

# Week 0e: Generative AI

## The Creation Challenge

Machine Learning for Smarter Innovation

BSc-Level Course

October 6, 2025

# Outline

# The Creation Challenge

Moving Beyond Classification

## Traditional ML: “What is this?”

- Email spam detector: Classify existing emails
- Medical diagnosis: Analyze X-ray images
- Sentiment analysis: Judge customer reviews

**Limitation:** Only analyzes, never creates

Fundamental shift: from pattern recognition to content generation

## Generative AI: “Create something new”

- Generate phishing emails for security training
- Synthesize medical images for rare diseases
- Write product descriptions automatically
- Compose music for video backgrounds

**Power:** Creation enables innovation

# Mathematical Foundation

## Two Approaches to Learning

### Discriminative Models

Learn:  $P(y|x)$  - Conditional probability

#### What it does:

- Given  $x$ , predict label  $y$
- Learns decision boundaries
- Divides input space

Examples: Logistic, RF, SVM

Can sample new  $x$ ? NO - only classifies existing data

### Generative Models

Learn:  $P(x)$  - Joint or marginal distribution

#### What it does:

- Models entire data distribution
- Samples new  $x \sim P(x)$
- Creates novel instances

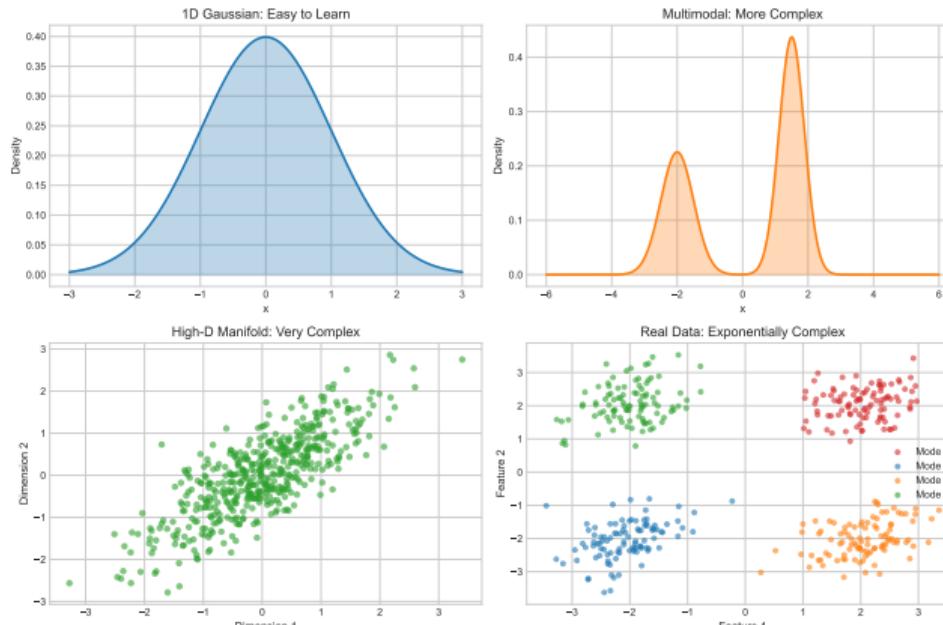
Examples: VAEs, GANs, Diffusion

Can sample new  $x$ ? YES - generates from distribution

Key distinction: Discriminative draws boundaries, Generative learns distributions enabling sampling

