

Week 0e: Generative AI

The Creation Challenge

Machine Learning for Smarter Innovation

BSc-Level Course

October 7, 2025

Contents

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

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

Generative models learn full data distributions enabling sampling - classification learns boundaries, generation learns manifolds

Mathematical Foundation

Two Approaches to Learning

Discriminative Models

Learn: $P(y|x)$ - Conditional probability
(Ng & Jordan 2002: Defined distinction)

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
- Sample via ancestral sampling or MCMC
- Samples new $x \sim P(x)$
- Creates novel instances

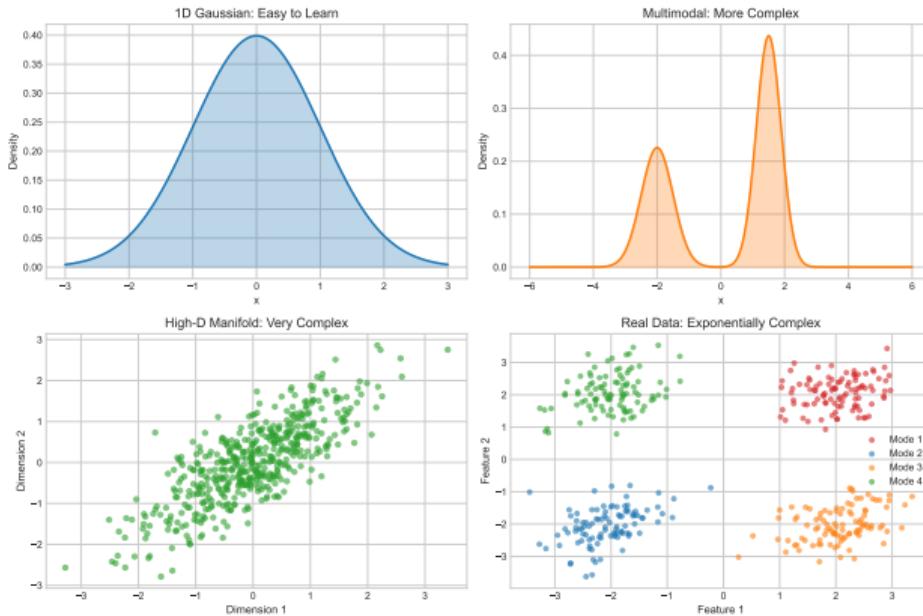
Examples: VAEs, GANs, Diffusion

Can sample new x ? YES - generates from distribution

Discriminative models $P(y|x)$ learn boundaries while generative models $P(x)$ or $P(x, y)$ learn distributions - fundamental distinction enables creation

The Hard Problem

Why Generation is Fundamentally Difficult



Challenges:

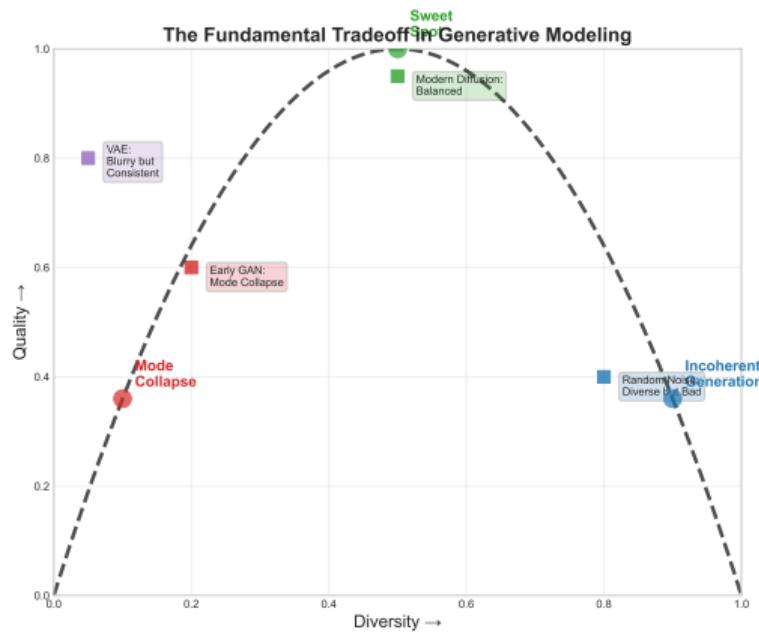
- High-dimensional spaces
- Multimodal distributions
- Curse of dimensionality (data lies on low-dimensional manifolds)
- Sample complexity grows exponentially

Requirements:

- Capture all patterns
- Maintain realism
- Computational tractability (exact inference intractable)

The Fundamental Tradeoff

Quality vs Diversity Dilemma



High Quality: Mode collapse, repetitive

Balanced: Realistic variety

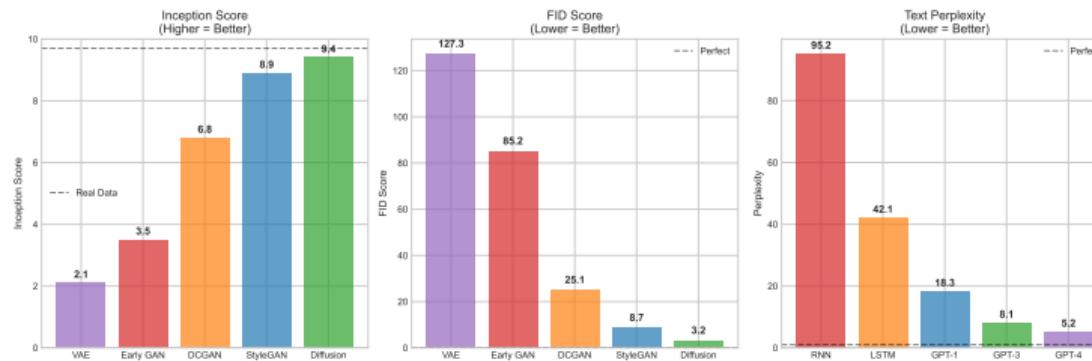
High Diversity: Unrealistic

Explanations:

