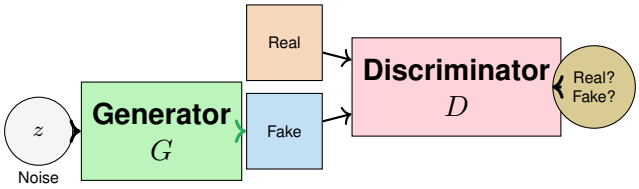
Deep Learning for Perception

Lecture 09: Generative Adversarial Networks (GANs)



"The coolest idea in machine learning in the last twenty years" — Yann LeCun (2016)

Contents

1 Why GANs? From VAEs to Adversarial Learning

Connection to Prior Learning

Your Learning Journey:
Autoencoders: Learned to compress and reconstruct—but *cannot* generate new data.
VAEs: Added probabilistic latent space—*can* generate by sampling $z \sim \mathcal{N}(0, I)$.
VAE's Limitation: Generated images are often **blurry**!

- VAE uses pixel-wise MSE loss: $||x - \hat{x}||^2$
- MSE averages over possibilities $\rightarrow$ blurry outputs
- Doesn't capture "what looks realistic to humans"

GAN's Solution: Replace pixel-wise loss with a **learned** loss—another neural network judges realism!

Why It Matters

The Paradigm Shift:

lightgray **VAE asks:**	**GAN asks:**
"Does output match input pixel-by-pixel?"	"Does output *look real* to a neural network?"
$\downarrow$	$\downarrow$
Blurry but stable	Sharp but tricky to train

Result: GANs produce **photorealistic** images—faces, art, scenes that never existed!

2 The Core Idea: Two Networks in Competition

Definition

A **GAN** consists of two neural networks in an adversarial game:
Generator (G) — The "Forger"

- **Input:** Random noise vector $z \sim \mathcal{N}(0, I)$ (e.g., 100 numbers)
- **Output:** Fake image $G(z)$
- **Goal:** Create fakes so good that D thinks they're real

Discriminator (D) — The "Detective"

- **Input:** Any image (real or fake)
- **Output:** Probability image is real: $D(x) \in [0, 1]$
- **Goal:** Correctly classify real vs fake

The Game: G improves to fool D; D improves to catch G. Both get better!

Analogy

Counterfeiter vs Detective
Round 1: Forger creates crude fake → Detective easily spots it.
Round 10: Forger improves technique → Detective learns new tells.
Round 1000: Forger creates masterpiece → Detective can barely tell!
Equilibrium: Fakes are indistinguishable from real. Detective outputs 50% (pure guess).

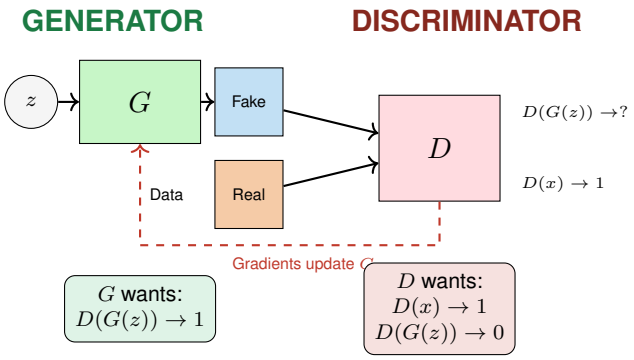


Figure 1: GAN training loop: Generator creates, Discriminator judges, both improve

3 The Minimax Objective

Key Formula

GAN Loss Function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]$$

Discriminator maximizes V:

- $D(x) \to 1$ for real images $\Rightarrow \log D(x) \to 0$ (high)
- $D(G(z)) \to 0$ for fakes $\Rightarrow \log(1 - D(G(z))) \to 0$ (high)

Generator minimizes V:

- Wants $D(G(z)) \to 1$ (fool D)
- Makes $\log(1 - D(G(z))) \to -\infty$ (low)

At equilibrium: $p_G = p_{data}$ (generated distribution matches real data)

Memory Hook

Remember the Goals:

lightgray **Input**	**D wants**	**G wants**
Real image x	$D(x) = 1$	(doesn't care)
Fake image $G(z)$	$D(G(z)) = 0$	$D(G(z)) = 1$

Perfect G: D outputs 0.5 for everything—can't distinguish!

4 Training Procedure

Definition

Alternating Optimization:
For each iteration:
Step 1 — Train D (make it better at detecting)

1. Sample real images $\{x_1, \ldots, x_m\}$
2. Sample noise $\{z_1, \ldots, z_m\}$, generate fakes $G(z_i)$
3. Update D by ascending: $\nabla_D \frac{1}{m} \sum [\log D(x_i) + \log(1 - D(G(z_i)))]$

Step 2 — Train G (make it better at fooling)

1. Sample new noise $\{z_1, \ldots, z_m\}$
2. Update G by ascending: $\nabla_G \frac{1}{m} \sum [\log D(G(z_i))]$

Repeat until images look realistic.