# The Hard Problem

Why Generation is Fundamentally Difficult



## Challenges:

- High-dimensional spaces
- Multimodal distributions

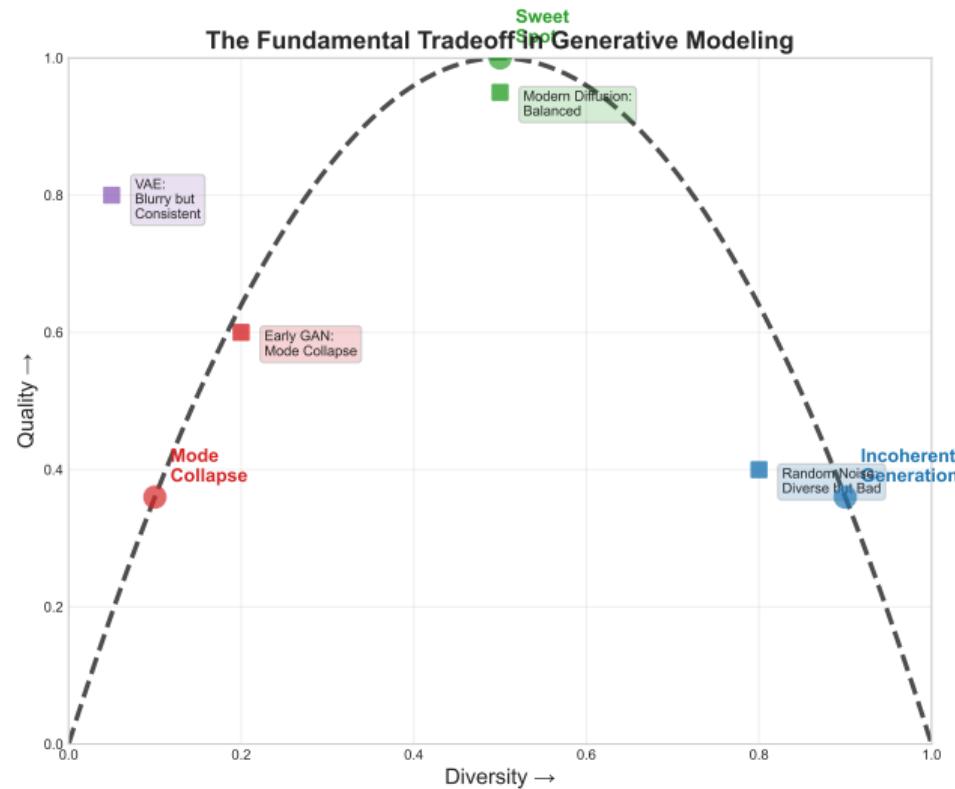
Real data lives on complex manifolds - learning full distribution is exponentially hard

