Loyola University Chicago

Generative Adversarial Networks (GANs)

Exploring the Art and Science of AI-generated Content

02

Loyola University Chicago

Introduction

A GAN is a type of machine learning model where two neural networks compete with each other to generate realistic data.

A “Generator” tries to create fake data, and a “Discriminator” tries to tell if it's real or fake. They improve together through competition.

next slide →

How GANs Work?

Two models:
Generator (G): *Learns to create fake data*
Discriminator (D): *Learns to detect fake data*

The goal: *G tries to fool D, D tries to catch G*
Training is a back-and-forth game (minimax optimization)

next slide →

Loyola University Chicago

Training Loop

- **Step 1:** *G creates fake data from random noise*
- **Step 2:** *D sees both real and fake data*
- **Step 3:** *D tries to classify correctly*
- **Step 4:** *G updates to better fool D*
- *Repeat for many epochs*

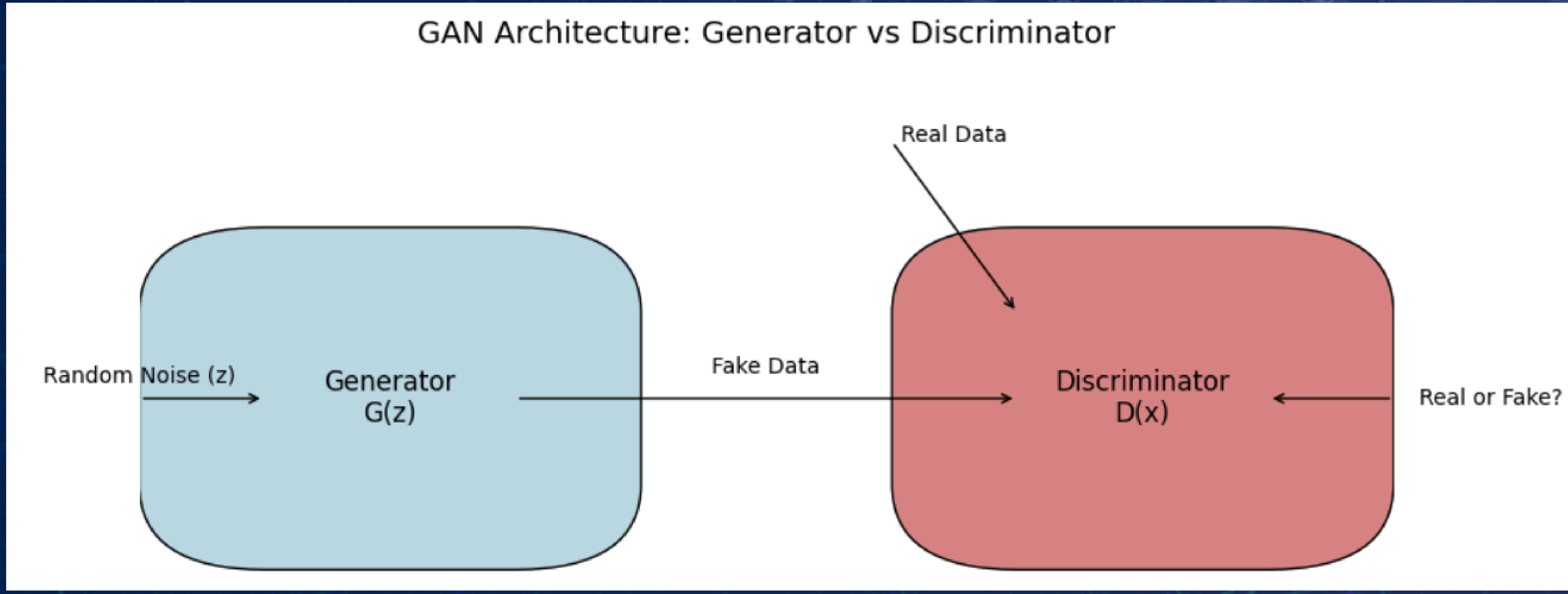
next slide

```python
import matplotlib.pyplot as plt
import matplotlib.patches as patches

# Create a diagram to show the interaction between Generator and Discriminator in a GAN
fig, ax = plt.subplots(figsize=(10, 6))

# Draw Generator box
generator_box = patches.FancyBboxPatch((0.1, 0.5), 0.2, 0.2, boxstyle="round,pad=0.1", edgecolor='black', facecolor='lightblue')
ax.add_patch(generator_box)
ax.text(0.2, 0.6, "Generator\nG(z)", ha="center", va="center", fontsize=12)

# Draw Discriminator box
discriminator_box = patches.FancyBboxPatch((0.7, 0.5), 0.2, 0.2, boxstyle="round,pad=0.1", edgecolor='black', facecolor='lightcoral')
ax.add_patch(discriminator_box)
ax.text(0.8, 0.6, "Discriminator\nD(x)", ha="center", va="center", fontsize=12)

# Add noise input arrow to Generator
ax.annotate("", xy=(0.1, 0.6), xytext=(0.0, 0.6), arrowprops=dict(arrowstyle="->"))
ax.text(0.0, 0.62, "Random Noise (z)", ha="center", fontsize=10)

# Arrow from Generator to Discriminator
ax.annotate("", xy=(0.7, 0.6), xytext=(0.3, 0.6), arrowprops=dict(arrowstyle="->"))
ax.text(0.5, 0.63, "Fake Data", ha="center", fontsize=10)
```