- Mode Collapse: Generator produces limited variety (high quality, low diversity)
- Coverage Issue: Generator spreads thin (high diversity, low quality)

Measuring Generation Quality

Metrics for Evaluating Generative Models



Inception Score (IS) (Salimans et al. 2016)

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

Interpretation:

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

Note: Human evaluation remains gold standard

Quantitative metrics approximate perceptual quality - IS measures label confidence and diversity, FID measures feature distribution distance, perplexity measures predictability

FID Score (Heusel et al. 2017: Fréchet Inception Distance)

- 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

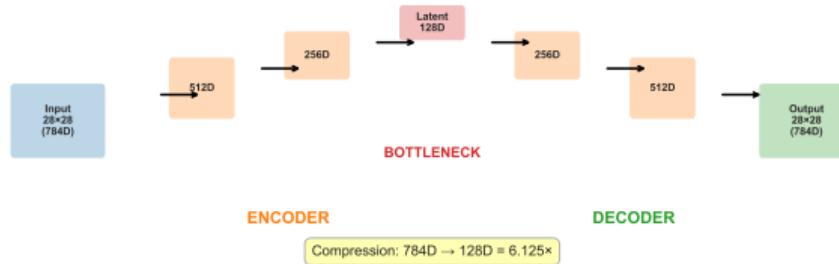
Interpretation:

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

Autoencoders: The Foundation

Learning Compressed Representations

Autoencoder Architecture: Compression Through Reconstruction



Encoder

- 784D \rightarrow 128D
- Learns $q(z|x)$ mapping
- Forces selective encoding
- Filters noise

Latent

- 128D bottleneck
- Key features only
- 6.1x compressed

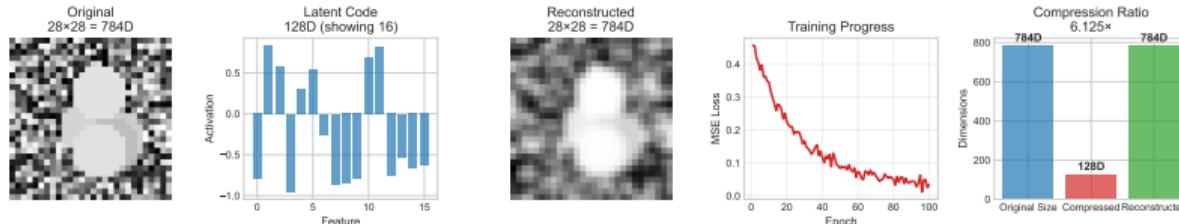
Decoder

- 128D \rightarrow 784D
- Learns $p(x|z)$ mapping
- Lossy reconstruction
- Preserves essentials

Bottleneck architecture forces dimensionality reduction - information bottleneck principle requires encoding only essential features for reconstruction

Worked Example: MNIST Compression

From 784 Pixels to 128 Features



Architecture:

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

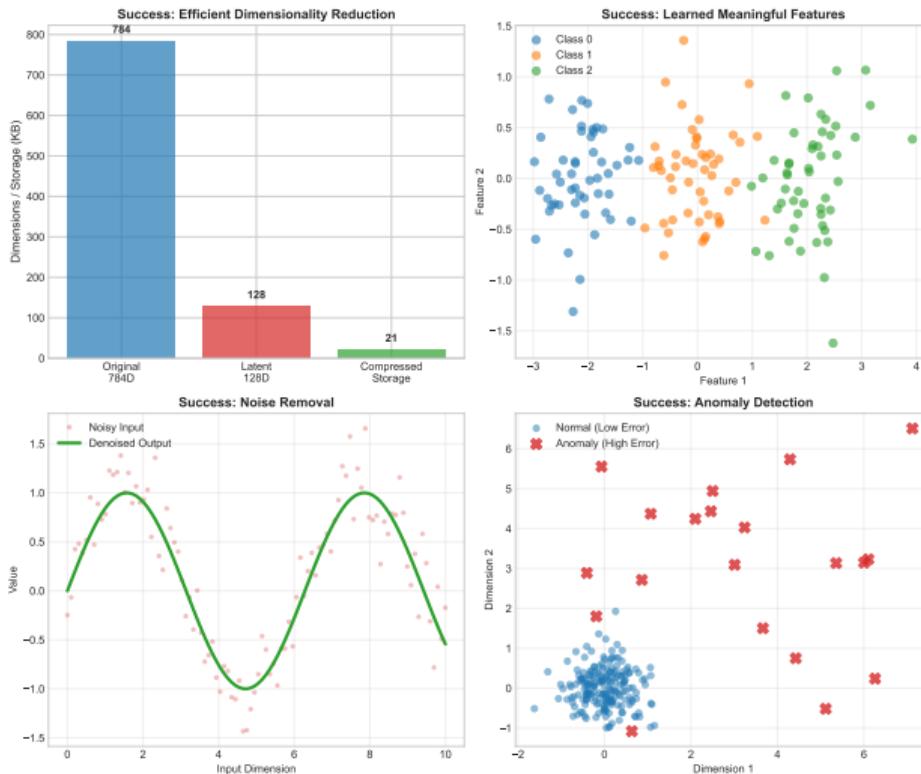
Training:

- Loss: $L = ||x - \hat{x}||^2$
- Optimizer: Adam
- Epoch 1: MSE=0.45
- Epoch 100: MSE=0.03
- Compression: 6.125x

Reconstruction loss decreases monotonically with training - 6x compression ratio demonstrates learned features capture digit essence while discarding pixel-level noise