## Requirements:

- Capture all patterns
- Maintain realism

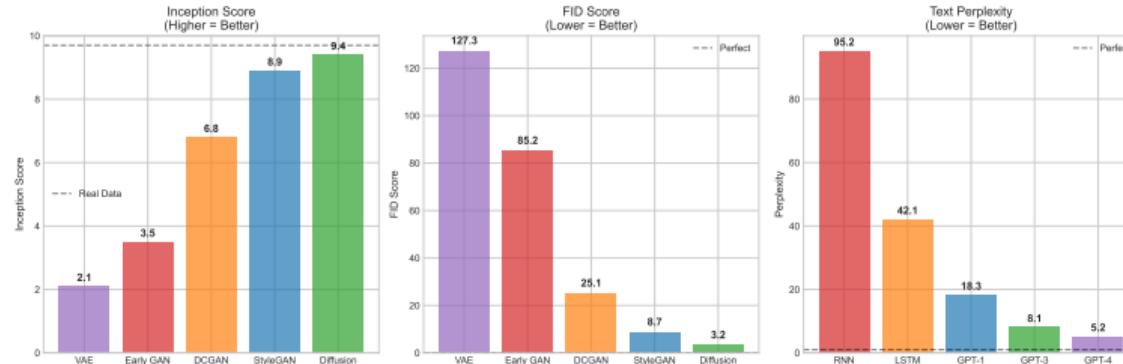
# The Fundamental Tradeoff

Quality vs Diversity Dilemma



# Measuring Generation Quality

## Metrics for Evaluating Generative Models



### Inception Score (IS)

- Range: 1-1000
- Higher = better
- Quality & diversity

#### Interpretation:

- >300: Excellent
- 100-300: Good
- <100: Poor

### FID Score

- Range: 0-500
- Lower = better
- Feature distance

#### Interpretation:

- <10: Photorealistic
- 10-50: Good quality
- >50: Noticeable artifacts

### Perplexity (Text)

- Range: 1-10,000
- Lower = better
- Predictability

#### Interpretation:

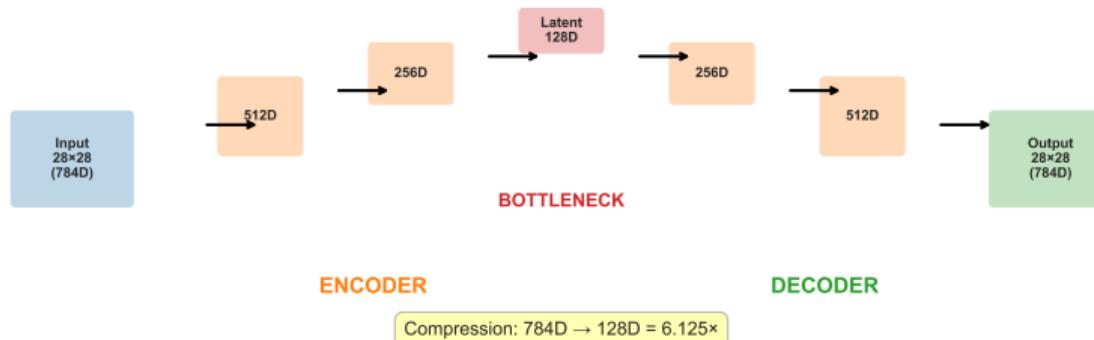
- <20: Human-like
- 20-100: Coherent
- >100: Gibberish

Quantitative metrics enable objective quality assessment and model comparison

# Autoencoders: The Foundation

## Learning Compressed Representations

Autoencoder Architecture: Compression Through Reconstruction



### Encoder

- 784D  $\rightarrow$  128D
- $z = f_{enc}(x)$

### Why compress?

- Forces selective encoding
- Filters noise

### Latent (Bottleneck)

- 128D representation
- Key features only
- Compressed 6.1x

### Bottleneck forces:

- Information prioritization

### Decoder

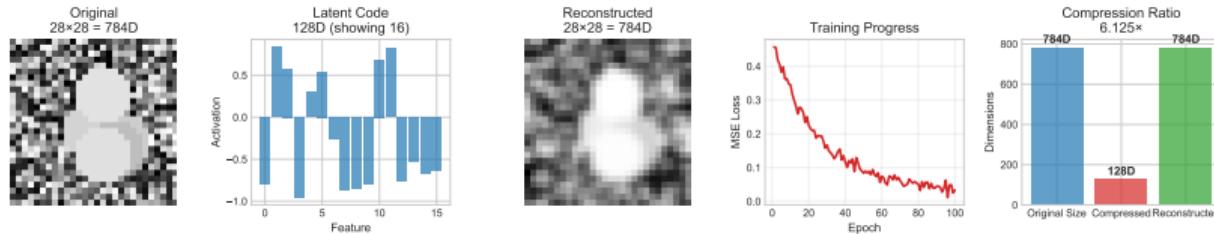
- 128D  $\rightarrow$  784D
- $\hat{x} = f_{dec}(z)$

### Reconstruction:

- Lossy process
- Preserves essentials

# Worked Example: MNIST Compression

From 784 Pixels to 128 Features



## Architecture:

- Input: 784 pixels
- Encoder:  $784 \rightarrow 128$
- Decoder:  $128 \rightarrow 784$

## Training:

- Loss:  $L = \|x - \hat{x}\|^2$
- Optimizer: Adam
- Compression: 6.125x

MSE drops 0.45  $\rightarrow$  0.03 over 100 epochs

# Autoencoder Successes

What Works Well

Autoencoder Successes

Visualization Placeholder

(Chart 12)

## [+] SUCCESSES:

- Dimensionality reduction: 784D  $\rightarrow$  128D

Machine Learning for Smarter Innovation (BSc-Level Course)

## Quantitative Results:

- MSE: 0.031, Compression: 6.125x

Week 0e: Generative AI

October 6, 2025

# Autoencoder Limitations

## The Generation Problem

Autoencoder Failures  
Visualization Placeholder  
(Chart 13)

### [−] FAILURES:

- Blurry outputs (averaging)

# Root Cause Analysis

Why Autoencoders Generate Poorly

Averaging Problem

Visualization Placeholder

(Chart 14)

## The Averaging Problem:

- Loss:  $L = \|x - \hat{x}\|^2$

## Mathematical Insight:

- $\hat{x} = \arg \min E[\|x - \hat{x}\|^2]$

# Variational Autoencoders (VAEs)

The Probabilistic Solution

Vae Framework

Visualization Placeholder

(Chart 15)

## Key Innovation:

- Encode to distribution:  $q_\phi(z|x) = \mathcal{N}(\mu, \sigma^2)$
- Sample:  $z = \mu + \sigma \odot \epsilon$

## Reparameterization Trick:

- Can't backprop through sampling
- Make  $z$  deterministic function
- Gradient flows through  $\mu, \sigma$