INPUT:

```python
# Arrow from Generator to Discriminator
ax.annotate("", xy=(0.7, 0.6), xytext=(0.3, 0.6), arrowprops=dict(arrowstyle="->"))
ax.text(0.5, 0.63, "Fake Data", ha="center", fontsize=10)

# Real data input to Discriminator
ax.annotate("", xy=(0.7, 0.7), xytext=(0.6, 0.9), arrowprops=dict(arrowstyle="->"))
ax.text(0.65, 0.9, "Real Data", ha="center", fontsize=10)

# Discriminator output
ax.annotate("", xy=(0.9, 0.6), xytext=(1.0, 0.6), arrowprops=dict(arrowstyle="->"))
ax.text(1.02, 0.6, "Real or Fake?", va="center", fontsize=10)

# Hide axes
ax.axis("off")
plt.title("GAN Architecture: Generator vs Discriminator", fontsize=14)
plt.tight_layout()
plt.show()

from pathlib import Path
import json

from pathlib import Path
import json

from pathlib import Path
import json

from pathlib import Path
import json
```

OUTPUT:



ŊG

Loyola University Chicago

Variants of GANs

VANILLA GAN

The original model

DCGAN

Uses convolutional layers, good for images

CYCLEGAN

Translates images (e.g., horse ↔ zebra)

STYLEGAN

Very high-quality face generation

CONDITIONAL GAN (CGAN)

Adds labels as input (e.g., generate cats or dogs)

next slide →

Loyola University Chicago

Applications of GANs

1. **Image generation:** *Portraits, fake people*
2. **Art and design:** *AI-generated paintings*
3. **Medical imaging:** *Generate synthetic data for rare diseases*
4. **Super-resolution:** *Upscaling low-quality images*
5. **Deepfakes:** *Face swapping, voice cloning (ethical concerns)*

next slide →

Challenges and Limitations

- **Mode collapse:** *Generator produces limited variety*
- **Training instability:** *Hard to balance G and D*
- **Computational cost:** *Needs a lot of GPU power*
- **Evaluation metrics:** *No perfect way to judge quality*
- **Hyperparameter Sensitivity:** *Small changes, big effects*
- **Overfitting Risks:** *May memorize training data*

next slide →

Loyola University Chicago

Ethics & Risks of GANs

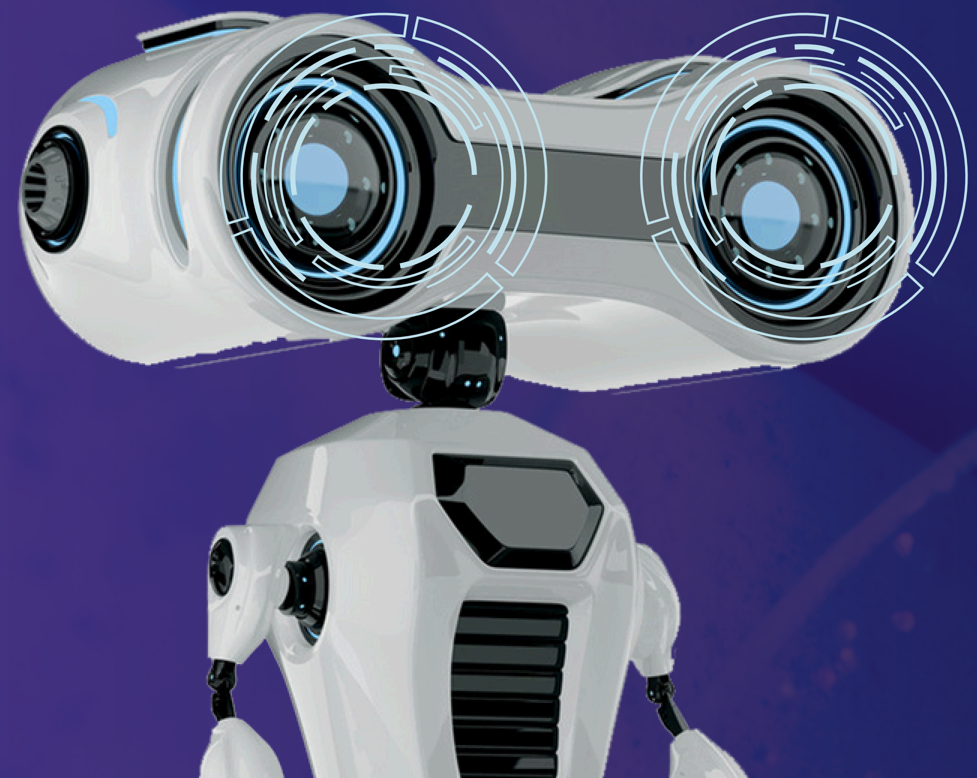
- **Deepfakes & Misinformation:** *Used to spread fake videos/images*
- **Copyright & Originality:** *Legal gray area for AI-generated content*
- **Bias in Outputs:** *Reflects biases in training data*
- **Privacy Concerns:** *GANs can unintentionally reproduce real faces*
- **Misuse Potential:** *Identity fraud, political manipulation, scams*

next slide →

Loyola University Chicago

The Future of GANs

- **Controllable Generation:** *Generate based on prompts or conditions*
- **Better Evaluation Metrics:** *More reliable scoring of quality*
- **Hybrid Models:** *GANs + Diffusion or Transformers*
- **Smarter Data Augmentation:** *Improve small dataset performance*
- **Creative Collaboration:** *Tools for artists, designers, and musicians*



next slide →

Let's Talk About It!

"Can GANs replace human creativity?"

"Would you buy art created by AI?"

"What safeguards should be in place for deepfakes?"

Conclusion

- *GANs are a powerful and creative branch of machine learning.*
- *They use two neural networks (Generator vs. Discriminator) in a game-like setup.*
- *Capable of generating realistic images, art, and more — but also come with ethical challenges.*
- *As GANs continue to evolve, so does their impact on industries, society, and the line between real and fake.*