Week 0e: Generative AI
The Creation Challenge

Machine Learning for Smarter Innovation

BSc-Level Course

October 6, 2025

Outline

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

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

Fundamental shift: from pattern recognition to content generation

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
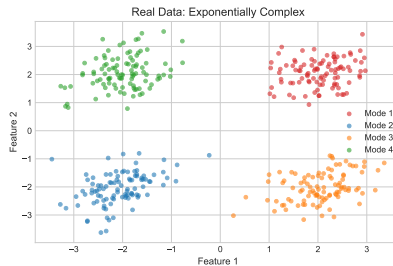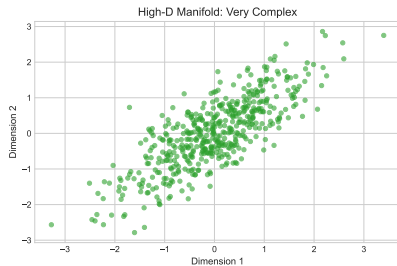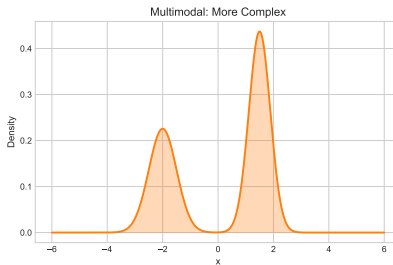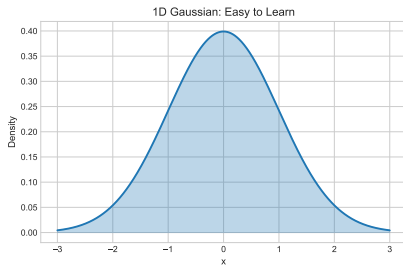- Samples new $x \sim P(x)$
- Creates novel instances

Examples: VAEs, GANs, Diffusion
Can sample new x? YES - generates from distribution

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

The Fundamental Tradeoff in Generative Modeling

- **Sweet Spot** — (green marker at top)
- **Modern Diffusion: Balanced** (green square)
- **VAE: Blurry but Consistent** (purple square)
- **Early GAN: Mode Collapse** (red square)
- **Mode Collapse** (red marker)
- **Random Noise: Diverse but Bad** / **Incoherent Generation** (blue markers)

Axis: Quality →
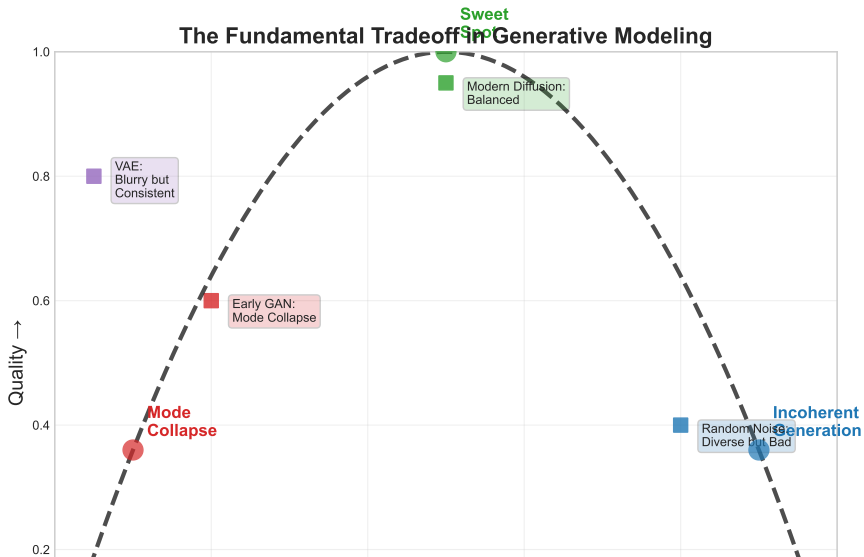
Inception Score (Higher = Better) / FID Score (Lower = Better) / Text Perplexity (Lower = Better)
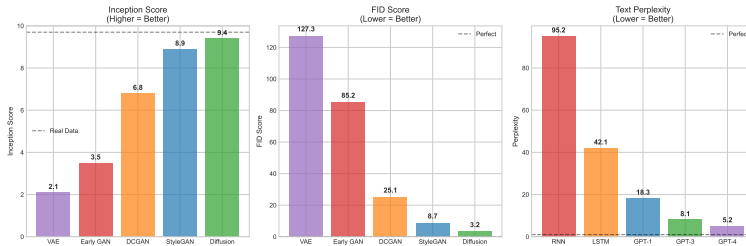
Inception Score (IS)
- Range: 1-1000
- Higher = better
- Quality & diversity