## VAE Loss (ELBO):

$$\mathcal{L} = -E[\log p(x|z)] + KL(q||p)$$

## Two terms:

- Reconstruction: Decode accurately
- KL: Keep  $z$  smooth, continuous
- $\beta$ -VAE:  $\beta \times KL$  for balance

# Human Learning Analogy

How Artists Develop Mastery

Artist Learning Process

Visualization Placeholder

(Chart 16)

## Traditional Art Education:

- Student creates artwork

## Key Insights:

- Adversarial feedback drives improvement

# Two Revolutionary Approaches

Beyond VAEs to Better Generation

Two Approaches

Visualization Placeholder

(Chart 17)

## Approach 1: Adversarial

- Two networks compete

## Approach 2: Diffusion

- Iterative denoising

# GANs: The Forger vs Detective Game

Adversarial Training in Plain English

Forger Detective Analogy

Visualization Placeholder

(Chart 18)

## Forger (Generator):

- Creates fakes from noise

## Detective (Discriminator):

- Examines: real or fake?

# Diffusion: The Reverse Corruption Process

Denoising in Plain English

Reverse Corruption Analogy

Visualization Placeholder

(Chart 19)

## Forward (Corruption):

- Clean image - $\downarrow$  pure noise

## Reverse (Generation):

- Pure noise - $\downarrow$  clean image

# GAN Dynamics: Geometric View

Understanding the Adversarial Process

Gan Geometric Dynamics

Visualization Placeholder

(Chart 20)

## Generator:

- Maps noise  $z$  to data  $x$

## Discriminator:

- Separates real from fake

# GAN Training: Step-by-Step Example

Real Loss Values from MNIST Training

Gan Training Walkthrough

Visualization Placeholder

(Chart 21)

**Epoch 1:**

● D\_loss: 1.386

**Epoch 100:**

● D\_loss: 0.695



# Diffusion Mathematical Framework

## Forward and Reverse Processes

Diffusion Mathematics

Visualization Placeholder

(Chart 22)

### Forward:

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

### Noise Schedule:

- Linear: 0.0001 - 0.02
- Cosine: Variable rate
- Matters: Smooth degradation

Linear noise schedule works for most cases

### Reverse:

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

### Training:

$$L = E[||\epsilon - \epsilon_\theta(x_t, t)||^2]$$

**Intuition:** Predict noise, subtract it

# Latent Space Interpolation

Smooth Transitions in Generated Content

Latent Interpolation

Visualization Placeholder

(Chart 23)

## GAN Interpolation:

- Sample  $z_1, z_2 \sim \mathcal{N}(0, I)$

## Applications:

- Style transfer, face morphing

# Diffusion Denoising Visualization

From Noise to Image in 1000 Steps

Denoising Steps

Visualization Placeholder

(Chart 24)

## Key Time Steps:

- T=1000: Pure noise

## Process Control:

- Guidance scale

# Why Adversarial Training Works

The Mathematical Guarantee

Adversarial Theory

Visualization Placeholder

(Chart 25)

## Theory:

- Minimax convergence

## Benefits:

- Sharp, realistic images

# Experimental Validation

## Quality Metrics vs Training Progress

Quality Metrics Over Time

Visualization Placeholder

(Chart 26)

### Results (MNIST):

Method	IS	FID	Time
Random	1.0	500	-
VAE	5.2	48	30min
GAN	9.1	9	2hr
Diffusion	9.3	3	8hr
Real	9.7	0	-

### Observations:

- Diffusion: Best quality
- GAN: 4x faster, nearly as good
- VAE: Fast but blurry

### Patterns:

- VAE: Monotonic
- GAN: Oscillates

# Implementation: Stable Diffusion API

Production-Ready Generative AI

Stable Diffusion Api

Visualization Placeholder

(Chart 27)

## Basic Usage:

```
import requests  
  
response = requests.post(  
    api_url,  
    headers={"Authorization": key},
```

## Parameters:

- cfg\_scale: Adherence (1-20)
- steps: Quality (10-150)
- seed: Reproducible

Cost: \$0.004 per image

# The Generative AI Landscape

## Four Fundamental Approaches

Generative Landscape  
Visualization Placeholder  
(Chart 28)

