z|x)}\,[\ln p_\theta(x \mid z)]$are explored in the context of variational autoencoders. Later sections examine applications of probability theory in generative adversarial networks (GANs) via divergence games, normalizing flows through change-of-variable formulae, and diffusion models via score-based SDEs. We also discuss how probability theory supports evaluation metrics (e.g., likelihood, bits-per-dimension) and sampling strategies, as well as emerging challenges such as high-dimensional integration, mode collapse, and measurement of generative quality. By emphasising the probabilistic foundations, this paper aims to provide researchers with a coherent mathematical framework to analyze, compare, and design generative AI models.

Keywords: *generative AI, probability theory, latent-variable models, variational autoencoder, normalizing flows, generative adversarial networks, diffusion models, divergence minimisation, likelihood modelling.*

I. Introduction

Probability theory provides the mathematical foundation upon which all generative artificial intelligence (AI) models are built. From classical Bayesian networks to modern large-scale diffusion and transformer-based architectures, the essence of generation lies in modeling uncertainty and learning data distributions. A generative model attempts to approximate an unknown probability distribution $p_{\text{data}}(x)$over complex high-dimensional data—such as images, text sequences, or audio signals—by learning a parameterized model $p_\theta(x)$that captures the underlying structure and variability of the observed data.

Formally, the objective of any generative model can be expressed as a density-estimation problem:

$$p_\theta(x) \approx p_{\text{data}}(x),$$

where the goal is to infer parameters θthat minimize the discrepancy between the true data distribution and the model distribution. This discrepancy is typically measured through a divergence function such as the Kullback–Leibler (KL) divergence:

$$D_{\text{KL}}(p_{\text{data}}(x) || \, p_\theta(x)) = \mathbb{E}_{p_{\text{data}}(x)}\,[\ln \frac{p_{\text{data}}(x)}{p_\theta(x)}]$$

Minimizing D_{KL}ensures that the model assigns high probability to the regions of the input space where real data occur.

Generative AI models—such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Normalizing Flows, and Diffusion Models—apply different probabilistic principles to achieve this approximation. Each approach defines an explicit or implicit model of $p_\theta(x)$and uses probabilistic reasoning to perform inference, sampling, or divergence minimization.

2.1 Probabilistic Modelling and Latent Variables

Most real-world data are too complex to model directly. To address this, latent-variable models introduce hidden variables $z$that represent unobserved factors influencing the observed data x. The model thus defines a joint distribution

$$p_\theta(x, z) = p_\theta(x|z)\, p(z)$$

where $p(z)$is a prior over the latent space (often Gaussian), and $p_\theta(x \mid z)$is the conditional likelihood of the observed data given the latent representation. The marginal likelihood of data is then obtained by integrating out the latent variable:

$$p_\theta(x) = \int p_\theta(x|z)\, p(z)\, dz.$$

Since this integral is intractable in most cases, probability theory provides several approximate inference techniques—such as variational inference, Monte Carlo sampling, and importance weighting—to estimate or maximize $\log\, p_\theta(x)$.

2.2 Maximum Likelihood and Divergence Minimization

A common training objective for generative models is the Maximum Likelihood Estimation (MLE) principle:

$$\theta^* = \arg\max_\theta \mathbb{E}_{p_{\text{data}}(x)} [\log p_\theta(x)]$$

Maximizing the log-likelihood is equivalent to minimizing $D_{\text{KL}}(p_{\text{data}} \parallel p_\theta)$. This probabilistic objective guarantees asymptotic consistency: as data increases, $p_\theta(x)$converges to the true data distribution. However, computing $\log\, p_\theta(x)$often requires evaluating complex integrals or determinants, motivating alternative formulations such as variational lower bounds, adversarial objectives, or flow-based transformations. In VAEs, for instance, the intractable likelihood integral is replaced by an Evidence Lower Bound (ELBO):

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(z|x)}[\log\, p_\theta(x|z)] - D_{\text{KL}}(q_\phi(z|x) \parallel p(z)),$$

where $q_\phi(z|x)$approximates the true posterior $p_\theta(z|x)$. Maximizing $\mathcal{L}$corresponds to probabilistically reconstructing data while regularizing latent representations toward the prior $p(z)$.

