# Basic Architecture of a Generative Model

Understanding how artificial intelligence learns to create new data – images, text, and much more.

Part of the "Introduction to Generative AI" module

# What is a Generative Model?

### Machine Learning Model

Creates new, realistic data similar to what it has learnt from training examples

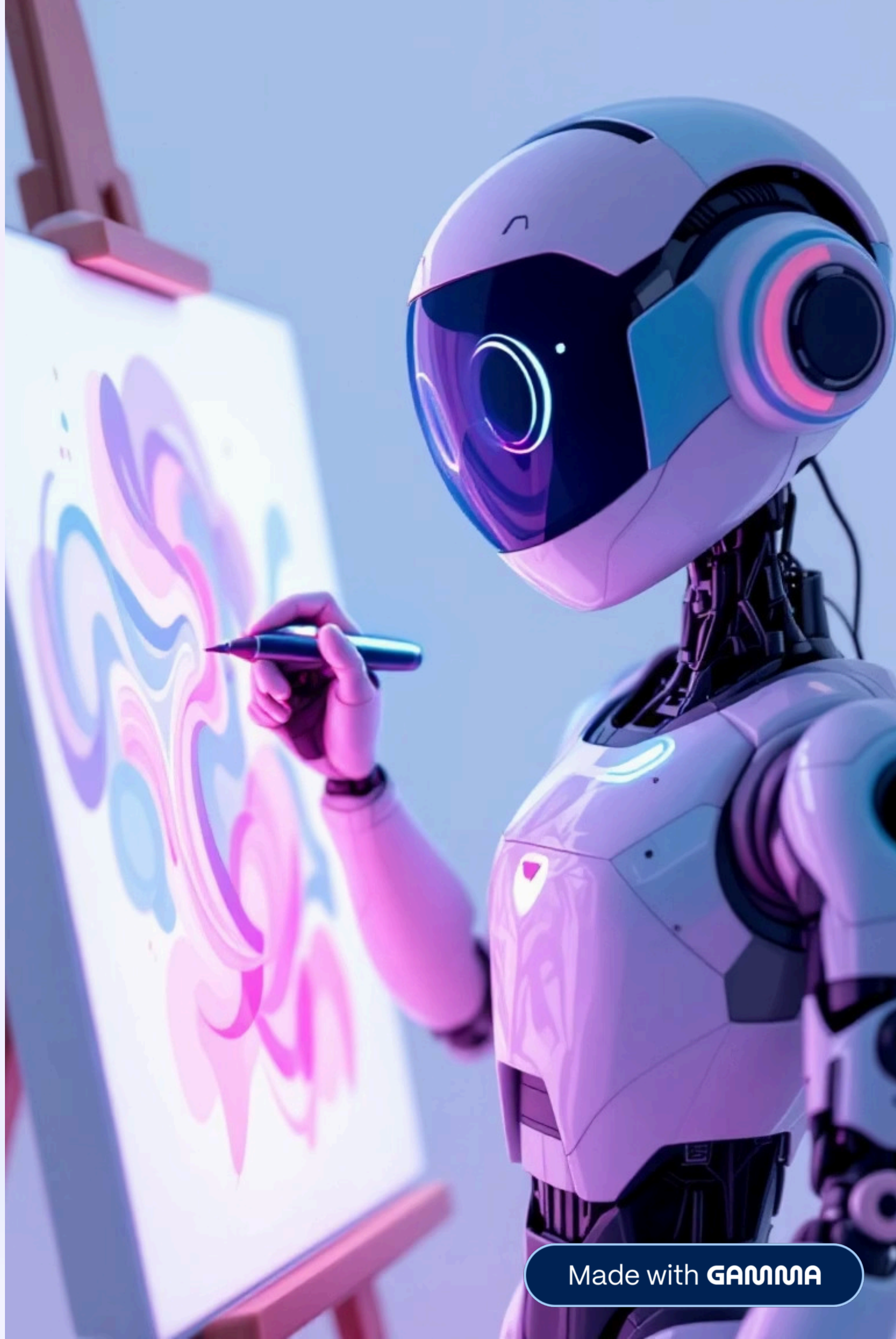
### Data Distribution Learning

Instead of predicting labels, it learns the underlying patterns in data

### Creative Output

Trained on 10,000 cat images → generates a brand new cat image that looks completely real

**Goal:** Learn to "imitate" data patterns and create new variations that feel authentic.



# Core Idea Behind Generative Models

### Pattern Recognition

Generative models **capture intricate patterns** present in existing datasets through sophisticated learning algorithms.