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
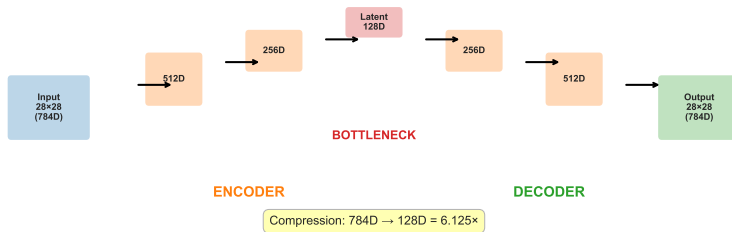- Lower = better
- Predictability

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

Quantitative metrics enable objective quality assessment and model comparison

Autoencoder Architecture: Compression Through Reconstruction

| Input 28×28 (784D) | 512D | 256D | Latent 128D | 256D | 512D | Output 28×28 (784D) |

BOTTLENECK

ENCODER   DECODER

Compression: 784D → 128D = 6.125×

Encoder

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

Reconstruction:

- Lossy process
- Preserves essentials
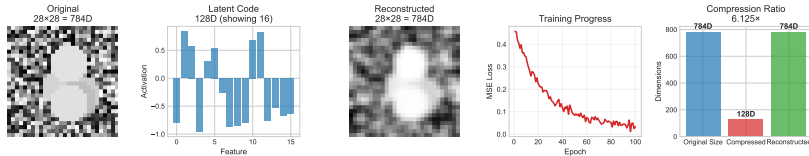
Architecture:

- Input: 784 pixels
- Encoder: 784 -¿ 128
- Decoder: 128 -¿ 784

Training:

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

MSE drops 0.45 -¿ 0.03 over 100 epochs

Autoencoder Successes

Visualization Placeholder

(Chart 12)

[+] SUCCESSES:
- Dimensionality reduction: 784D -¿ 128D

Quantitative Results:
- MSE: 0.031, Compression: 6.125x

Autoencoder Failures

Visualization Placeholder

(Chart 13)

[-] FAILURES:

- Blurry outputs (averaging)

Generation Metrics:

Metric	Score
IS	2.1

Averaging Problem

Visualization Placeholder

(Chart 14)

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

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

Vae Framework

Visualization Placeholder

(Chart 15)

Key Innovation:

- Encode to distribution, not point
- $q_\phi(z|x) = \mathcal{N}(\mu(x), \sigma^2(x))$

VAE Loss (ELBO):

$$\mathcal{L} = \underbrace{-E[\log p_\theta(x|z)]}_{\text{Reconstruction}} + \underbrace{KL(q_\phi(z|x)||p(z))}_{\text{Regularization}}$$

Artist Learning Process

Visualization Placeholder

(Chart 16)

Traditional Art Education:

- Student creates artwork

Key Insights:

- Adversarial feedback drives improvement

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?