Autoencoder Successes

What Works Well

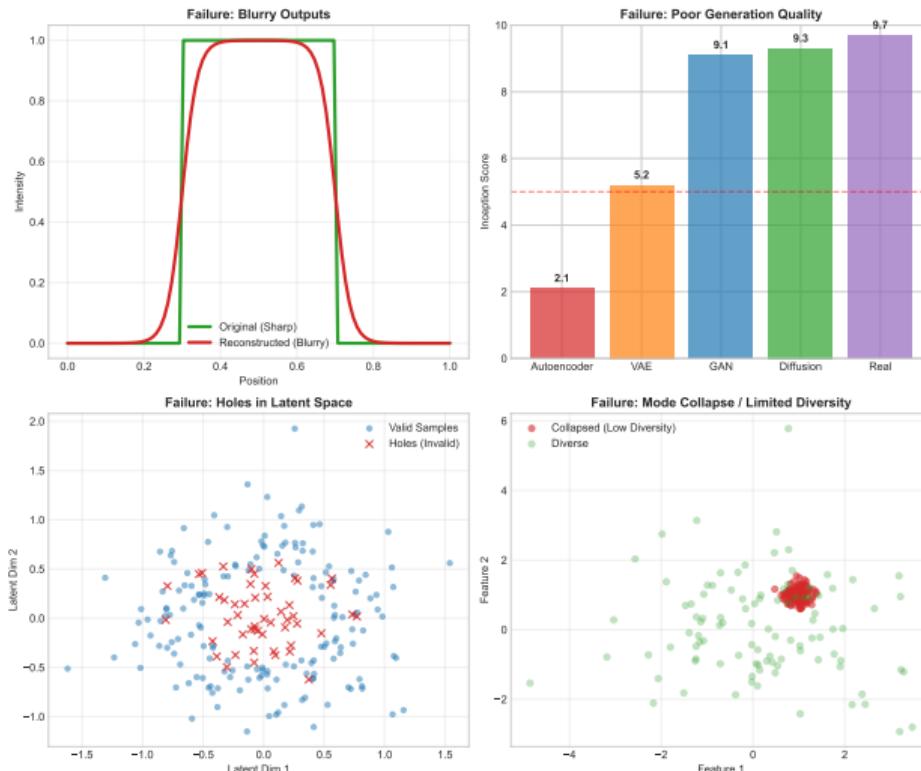


[+] SUCCESSES:

Results:

Autoencoder Limitations

The Generation Problem

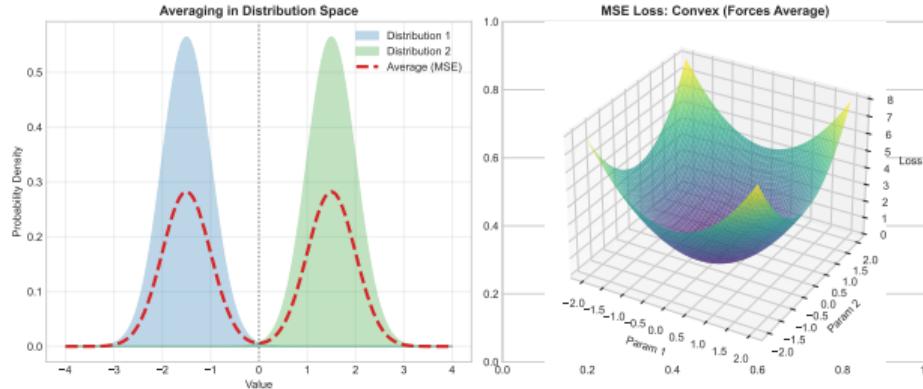
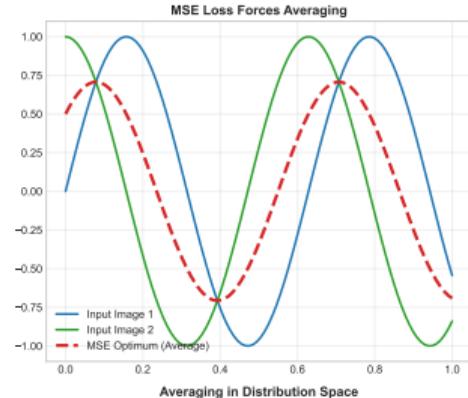


[] FAILURES:

Machine Learning
IS 2.1

Root Cause Analysis

Why Autoencoders Generate Poorly

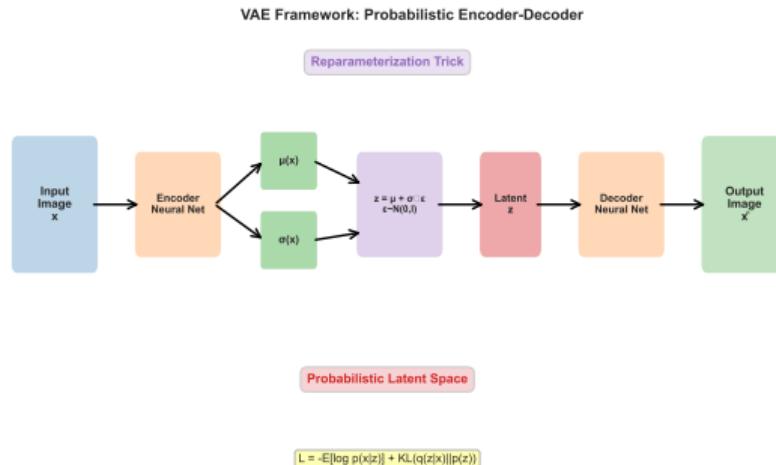


Problem:

Math:

Variational Autoencoders (VAEs)

The Probabilistic Solution (Kingma & Welling 2013, Rezende et al. 2014)



Key Innovation:

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

Reparameterization:

- Make z deterministic
- Gradient flows

VAE Loss (ELBO):

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

Two terms:

- Reconstruction quality
- KL forces $q(z|x)$ close to prior $p(z) = \mathcal{N}(0, I)$
- β -VAE balances