**VAEs:** Probabilistic, smooth latent, blurry  
**GANs:** Adversarial, sharp outputs, unstable

Each approach has unique strengths - modern systems combine techniques

**Diffusion:** Iterative denoising, high quality, slow  
**Transformers:** Sequential, excellent text, scalable

# Choosing Your Generative Model

Decision Framework for Practitioners

## Decision Criteria:

### 1. What are you generating?

- Images: Diffusion or GAN
- Text: Transformer (GPT family)
- Structured data: VAE
- Multimodal: Diffusion + Transformer

### 2. Data size?

- < 10k samples: VAE (stable)
- 10k-100k: GAN or VAE
- > 100k: Diffusion or Transformer

### 3. Priority?

- Quality: Diffusion (FID  $\downarrow$  5)
- Speed: GAN (single pass)
- Stability: VAE (always converges)
- Control: Diffusion (guidance)

## Recommendation Table:

Use Case	Best	Why
Photorealistic	Diffusion	Quality
Fast prototype	GAN	Speed
Data augment	VAE	Stable
Text gen	Transformer	Sequential
Style transfer	VAE	Interpolate
Research	VAE	Interpret

## When NOT to Use:

- VAE: Need sharp images
- GAN: Limited data, need stability
- Diffusion: Real-time inference required
- All: Insufficient compute resources

Model selection requires balancing quality, speed, stability against problem constraints

# Common Pitfalls: What Can Go Wrong

## Failure Modes and Solutions

### VAE Pitfalls

#### 1. Posterior Collapse

- KL  $\rightarrow 0$
- Fix:  $\beta$ -VAE, warm-up

#### 2. Blurry

- MSE averages
- Fix: Perceptual loss

### GAN Pitfalls

#### 1. Mode Collapse

- Limited variety
- Fix: Minibatch disc

#### 2. Unstable

- Oscillates
- Fix: Wasserstein, spectral norm

### Diffusion Pitfalls

#### 1. Slow (1000 steps)

- Latency issue
- Fix: DDIM (50 steps)

#### 2. Memory

- High-res costly
- Fix: Latent diffusion

Each approach has characteristic failure modes with specific solutions

# Generative AI Best Practices

From Research to Production

## Training:

### 1. Start Simple

- Low res first (64x64 before 1024x1024)
- Validate on toy datasets

### 2. Monitor Obsessively

- Log every 100 steps
- Visual sample inspection
- Track FID/IS

### 3. Use Pretrained

- Transfer learning saves weeks
- Fine-tune Stable Diffusion

### 4. Ablation Studies

- Test components independently

## Deployment:

### 1. Quality Control

- Human-in-the-loop review
- Content filtering
- Watermarking

### 2. Performance

- Quantization (FP16, INT8)
- Distillation for speed
- Caching

### 3. Safety

- Rate limiting
- Content moderation
- Prompt injection defenses

### 4. Continuous Improvement

- User feedback
- A/B testing

Production requires systematic validation and continuous monitoring

# Comprehensive Trade-offs

No Free Lunch in Generative Modeling

Generative Tradeoffs

Visualization Placeholder

(Chart 29)

## Stability:

- VAEs, Diffusion: Stable
- GANs: Unstable

## Speed:

- VAEs, GANs: Fast
- Diffusion: Slow

Choose based on requirements

## Quality:

- Diffusion, GANs: Excellent
- VAEs: Blurry

## Control:

- Diffusion, Transformers: High
- GANs: Limited

# State-of-the-Art Applications

## Production Generative AI Systems

Modern Applications

Visualization Placeholder

(Chart 30)

### Image:

- DALL-E 3, Midjourney
- Stable Diffusion, Firefly
- 1024x1024, 10-30 sec

Production systems achieve human-level performance

### Text:

- GPT-4, Claude, Gemini
- Llama 2 (open)
- 32k-200k tokens, 100+ languages

# Summary & Future of Generative AI

What We Learned and What's Next

Ethics Summary

Visualization Placeholder

(Chart 31)

## What We Learned:

- Autoencoders: Compress but blurry
- VAEs: Probabilistic latent space
- GANs: Adversarial for realism
- Diffusion: Best quality, slow
- When to use each approach
- Common pitfalls and solutions

## Ethical Challenges:

- Deepfakes, misinformation
- Copyright, attribution
- Bias amplification
- Worker displacement

## Solutions:

- Watermarking, detection