2.3 Bayesian Foundations in Generative Learning

Bayesian inference offers a natural framework for learning in generative models by treating parameters as random variables with priors $p(\theta)$. Given data x, the posterior distribution is obtained via Bayes' theorem:

$$p(\theta \mid x) = \frac{p_\theta(x)\, p(\theta)}{p(x)}.$$

This formulation provides a principled mechanism for incorporating prior knowledge and quantifying uncertainty. Many modern approaches, such as Bayesian VAEs and probabilistic transformers, use Monte Carlo or variational approximations to represent the posterior distribution. The Bayesian perspective unifies learning and inference as probabilistic reasoning processes.

2.4 The Role of Probability in Generative Sampling

The generative process is mathematically equivalent to sampling from a learned probability distribution. Once a model has estimated $p_\theta(x)$, new data can be synthesized by first sampling from the latent prior and then generating from the conditional likelihood:

$$z \sim p(z), x \sim p_\theta(x \mid z).$$

This probabilistic two-step sampling defines the backbone of generative synthesis in VAEs and diffusion models. In implicit models like GANs, the generator learns a deterministic mapping $x = G_\theta(z)$that implicitly defines a distribution $p_\theta(x)$via the transformation of the latent prior through G_θ. Even in such non-explicit cases, probability theory governs the training process through divergence minimization (e.g., Jensen–Shannon divergence) and statistical equilibrium between the generator and discriminator.

2.5 Scope of This Paper

This paper aims to (1) analyze how probabilistic reasoning shapes the design of major generative AI models, (2) derive their central equations from fundamental probabilistic laws, and (3) connect them to broader theoretical principles such as variational inference, Markov processes, and stochastic differential equations. Later sections discuss explicit likelihood-based models (Section 3), adversarial and implicit probabilistic formulations (Section 4), and probabilistic dynamics in diffusion and score-based models (Section 5). The paper concludes with mathematical challenges and open problems regarding the probabilistic representation of creativity and uncertainty in artificial intelligence.

3. Probability Foundations in Explicit Likelihood-Based Generative Models

Explicit likelihood–based generative models directly specify a parametric probability distribution $p_\theta(x)$or its tractable approximation, and optimise model parameters by maximising the likelihood of observed data. These models embody the most transparent application of probability theory in generative AI, since they explicitly evaluate or approximate probability densities and use statistical principles such as maximum likelihood, expectation–maximisation, and variational inference.

Three major classes dominate this family: Variational Autoencoders (VAEs), Normalizing Flows (NFs), and Autoregressive Models. All rest upon core probabilistic ideas—latent variables, change-of-variables, and conditional factorisation of joint densities.

3.1 Variational Autoencoders (VAEs)

The Variational Autoencoder combines probabilistic modelling with neural-network parameterisation to learn an approximate posterior distribution over latent variables. A VAE assumes that each observation $x$arises from a latent variable $z$via a generative process defined by the joint distribution

$$p_\theta(x, z) = p_\theta(x \mid z)\, p(z),$$

where $p(z)$is a simple prior (usually $\mathcal{N}(0, I)$), and $p_\theta(x \mid z)$is the conditional likelihood modelled by a neural decoder.

Because direct computation of the marginal likelihood

$$p_\theta(x) = \int p_\theta(x \mid z)\, p(z)\, dz$$

is intractable, VAEs use an approximate posterior $q_\phi(z \mid x)$(the encoder) to estimate it through variational inference. Applying Jensen's inequality to $\ln\, p_\theta(x)$yields the Evidence Lower Bound (ELBO):

$$\ln\, p_\theta(x) = \ln \int q_\phi(z \mid x)\, \frac{p_\theta(x, z)}{q_\phi(z \mid x)}\, dz$$

$$\geq \mathbb{E}_{q_\phi(z|x)}[\ln\, p_\theta(x \mid z)] - D_{\mathrm{KL}}(q_\phi(z \mid x) \parallel p(z)) \equiv \mathcal{L}(\theta, \phi).$$

Maximising $\mathcal{L}$simultaneously encourages high data-likelihood under the decoder and closeness between the approximate posterior and the prior distribution. Probabilistically, the VAE objective can be understood as minimising the divergence between the approximate joint $q_\phi(x, z) = q_\phi(z \mid x)\, p_{\text{data}}(x)$and the model joint $p_\theta(x, z)$:

$$D_{\mathrm{KL}}(q_\phi(x, z) \parallel p_\theta(x, z)) = \text{const} - \mathbb{E}_{p_{\text{data}}(x)}[\mathcal{L}(\theta, \phi)].$$

This dual probabilistic interpretation makes the VAE one of the most direct bridges between information theory and generative learning.