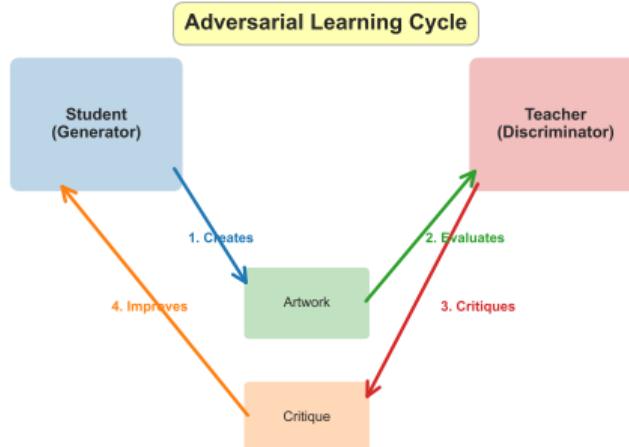
ELBO = Evidence Lower Bound: Tractable objective

Reparameterization trick $z = \mu + \sigma \odot \epsilon$ separates stochasticity enabling backpropagation through sampling - VAEs optimize variational lower bound on log-likelihood

Human Learning Analogy

How Artists Develop Mastery

How Artists Improve Through Critique → GANs



Both Student and Teacher Improve Through Competition

Art Education:

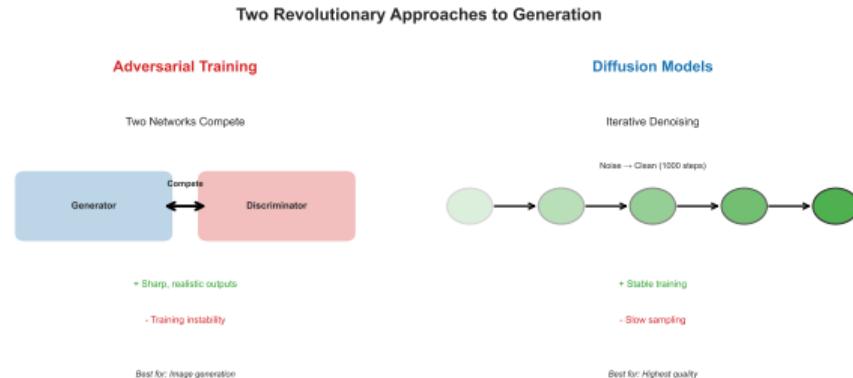
- Student creates
- Teacher critiques
- Student improves

Insights:

- Adversarial feedback drives improvement
- Both improve together

Two Revolutionary Approaches

Beyond VAEs to Better Generation



Adversarial

- Two networks compete
- Sharp, realistic
- No explicit likelihood

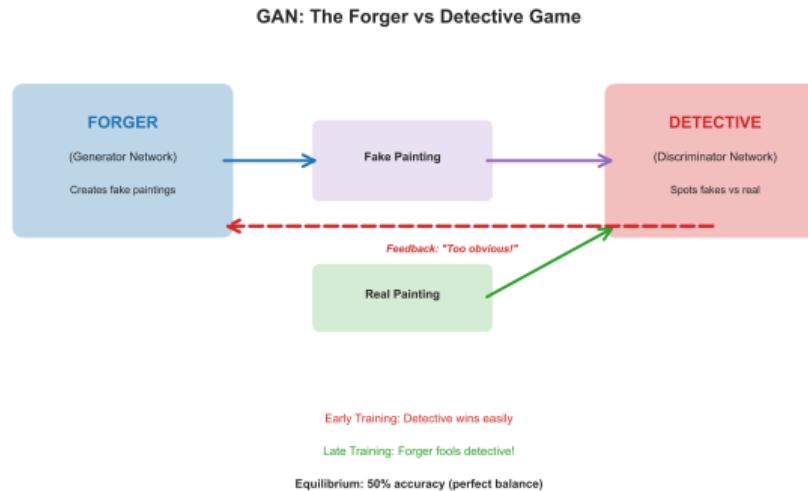
Diffusion

- Iterative denoising
- Stable, controllable
- Likelihood-based, traceable gradients

Adversarial and diffusion approaches overcome VAE's MSE averaging problem through different mechanisms - competition vs iterative refinement both avoid explicit averaging

GANs: The Forger vs Detective Game

Adversarial Training in Plain English



Forger:

- Creates fakes
- Fools detective

Result: Detective can't tell fake from real!

Equilibrium: When forger succeeds 50% of time

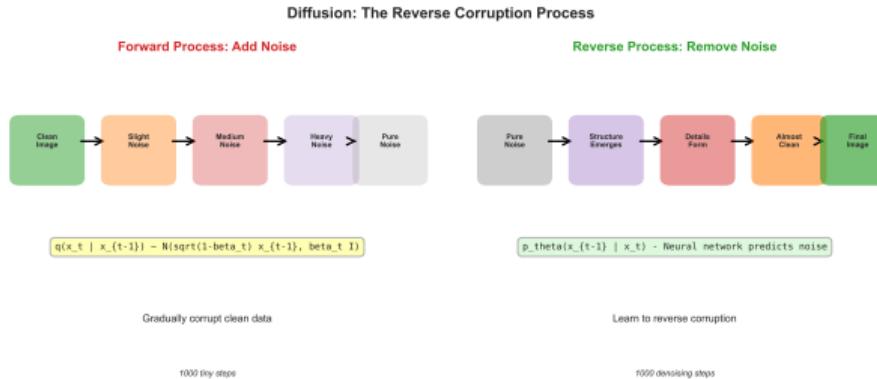
Competition drives both to excellence

Detective:

- Examines: real/fake?
- Gets better at detection

Diffusion: The Reverse Corruption Process

Denoising in Plain English



Forward:

- Clean - \downarrow noise
- 1000 steps

Reverse:

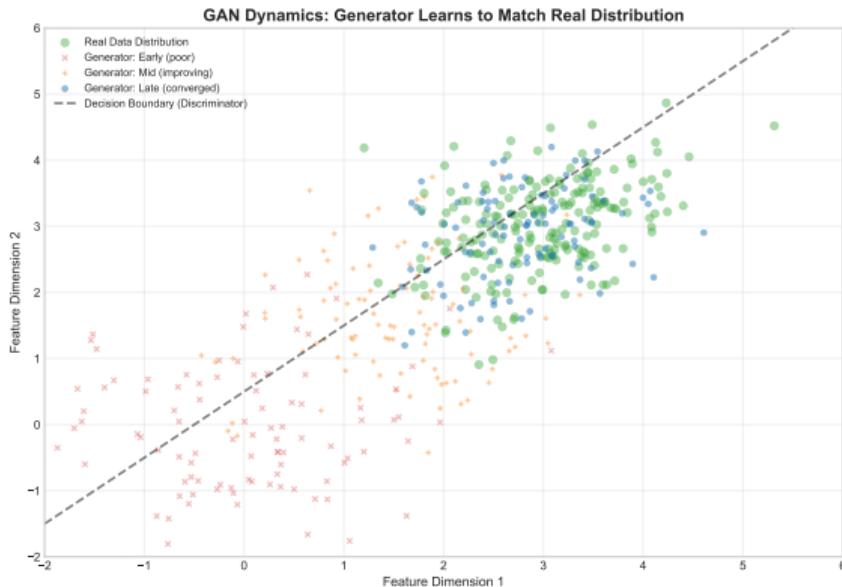
- Noise - \downarrow clean
- 1000 steps
- Training: Learn to predict noise at each step
- Inference: Start from pure noise, denoise 1000 times

Key: Learn to undo corruption

Diffusion inverts gradual noise corruption process - learning reverse process enables sampling by denoising pure noise, avoiding VAE averaging through iterative refinement

GAN Dynamics: Geometric View

Understanding the Adversarial Process (Goodfellow et al. 2014)



Generator:

- Maps z to x
- Loss: $-\log D(G(z))$

Minimax objective:

$$\min_G \max_D V(D, G)$$

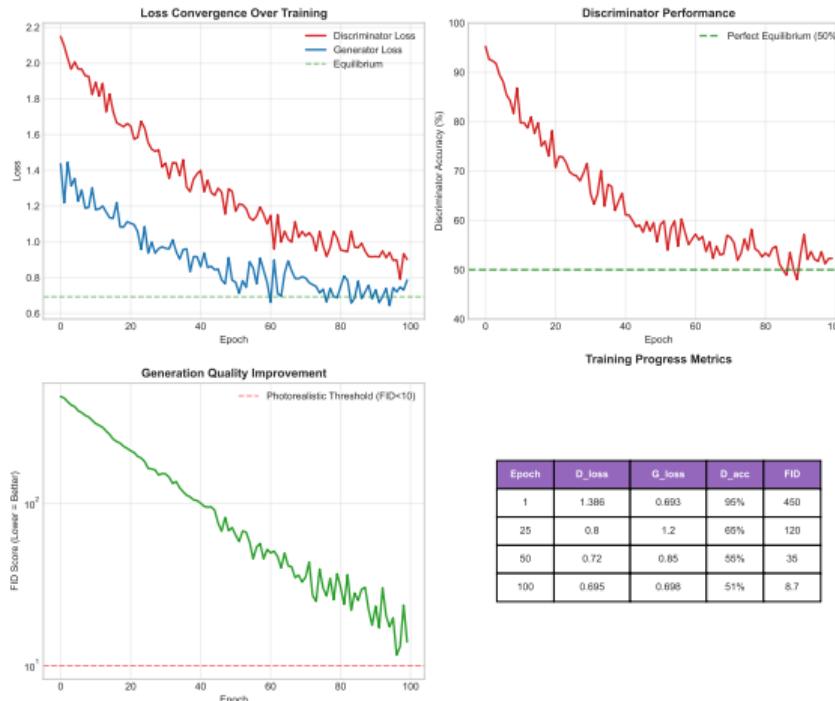
Nash equilibrium occurs when $p_g = p_{data}$ and discriminator accuracy equals 50% - adversarial objective mathematically guarantees convergence under ideal

Discriminator:

- Separates real/fake
- Loss: $-\log D(x) - \log(1 - D(G))$

GAN Training: Step-by-Step Example

Real Loss Values from MNIST Training



Epoch 1:

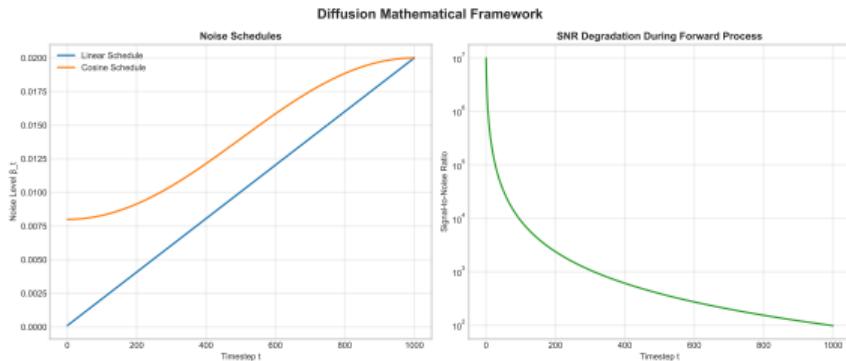
- D: 1.386, G: 0.693
- Images: noise

Epoch 100:

- D: 0.695, G: 0.698
- Images: realistic

Diffusion Mathematical Framework

Forward and Reverse Processes (Ho et al. 2020 - DDPM)



Forward:

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

Noise Schedule:

- β_t controls noise schedule
- Linear: 0.0001 - \downarrow 0.02
- Cosine: Variable rate
- Matters: Smooth degradation

Noise prediction objective enables stable training - predicting $\epsilon_\theta(x_t, t)$ instead of x_0 reduces variance, noise schedule controls diffusion speed (linear $\beta_t : 10^{-4} \rightarrow 2 \times 10^{-2}$ standard)

Reverse:

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

Training:

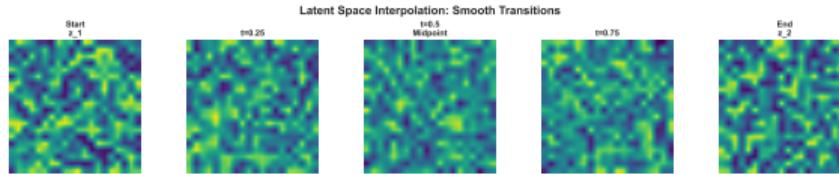
$$L = E[||\epsilon - \epsilon_\theta(x_t, t)||^2]$$

Denoising objective: Predict noise ϵ , not image x_0

Intuition: Predict noise, subtract it

Latent Space Interpolation

Smooth Transitions in Generated Content



Method:

- Sample z_1, z_2
- Interpolate: $z_t = (1 - t)z_1 + tz_2$
- Generate: $x_t = G(z_t)$
- Spherical interpolation: Better than linear for normalized spaces

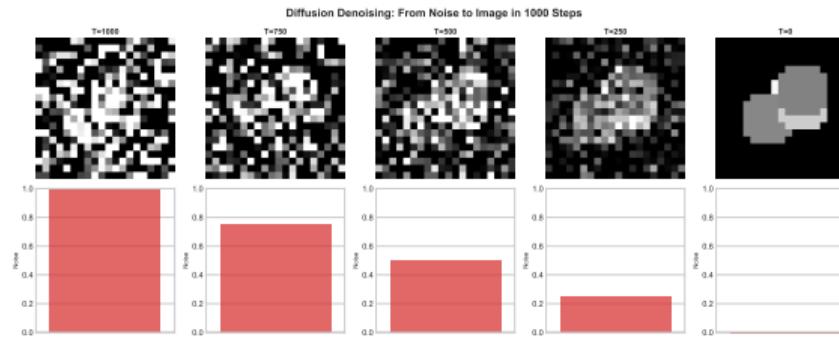
Applications:

- Style transfer
- Face morphing
- Molecule generation (drug discovery latent optimization)
- Drug discovery

Continuous latent spaces enable semantic interpolation - walking along manifold generates smooth transitions revealing learned structure organization and enabling controlled generation

Diffusion Denoising Visualization

From Noise to Image in 1000 Steps



Steps:

- T=1000: Noise
- T=500: Structure
- T=0: High quality
- Controllable: Stop early for variations
- DDIM (Song et al. 2020): 50 steps, 20x speedup

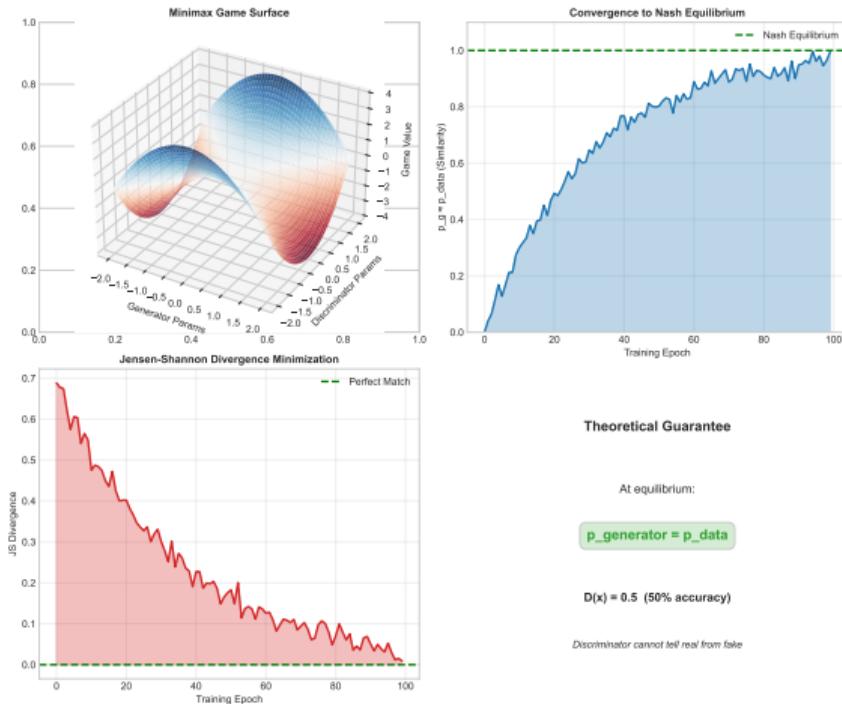
Control:

- Guidance scale
- Step count

Progressive denoising reveals hierarchical generation - coarse structure emerges early ($T=1000 \rightarrow 500$), fine details refine late ($T=500 \rightarrow 0$), enabling quality-speed trade-offs

Why Adversarial Training Works

The Mathematical Guarantee



Theory:

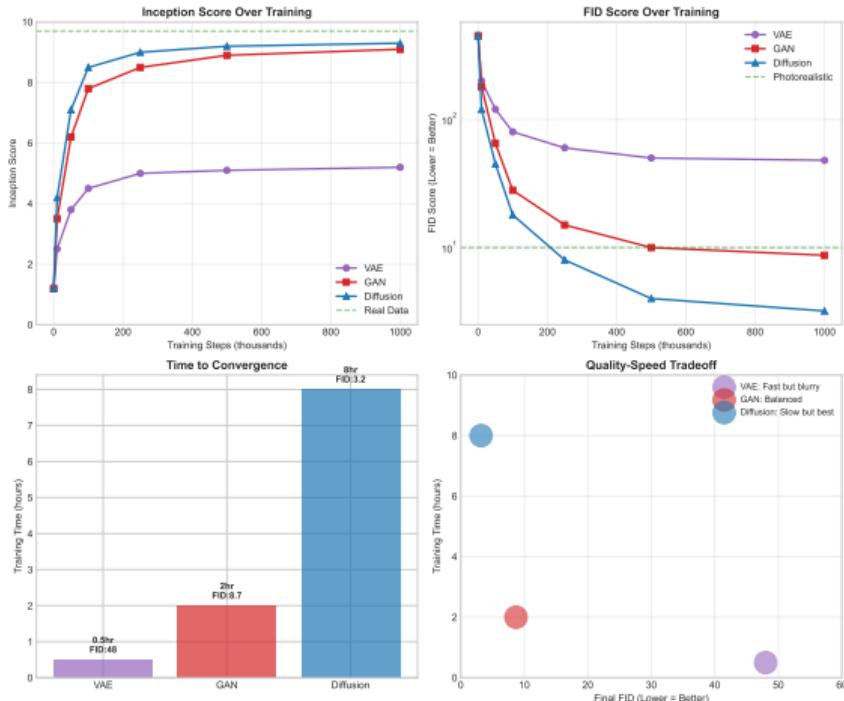
- Minimax convergence
- Equilibrium: $p_g = p_{\text{data}}$