### Data Sampling

Once trained, they can **sample new data points** from the learned distribution with remarkable accuracy.

### Mathematical Foundation

They learn **P(X)** — the probability distribution of how data points naturally occur in the real world.

**Example:** AI learns what "faces" look like through thousands of examples → then generates new, realistic faces that have never existed before.

# General Architecture Overview

### Latent Space

**Function:** Compressed internal representation of data

**Analogy:** AI's imagination space where ideas form

### Generator / Decoder

**Function:** Converts simple input noise into meaningful data

**Analogy:** An artist creating masterpieces from rough sketches

### Encoder

**Function:** Maps real data to latent space (optional component)

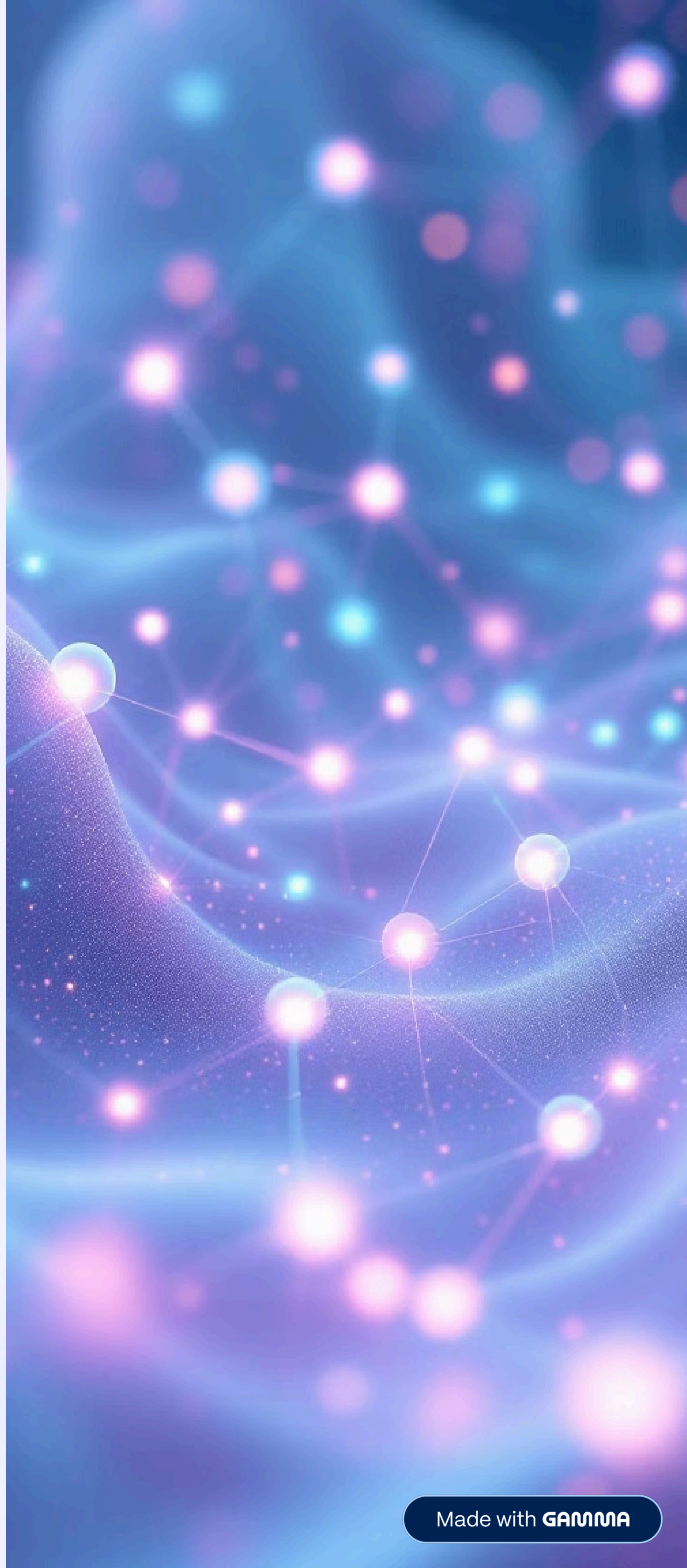
**Analogy:** An analyser summarising complex data patterns