3.2 Normalizing Flow Models

Normalizing Flows provide an exact likelihood framework by transforming a simple base distribution $p_z(z)$through a sequence of invertible, differentiable mappings f_i.

Let $x = f_\theta(z)$with $z \sim p_z(z)$; the resulting probability density is computed via the change-of-variables theorem:

$$p_\theta(x) = p_z(f_\theta^{-1}(x)) \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right|$$

In logarithmic form:

$$\log\, p_\theta(x) = log\ \ p_z(f_\theta^{-1}(x)) + \log \left| \det J_{f_\theta^{-1}}(x) \right|$$

where $J_{f_\theta^{-1}}(x)$is the Jacobian matrix of the inverse transformation.

Each flow step is designed so that both the inverse function and the determinant of the Jacobian are efficiently computable. By composing multiple flows, $x = f_K \circ f_{K-1} \circ \cdots \circ f_1(z)$, the model represents highly complex distributions while preserving tractable density evaluation.

Optimising parameters by maximum-likelihood estimation

$$\max_\theta \ \ \mathbb{E}_{p_{\text{data}}(x)}[\log p_\theta(x)]$$

ensures that the learned distribution approximates the empirical data distribution as closely as possible.

Normalizing Flows thus express the probability of observed data through explicit transformations of base distributions—each transformation controlled by the determinant term that encodes local volume change in probability space. This is a direct embodiment of differential probability calculus in deep learning.

3.3 Autoregressive and Sequential Models

Another probabilistic structure used in generative AI is the autoregressive factorisation of the joint distribution. For a data vector $x = (x_1, x_2, \ldots, x_n)$, probability theory provides an exact decomposition:

$$p_\theta(x) = \prod_{i=1}^{n} p_\theta\,(x_i | x_{<i}),$$

where $x_{<i}$denotes all preceding elements.This factorisation forms the theoretical backbone of models such as PixelRNN, WaveNet, and large language models like GPT.

Training is achieved by maximising the log-likelihood:

$$\mathcal{L}(\theta) = \mathbb{E}_{p_{\text{data}}(x)}\left[\sum_{i=1}^{n} log\ p_\theta(x_i|x_{<i})\right]$$

Each conditional probability is parameterised by a neural network that outputs a valid probability distribution (e.g., softmax for discrete tokens, Gaussian mixture for continuous data).Since the joint distribution is explicitly normalised, these models provide exact likelihoods and straightforward probabilistic sampling by sequentially drawing $x_i \sim p_\theta(x_i \mid x_{<i})$.

Autoregressive modelling highlights the role of probabilistic factorisation in managing high-dimensional data: probability theory enables a complex joint to be represented as a structured product of simpler conditional densities.

3.4 Comparative Probabilistic Insights

Although VAEs, Normalizing Flows, and Autoregressive Models differ architecturally, their probabilistic principles are unified. Each defines an explicit form of $p_\theta(x)$, satisfies the probability axioms (non-negativity and normalisation), and uses divergence minimisation or likelihood maximisation as its learning criterion.

- VAEs rely on approximate inference through variational bounds.
- Normalizing Flows employ exact transformations using Jacobian determinants.
- Autoregressive models exploit chain-rule decomposition for sequential prediction.

All three are direct computational embodiments of probability theory in high-dimensional representation learning.

3.5 Transition to Implicit and Adversarial Models

While explicit likelihood-based models adhere to analytical probability formulas, many successful modern generative models—such as Generative Adversarial Networks (GANs) and Diffusion Models—operate with implicit densities that cannot be expressed in closed form.
In these frameworks, probability theory shifts from explicit evaluation of $p_\theta(x)$to divergence estimation and probabilistic sampling through adversarial or stochastic processes.

4. Probability in Implicit and Adversarial Generative Models

While explicit likelihood-based models define probability densities in closed form, many of the most successful modern generative frameworks—such as Generative Adversarial Networks (GANs) and Energy-Based Models (EBMs)—are implicit probabilistic models. These systems do not provide a tractable expression for $p_\theta(x)$; instead, they define a stochastic generative process that samples from the underlying model distribution without explicitly computing its density.