Benefits:

- Sharp, realistic
- Fine details

Experimental Validation

Quality Metrics vs Training Progress



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

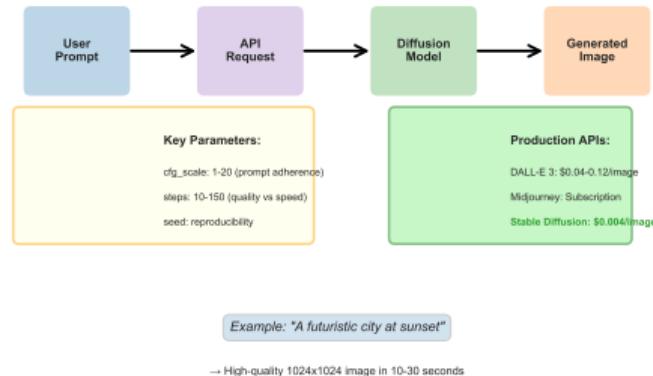
Observations:

- Diffusion: Best
- GAN: 4x faster
- VAE: Fast, blurry

Implementation: Stable Diffusion API

Production-Ready Generative AI

Stable Diffusion API: Production-Ready Generation



Usage:

```
response = requests.post(
    api_url,
    headers={"Auth": key},
    json={
        "text_prompts": [{"text": "city"}],
        "cfg_scale": 7,
        "steps": 30
    })
```

Production APIs abstract complexity - configuration parameters control guidance strength and quality-speed tradeoffs, enabling accessible deployment without deep expertise.

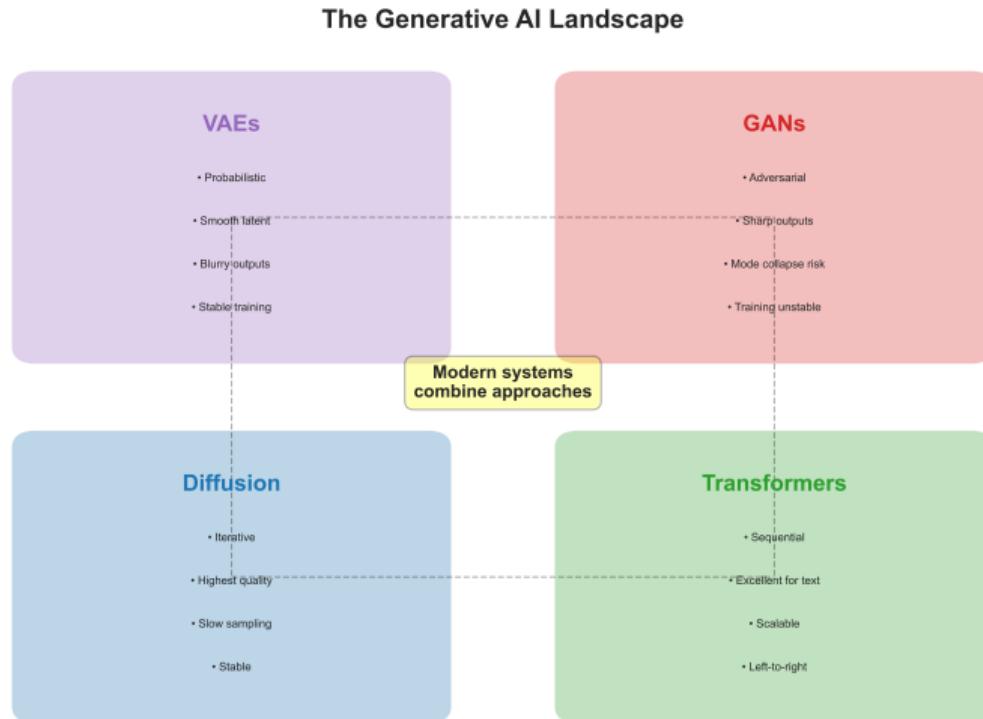
Parameters:

- cfg scale: 1-20 (balance prompt adherence vs diversity)
- steps: 10-150 (quality-speed trade-off: 10=fast/rough, 150=slow/perfect)

Cost: \$0.004/image

The Generative AI Landscape

Four Fundamental Approaches



VAEs (2013): Probabilistic, smooth latent, blurry - First scalable

GANs (2014): Adversarial, sharp outputs, unstable - Realism breakthrough

Diffusion (2020): Iterative denoising, high quality, slow - SOTA quality

Transformers (2017): Sequential, excellent text, scalable - Attention

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)

Model selection requires systematic decision framework - prioritize constraints (data size, latency, quality requirements) then match to architectural strengths balancing engineering and scientific considerations

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
- All: Need deterministic outputs (use retrieval instead)

Common Pitfalls: What Can Go Wrong

Failure Modes and Solutions

VAE Pitfalls

1. Posterior Collapse

- KL $-l \approx 0$
- Fix: β -VAE, warm-up

2. Blurry

- MSE averages
- Fix: Perceptual loss

3. KL Annealing

- Warm-up schedule prevents collapse

GAN Pitfalls

1. Mode Collapse

- Limited variety
- Fix: Minibatch disc

2. Unstable

- Oscillates
- Fix: Wasserstein, spectral norm

3. Label Smoothing

- Prevents D overconfidence

Diffusion Pitfalls

1. Slow (1000 steps)

- Latency issue
- Fix: DDIM (50 steps)