Reverse Corruption Analogy

Visualization Placeholder

(Chart 19)

Forward (Corruption):

- Clean image -¿ pure noise

Reverse (Generation):

- Pure noise -¿ clean image

Gan Geometric Dynamics

Visualization Placeholder

(Chart 20)

Generator:

- Maps noise z to data x

Discriminator:

- Separates real from fake

Gan Training Walkthrough

Visualization Placeholder

(Chart 21)

Epoch 1:

- D_loss: 1.386

Epoch 100:

- D_loss: 0.695

Diffusion Mathematics

Visualization Placeholder

(Chart 22)

Forward (Fixed):

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

Noise Schedule β_t:

- Linear: 0.0001 -¿ 0.02

Reverse (Learned):

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

Network ϵ_θ predicts added noise
Training objective:

Latent Interpolation

Visualization Placeholder

(Chart 23)

GAN Interpolation:

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

Applications:

- Style transfer, face morphing

Denoising Steps

Visualization Placeholder

(Chart 24)

Key Time Steps:

- T=1000: Pure noise

Process Control:

- Guidance scale

Adversarial Theory

Visualization Placeholder

(Chart 25)

Theory:

- Minimax convergence

Benefits:

- Sharp, realistic images

Quality Metrics Over Time

Visualization Placeholder

(Chart 26)

Generative Model Results (MNIST):

Method	IS	FID	Time
Baseline (random)	1.0	500	-

Key Observations:

- Diffusion: Best quality, longest training
- GAN: Near-diffusion quality, 4x faster

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

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

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 ¡ 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
- Symptom: KL term -¿ 0
- Decoder ignores latent
- Fix: β-VAE, KL warm-up

2. Blurry Outputs
- Inherent to MSE loss
- Reconstruction averages
- Fix: Perceptual loss, adversarial

GAN Pitfalls

1. Mode Collapse
- Generator ignores diversity
- Produces limited variety
- Fix: Minibatch discrimination, unrolled GAN

2. Training Instability
- Loss oscillates wildly
- No convergence
- Fix: Wasserstein loss, spectral norm, TTUR

Diffusion Pitfalls

1. Slow Sampling
- 1000 steps = slow
- Latency bottleneck
- Fix: DDIM, distillation, 50 steps

2. Memory Intensive
- High-res = huge memory
- Training cost
- Fix: Latent diffusion, gradient checkpointing

Each approach has characteristic failure modes requiring specific mitigation strategies

Generative AI Best Practices
From Research to Production

Training Best Practices:

1. Start Simple
- Low resolution first (64x64 before 1024x1024)
- Small models before large
- Validate on toy datasets

2. Monitor Obsessively
- Log every 100 steps
- Visual inspection of samples
- Track FID/IS throughout
- Watch for collapse/divergence

3. Use Pretrained Models
- Transfer learning saves weeks
- Fine-tune from Stable Diffusion
- Leverage foundation models

4. Ablation Studies
- Test each component independently
- Measure contribution to performance
- Simplify when possible

Deployment Best Practices:

1. Quality Control
- Human-in-the-loop review
- NSFW content filtering
- Bias detection systems
- Watermarking for traceability

2. Performance Optimization
- Model quantization (FP16, INT8)
- Distillation for speed
- Caching common generations
- Batch processing

3. Safety Measures
- Rate limiting to prevent abuse
- Content moderation
- Prompt injection defenses
- User consent and attribution

4. Continuous Improvement
- Collect user feedback
- Retrain on curated data

Fundamental Trade-offs
No Free Lunch in Generative Modeling

Generative Tradeoffs

Visualization Placeholder

(Chart 29)

Training Stability:
- VAEs, Diffusion: Stable
- GANs: Unstable

Quality:
- Diffusion, GANs: Excellent
- VAEs: Blurry

Modern Applications

Visualization Placeholder

(Chart 30)

Image Generation:
- DALL-E 3, Midjourney
- Stable Diffusion, Firefly

Text Generation:
- GPT-4, Claude, Gemini
- Llama 2 (open)

Ethics Summary

Visualization Placeholder

(Chart 31)

Capabilities:

- Realistic images from text
- Human-like writing

Challenges:

- Deepfakes, misinformation
- Copyright issues