Despite this apparent departure from classical density estimation, probability theory remains central: GANs are trained via divergence minimisation between distributions, and EBMs rely on probabilistic energy formulations that normalise through partition functions. This section develops the probabilistic principles governing adversarial learning, statistical divergence estimation, and related implicit modelling paradigms.

4.1 Implicit Probabilistic Modelling

An implicit model defines a stochastic mapping from a latent space $\mathcal{Z}$to data space $\mathcal{X}$:

$$z \sim p(z), \quad x = G_\theta(z),$$

where $p(z)$is a known prior (e.g., standard Gaussian) and $G_\theta$is a deterministic neural network. This mapping induces a probability distribution $p_\theta(x)$implicitly through the pushforward measure of $p(z)$:

$$p_\theta(x) = p(z)\ |\det \frac{\partial G_\theta^{-1}(x)}{\partial x}|\ ,\text{if } G_\theta \text{ is invertible.}$$

In practice, however, $G_\theta$is not invertible, making the density intractable. Probability theory then provides alternative means to align $p_\theta(x)$with $p_{\text{data}}(x)$via divergence minimisation.

4.2 Generative Adversarial Networks (GANs)

Introduced by Goodfellow et al. (2014), GANs cast generative learning as a two-player minimax game between a *generator*$G_\theta$and a *discriminator*D_ψ. The generator produces samples $G_\theta(z)$from a latent prior, while the discriminator attempts to distinguish between real and generated data.
The canonical objective function is derived from the Jensen–Shannon divergence (JSD) between the data and model distributions:

$$\min_\theta \max_\psi\ V(D_\psi, G_\theta) = \mathbb{E}_{x\sim p_{\text{data}}}[\log D_\psi(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_\psi(G_\theta(z)))]$$

Under an optimal discriminator $D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x)+p_\theta(x)}$, the value function reduces to

$$V(G_\theta) = -2\ln 2 + 2\, D_{\text{JS}}(p_{\text{data}} \parallel p_\theta)$$

Thus, minimising $V(G_\theta)$corresponds to minimising the Jensen–Shannon divergence—a symmetric measure of probabilistic distance between the true and generated distributions.
From a probabilistic standpoint, GAN training implicitly performs divergence estimation without explicitly evaluating densities, relying instead on the discriminator as a learned statistical estimator of distributional separability.

4.3 Alternative Divergences and the Probability–Distance Spectrum

The probabilistic perspective on GANs generalises naturally to other divergence metrics, leading to multiple GAN variants unified under the f-divergence framework.
Given convex $f(t)$, the f-divergence between distributions $p$and $q$is defined as:

$$D_f(p\|q) = \int q(x)\ f\left(\frac{p(x)}{q(x)}\right) dx$$

The Jensen–Shannon divergence is obtained when $f(t) = t\ln\ t - (t+1)\ln(t+1) + \ln\ 4$.
Recent research has shown that minimising f-divergences is equivalent to solving a variational estimation problem:

$$D_f(p\|q) = \sup_{T\in\mathcal{T}} (\mathbb{E}_{x\sim p}[T(x)] - \mathbb{E}_{x\sim q}[f^*(T(x))]),$$

where $f^*$is the convex conjugate of $f$and $T(x)$plays the role of the discriminator.
In Wasserstein GANs (WGANs), the Jensen–Shannon divergence is replaced by the Earth Mover's (Wasserstein-1) distance:

$$W(p_{\text{data}}, p_\theta) = \inf_{\gamma\in\Pi(p_{\text{data}},p_\theta)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|],$$

where $\Pi(p_{\text{data}}, p_\theta)$is the set of joint couplings with given marginals.By Kantorovich–Rubinstein duality, this distance admits a variational form:

$$W(p_{\text{data}}, p_\theta) = \sup_{\|f\|_L\le 1} \left(\mathbb{E}_{x\sim p_{\text{data}}}[f(x)] - \mathbb{E}_{x\sim p_\theta}[f(x)]\right)$$

where $f$is constrained to be 1-Lipschitz.This probabilistic distance provides improved stability and a meaningful geometric interpretation, grounding adversarial training firmly in measure theory.