2. Memory

- High-res costly
- Fix: Latent diffusion

3. Classifier-Free Guidance

- Better control

Understanding failure modes enables proactive mitigation - posterior collapse, mode collapse, and inference speed have well-established solutions requiring architecture-specific debugging strategies

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

5. Reproducibility

- Fix seeds, log hyperparameters, version data

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

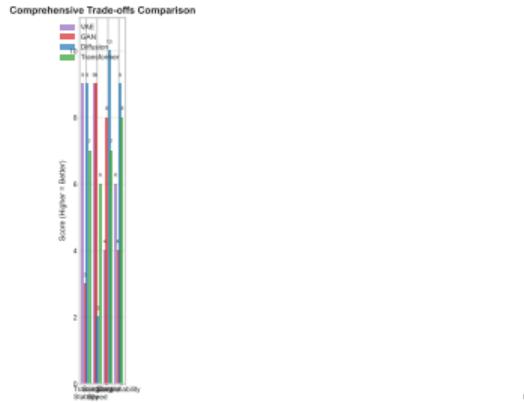
5. Versioning

- Model registry, A/B testing, rollback capability

Production deployment requires systematic engineering - start simple for validation, monitor obsessively for failure detection, use transfer learning for efficiency, implement safety guardrails for responsible deployment

Comprehensive Trade-offs

No Free Lunch in Generative Modeling



Stability:

- VAEs, Diffusion: Stable
- GANs: Unstable

Speed:

- VAEs, GANs: Fast
- Diffusion: Slow

Data Efficiency:

- VAE > Diffusion > GAN (sample requirements)

Quality:

- Diffusion, GANs: Excellent
- VAEs: Blurry

Control:

- Diffusion, Transformers: High
- GANs: Limited

Interpretability:

- VAE > Diffusion > GAN (latent structure)

Training Stability:

- Diffusion > VAE > GAN (convergence reliability)

No free lunch theorem applies - stability vs quality vs speed form fundamental trade-off triangle, optimal choice depends on problem constraints and deployment

State-of-the-Art Applications

Production Generative AI Systems

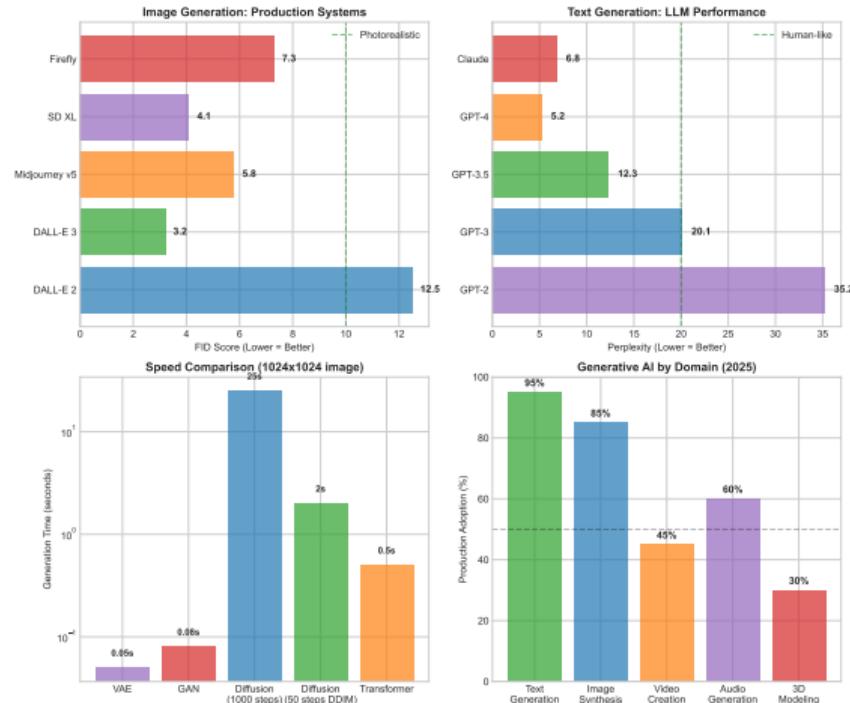


Image:

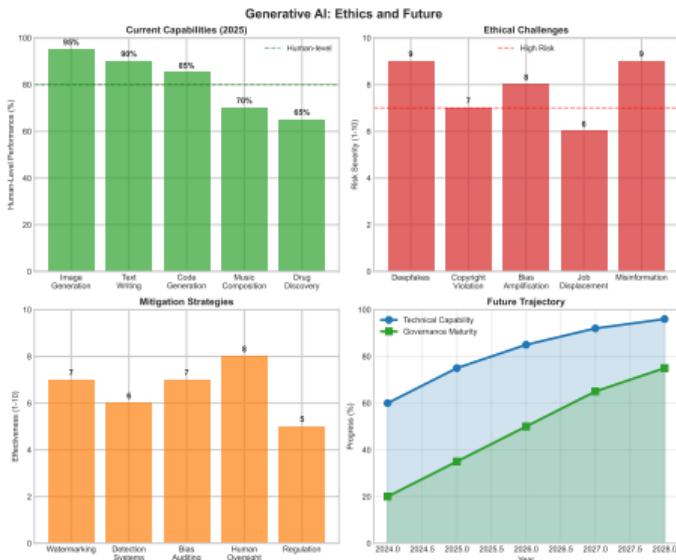
- DALL-E 3, Midjourney
- Stable Diffusion, Firefly

Text:

- GPT-4, Claude, Gemini
- Llama 2 (open)

Summary & Future of Generative AI

What We Learned and What's Next



Learned:

- VAEs: Probabilistic, blurry
- GANs: Adversarial, realistic
- Diffusion: Best quality
- Decision framework, pitfalls

Future:

Ethics:

- Deepfakes, copyright
- Bias, displacement
- Attribution: Training data transparency

Solutions:

- Watermarking, auditing