### Loss Function

**Function:** Evaluates how close generated data is to real data

**Analogy:** A teacher providing continuous feedback for improvement

Together, these components form the **generative learning loop** that enables AI creativity.



# Step-by-Step Data Flow

01

### Input Stage

Random numbers or real data samples serve as the starting point for generation

02

### Encoder Processing

Optional component that compresses input into meaningful latent features

03

### Latent Space

Represents the "essence" of the data in a compressed, learnable format

04

### Generator/Decoder

Transforms latent vectors into realistic, high-quality data outputs

05

### Loss Evaluation

Measures and continuously improves output realism through iterative feedback

**Goal:** The generated output should be completely indistinguishable from authentic real data.

# Example Architectures of Generative Models

| Model Type | Main Components | Training Mechanism | Output Example |
| --- | --- | --- | --- |
| **GAN** | Generator + Discriminator | Adversarial competition | Realistic images |
| **VAE** | Encoder + Decoder | Reconstruction + KL loss | Data reconstruction |
| **Autoregressive** | Sequence model | Next-step prediction | Text, audio |
| **Diffusion** | Noise + Denoising | Step-by-step denoising | High-quality images |

# Architecture 1 – GAN (Generative Adversarial Network)

## Key Components

- **Generator:** Creates convincing fake data from random noise input
- **Discriminator:** Acts as a judge, determining whether data is real or artificially generated
- **Training Process:** Both components compete intensively until generator successfully "fools" the discriminator

## Architecture Flow

`Noise (z) → Generator → Fake Data → Discriminator → Real/Fake`

`Real Data → Discriminator → Real/Fake`

**Primary Uses:** Image generation, artistic style transfer, realistic face synthesis.

# Architecture 2 – VAE (Variational Autoencoder)

1

### Encoder

Compresses input data into mean and variance of latent variables

2

### Sampling

Generates latent vector from learned distribution

3

### Decoder

Reconstructs original data from compressed latent representation

**Loss Function:** Combines *Reconstruction Loss + KL Divergence* for optimal learning

**Applications:** Data compression, image reconstruction, synthetic data generation for research.

# Other Popular Architectures

### Autoregressive Models (GPT, PixelRNN)

Predicts the next token or pixel based on all previous elements in the sequence.

Excels at text and audio generation tasks.

$x_1 \rightarrow x_2$ (depends on $x_1$) $\rightarrow x_3$ (depends on $x_1, x_2$) $\rightarrow \ldots$

### Diffusion Models (DALL·E 3, Stable Diffusion)

Adds noise step-by-step to training data, then learns to systematically denoise.

Generates exceptionally high-quality images through iterative refinement.

Data → +Noise → Pure Noise → Denoising → Generated Data

# Summary

### GAN

**Key Feature:** Adversarial training through competition

**Example Use:** Photo-realistic image generation

### VAE

**Key Feature:** Latent representation learning

**Example Use:** Data compression and generation

### Autoregressive

**Key Feature:** Sequential modelling approach

**Example Use:** Text and speech generation

### Diffusion

**Key Feature:** Stepwise noise removal

**Example Use:** High-resolution image synthesis

> **Key Takeaway:** All generative models learn the **underlying structure** of data to create **new, realistic outputs** — each architecture achieves this goal through different innovative approaches.