4.4 Energy-Based and Implicit Density Models

An Energy-Based Model (EBM) defines an unnormalised probability density via an energy function $E_\theta(x)$:

$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z_\theta},$$

where $Z_\theta = \int e^{-E_\theta(x)}dx$is the partition function ensuring normalisation.
Minimising the negative log-likelihood yields:

$$\nabla_\theta \mathcal{L} = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\nabla_\theta E_\theta(x)] - \mathbb{E}_{x\sim p_\theta(x)}[\nabla_\theta E_\theta(x)]$$

This equation mirrors the gradient update in GANs: the model learns to decrease energy (increase probability) for real samples and increase energy (decrease probability) for generated ones.
Sampling from EBMs often relies on Markov Chain Monte Carlo (MCMC) techniques such as Langevin dynamics, which are themselves derived from probability theory through stochastic differential equations:

$$x_{t+1} = x_t - \frac{\epsilon^2}{2}\nabla_x E_\theta(x_t) + \epsilon\,\eta_t, \qquad \eta_t \sim \mathcal{N}(0, I).$$

This update rule describes diffusion of probability mass guided by the energy gradient and stochastic noise—a probabilistic mechanism that parallels gradient-based learning in GANs.

4.5 Probabilistic Interpretation of Adversarial Equilibrium

At optimality, adversarial learning achieves a Nash equilibrium between the generator and discriminator distributions:

$$p_{\theta^*}(x) = p_{\text{data}}(x),$$

such that neither the generator nor the discriminator can improve their objective without changing the other's parameters.From a probabilistic viewpoint, this equilibrium corresponds to equality of expected log-probabilities:

$$\mathbb{E}_{p_{\text{data}}(x)}[\ln\ D^*(x)] = \mathbb{E}_{p_\theta(x)}[\ln(1 - D^*(x))].$$

This state represents the convergence of the model to the true data distribution—a stochastic realisation of maximum entropy subject to data constraints.

4.6 Limitations and Theoretical Challenges

Despite their empirical success, adversarial models pose unresolved probabilistic issues. The lack of explicit density estimation precludes direct evaluation of $p_\theta(x)$or likelihood-based metrics, complicating theoretical analysis. Moreover, training instability, mode collapse, and non-convex loss surfaces arise from

imperfect divergence approximation and high-variance gradient estimates. Mathematically, the non-overlapping supports of $p_{\text{data}}$and $p_\theta$can make the Jensen–Shannon divergence ill-defined, motivating the shift toward Wasserstein and energy-based formulations.

5. Probabilistic Foundations of Diffusion and Score-Based Generative Models

Diffusion and score-based generative models represent one of the most profound probabilistic advances in modern generative AI. These models define a stochastic process that gradually transforms a simple noise distribution into complex structured data through the probabilistic principles of Markov processes, stochastic differential equations (SDEs), and score matching.Unlike GANs, which implicitly approximate probability distributions via adversarial training, diffusion models construct them explicitly by learning the gradients of log-densities—known as scores—of intermediate noisy distributions.This section explores the probabilistic foundations of such models, deriving their central equations and demonstrating how they operationalize key ideas from probability theory and statistical physics.

5.1 The Forward Diffusion Process

The diffusion process begins by progressively corrupting data with Gaussian noise over discrete time steps. Let $x_0 \sim p_{\text{data}}(x)$denote a data sample. The forward (noising) process defines a Markov chain:

$$q(x_t|x_{t-1}) = \mathcal{N}\,(x_t; \sqrt{1-\beta_t}\,x_{t-1}, \beta_t I), t = 1,2, \dots, T,$$

where $\{\beta_t\}_{t=1}^T$is a variance schedule controlling the noise magnitude. By iteratively applying this process, the data distribution $q(x_0)$converges to an isotropic Gaussian $\mathcal{N}(0, I)$as $t \to T$.

Using properties of Gaussian distributions, the marginal of the noising process can be expressed directly as:

$$q(x_t|x_0) = \mathcal{N}\,(x_t; \sqrt{\bar{\alpha}_t}\,x_0, (1-\bar{\alpha}_t)I),$$

where $\alpha_t = 1 - \beta_t$and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$

This defines a sequence of intermediate distributions $q(x_t)$over time, which together form a probabilistic diffusion trajectory between data and noise.

5.2 The Reverse Diffusion Process

The generative (denoising) process seeks to invert this diffusion by gradually reconstructing clean samples from pure noise. Since the forward process is Markovian, the true reverse transitions also form a Markov chain with conditionals $p_\theta(x_{t-1} \mid x_t)$:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\,(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

where the mean and variance functions are parameterised by neural networks.