Common Pitfall

GANs Are Hard to Train!
Mode Collapse: G produces only one type of output (e.g., same face repeatedly).
Oscillation: G and D chase each other, never converging.
Vanishing Gradients: If D is too good, $D(G(z)) \approx 0$ always $\rightarrow$ no gradient for G.
Solutions: WGAN (better loss), spectral normalization, progressive growing, careful hyperparameters.

5 Popular GAN Variants

Key GAN Architectures

DCGAN (Deep Convolutional GAN)

- Uses CNNs with strided convolutions (no pooling)
- BatchNorm + LeakyReLU in D, ReLU in G
- First stable architecture for high-quality images

Conditional GAN (cGAN)

- Adds label y to both G and D: $G(z, y)$, $D(x, y)$
- Generates *specific* classes: "Generate a cat"

Wasserstein GAN (WGAN)

- Uses Earth Mover distance instead of JS divergence

- Much more stable training
- D becomes "Critic" (no sigmoid, outputs real number)

StyleGAN

- Controls "style" at multiple scales
- Generates photorealistic faces
- Powers "This Person Does Not Exist"

Pix2Pix / CycleGAN

- Image-to-image translation
- Pix2Pix: Paired data (edges $\rightarrow$ photo)
- CycleGAN: Unpaired data (horse $\leftrightarrow$ zebra)

6 GAN vs VAE: Choosing the Right Model

Head-to-Head Comparison

primaryblue!15 **Aspect**	**VAE**	**GAN**
Output Quality	Blurry	Sharp, realistic
Training	Stable	Unstable, tricky
Latent Space	Smooth, structured	Unstructured
Mode Coverage	Covers all modes	May collapse
Has Encoder?	Yes (can encode $x \rightarrow z$)	No (generation only)
Loss Meaning	Meaningful (ELBO)	Not meaningful
Best For	Representation learning	Image generation

Memory Hook

Quick Decision:
Need encoder? $\rightarrow$ VAE
Need sharp images? $\rightarrow$ GAN
Need both? $\rightarrow$ VAE-GAN hybrid

7 Applications

- **Face Generation:** StyleGAN creates photorealistic faces that don't exist
- **Image-to-Image:** Sketch $\rightarrow$ photo, day $\rightarrow$ night, low-res $\rightarrow$ high-res
- **Data Augmentation:** Generate training data for rare classes
- **Art & Design:** Generate art, logos, fashion designs

- **Video:** Generate realistic video frames
- **Medical:** Synthesize medical images for training
- **Deepfakes:** Face swapping (raises ethical concerns!)

8 Summary

Key Takeaways

1. What is a GAN?

- Two networks: Generator (creates fakes) vs Discriminator (detects fakes)
- Adversarial game: Both improve through competition

2. The Objective

$$\min_G \max_D \; \mathbb{E}[\log D(x)] + \mathbb{E}[\log(1 - D(G(z)))]$$

3. Training Challenges

- Mode collapse, oscillation, vanishing gradients
- Solutions: WGAN, careful architecture, hyperparameter tuning

4. GAN vs VAE

- GAN: Sharp images, no encoder, unstable
- VAE: Blurry images, has encoder, stable

5. Key Variants: DCGAN, cGAN, WGAN, StyleGAN, CycleGAN

Self-Test

Q1: What are the two networks in a GAN and what does each do?
A: Generator creates fake images from noise; Discriminator classifies real vs fake.
Q2: Why are GANs sharper than VAEs?
A: GANs use adversarial loss ("does it look real?") vs VAE's pixel-wise MSE (which averages $\rightarrow$ blur).
Q3: What is mode collapse?
A: Generator produces only one type of output, ignoring diversity in the data.
Q4: At equilibrium, what does D output?
A: 0.5 for everything—it can't distinguish real from fake.
Q5: When would you use VAE over GAN?
A: When you need an encoder, smooth latent space, or stable training.

9 Glossary

primaryblue!15 **Term**	**Definition**
Generator	Network that creates fake data from random noise
Discriminator	Network that classifies inputs as real or fake
Adversarial	Competing/opposing objectives between two networks
Mode Collapse	Generator produces limited variety of outputs
Minimax	Game where one player minimizes, other maximizes
DCGAN	GAN using deep convolutional networks
cGAN	Conditional GAN that takes class labels as input
WGAN	GAN using Wasserstein distance for stable training