The goal of training is to approximate the posterior reverse conditional$q(x_{t-1} \mid x_t, x_0)$, which is also Gaussian with closed-form mean:

$$\tilde{\mu}_t(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_t\right),$$

where $\epsilon_t$represents the Gaussian noise added during the forward process.

Thus, the reverse model learns to predict and remove the noise at each step, effectively denoising the sample while preserving probabilistic consistency.

5.3 Variational Objective and Likelihood Derivation

The training objective for diffusion models is derived from variational inference applied to the data likelihood. The marginal likelihood of data is:

$$\log\, p_\theta(x_0) = \log \int p_\theta(x_{0:T})\, dx_{1:T},$$

where $p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^T p_\theta(\,x_{t-1} \mid x_t\,)$and $p(x_T) = \mathcal{N}(0, I)$.

Applying the variational lower bound yields:

$$\log\, p_\theta(x_0) \geq \mathbb{E}_{q(x_{1:T}|x_0)} \log\left[\frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right] \equiv -\mathcal{L}_{\text{VLB}}.$$

After simplification, this results in a tractable objective:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon\sim\mathcal{N}(0,I)}[\| \epsilon - \epsilon_\theta(x_t, t) \|^2],$$

where the neural network $\epsilon_\theta$learns to predict the injected Gaussian noise at each timestep. Minimising this loss corresponds to maximising a lower bound on the data log-likelihood—a purely probabilistic objective.

5.4 Continuous-Time Formulation and Stochastic Differential Equations

Recent score-based diffusion models reinterpret the discrete diffusion chain as a stochastic differential equation (SDE) describing the time evolution of the probability density $p_t(x)$.

The forward SDE (perturbation) is given by:

$$dx = f(x,t)\,dt + g(t)\,dW_t,$$

where $W_t$is a Wiener process, $f(x,t)$is the drift coefficient, and $g(t)$is the diffusion coefficient controlling noise magnitude.

The corresponding reverse-time SDE, derived from the Fokker–Planck equation, expresses the backward evolution of probability:

$$dx = [f(x,t) - g(t)^2 \nabla_x \log\, p_t(x)]dt + g(t)\,d\bar{W}_t,$$

where $\nabla_x \log\, p_t(x)$is the score function, representing the gradient of the log-density with respect to x.

By learning the score function $s_\theta(x,t) \approx \nabla_x \log\, p_t(x)$, one can simulate the reverse SDE and thus generate new samples from the target distribution.This connection between stochastic calculus and generative modelling exemplifies the deep integration of probability theory and differential equations in AI.

5.5 Score Matching and Denoising Interpretation

The idea of score matching, introduced by Hyvärinen (2005), provides a probabilistic framework for learning unnormalised densities. It minimises the Fisher divergence between the model score and the true data score:

$$\mathcal{L}_{\text{score}}(\theta) = \frac{1}{2}\,\mathbb{E}_{x\sim p_{\text{data}}(x)}[\,\|\,\nabla_x \log\, p_\theta(x) - \nabla_x \log\, p_{\text{data}}(x)\,\|^2].$$

In diffusion models, the score function is trained over progressively noisier versions of data x_t, leading to the **Denoising Score Matching (DSM) objective**:

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}_{t,x_0,\epsilon}\,[\lambda(t)\;\|\,s_\theta(x_t,t) + \epsilon/\sigma_t\,\|^2],$$

where $\sigma_t$controls noise variance and $\lambda(t)$is a weighting function.Minimising this loss enables the model to estimate the gradient of the log-probability at every noise level, providing the foundation for efficient sampling via reverse diffusion.

5.6 Sampling and Probabilistic Generation

Once the model learns the score $s_\theta(x,t)$, new data can be generated by simulating the reverse SDE:

$$dx = [f(x,t) - g(t)^2 s_\theta(x,t)]dt + g(t)\,d\bar{W}_t.$$

This iterative stochastic process reconstructs data from Gaussian noise, gradually restoring high-probability structure under the learned distribution.

In practice, deterministic approximations such as the probability flow ODE:

$$\frac{dx}{dt} = f(x,t) - \frac{1}{2} g(t)^2 s_\theta(x,t)$$

are also used, offering exact likelihood computation under the same probabilistic dynamics.

The generative process thus becomes a continuous probabilistic transformation between distributions, grounded in the mathematical equivalence between diffusion dynamics and density evolution in stochastic systems.

5.7 Theoretical Insights

Diffusion and score-based models unify multiple probabilistic concepts:

- Markov chains model conditional independence between noise levels.
- Stochastic calculus connects diffusion processes to continuous probability flows.
- Bayesian inference appears implicitly through the estimation of posteriors $p(x_0 \mid x_t)$.
- Energy-based modelling reemerges as the learned score corresponds to the negative gradient of an implicit energy landscape.

Thus, probability theory does not merely support diffusion models—it defines their entire architecture, training objective, and sampling mechanism. The precision with which these models approximate complex data distributions reaffirms probability as the natural language of generative intelligence.

Challenges, Open Problems, and Future Directions

Although probability theory forms the backbone of modern generative AI, several mathematical and computational challenges persist. One key limitation lies in the difficulty of evaluating or normalising complex high-dimensional probability distributions. In models such as Variational Autoencoders and Diffusion Models, the marginal likelihood $p_\theta(x) = \int p_\theta(x \mid z)p(z)\,dz$or the reverse-time conditional $p_\theta(x_{t-1} \mid x_t)$remains analytically intractable, forcing reliance on variational or Monte Carlo approximations. This introduces estimation bias and complicates convergence proofs. Another challenge arises from non-convex optimisation surfaces inherent in probabilistic divergence minimisation; the objectives based on D_{KL}or Jensen–Shannon divergence may yield multiple equilibria, often leading to unstable training or mode collapse.

From a computational standpoint, generative probability models scale poorly as dimensionality increases, since evaluating Jacobian determinants or solving stochastic differential equations such as

$$dx = [f(x,t) - g(t)^2 \nabla_x \log\, p_t(x)]\,dt + g(t)\,dW_t$$

is expensive for large-scale data. Furthermore, the theoretical understanding of generalisation in probabilistic generators is incomplete—while these models can approximate $p_{\text{data}}(x)$well, formal bounds on likelihood estimation error or sample quality remain open research questions. Recent efforts attempt to unify explicit and implicit probabilistic learning through hybrid systems combining variational inference, adversarial training, and diffusion dynamics, suggesting a direction where the strengths of each probabilistic paradigm may be merged.

Future progress will depend on bridging probability theory with geometry and physics, allowing models to learn not only densities but also structural invariants of data. Quantum-inspired probabilistic frameworks and information-theoretic regularisation may offer improved tractability and theoretical guarantees. In essence, the next generation of generative AI will continue to rely on probability—not just as a modelling tool but as a universal mathematical language for reasoning about uncertainty, transformation, and creativity in artificial intelligence.

Conclusion

Probability theory provides the mathematical skeleton of every modern generative AI system. Whether through explicit likelihood estimation, variational inference, adversarial divergence minimisation, or stochastic differential equations, the essence of generation is probabilistic reasoning about data distributions. The paper has shown how latent-variable models like Variational Autoencoders express joint probabilities $p_\theta(x,z) = p_\theta(x \mid z)p(z)$, how adversarial frameworks implicitly minimise divergences between $p_{\text{data}}$and p_θ, and how diffusion and score-based models reconstruct data by solving reverse stochastic processes.

Collectively, these architectures demonstrate that the act of "creating" data synthetically is equivalent to sampling from a learned probabilistic manifold. Probability not only governs model training and sampling but also connects learning dynamics to fundamental concepts such as entropy, uncertainty, and information flow. The remaining theoretical challenge is to unify explicit and implicit probabilistic paradigms under a single framework that preserves both tractability and expressiveness. As generative AI continues to expand—from text-to-image synthesis to autonomous creativity—the mathematical future of the field will depend on deeper integration between probability theory, geometry, and computational physics.

References

[1]. Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein GAN.* arXiv:1701.07875.
[2]. Bishop, C. M. (2006). *Pattern recognition and machine learning.* Springer.
[3]. Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). Wiley.
[4]. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems, 27*, 2672–2680.
[5]. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems, 33*, 6840–6851.
[6]. Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research, 6*, 695–709.
[7]. Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR).*
[8]. Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *Proceedings of the 31st International Conference on Machine Learning*, 1278–1286.
[9]. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. *Proceedings of the 32nd International Conference on Machine Learning*, 2256–2265.
[10]. Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems, 32*, 11895–11907.
[11]. Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation, 23*(7), 1661–1674.