University of Trieste
Architecture and Engineering Department

Master Degree in Computer Science

Generative Adversarial Network

Stefano Chen
IN2000246

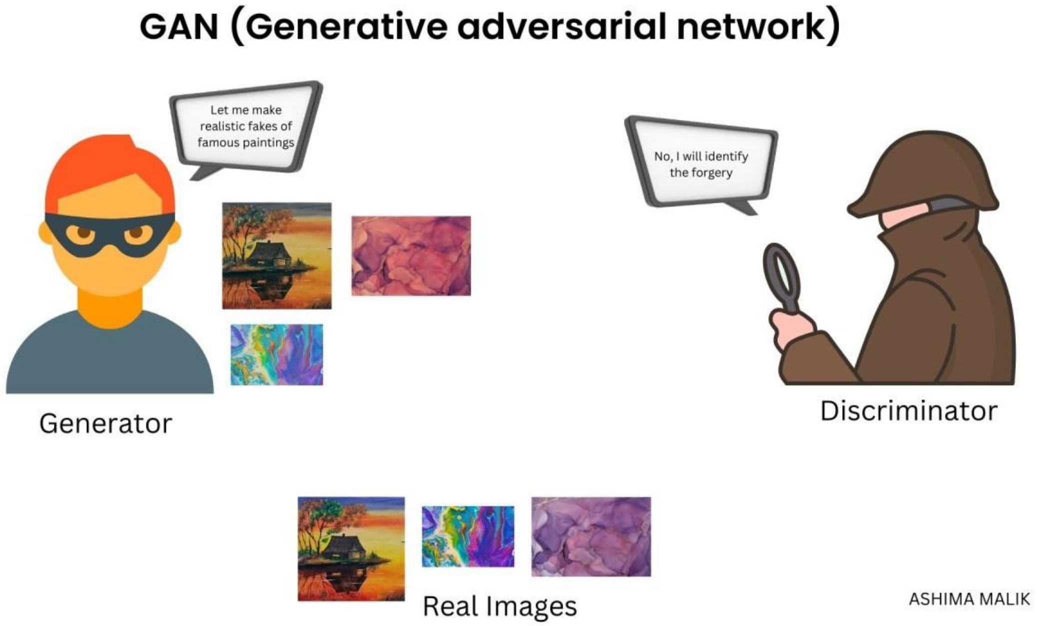
Agenda

- ❑ GAN: The Core Idea
- ❑ How They Work
- ❑ Pros & Cons
- ❑ The GAN Family
- ❑ Experiment: DCGAN

The Core Idea - An Adversarial Game

Two Neural Networks in a Creative Battle:

- ❑ Generator (G)
  - The “Forger”
  - Its job is to create synthetic data (e.g. images) that looks real
- ❑ Discriminator (D)
  - The “Art Critic”
  - Its job is to look at an image and decide if it’s authentic or a forgery



A Zero-Sum Game:

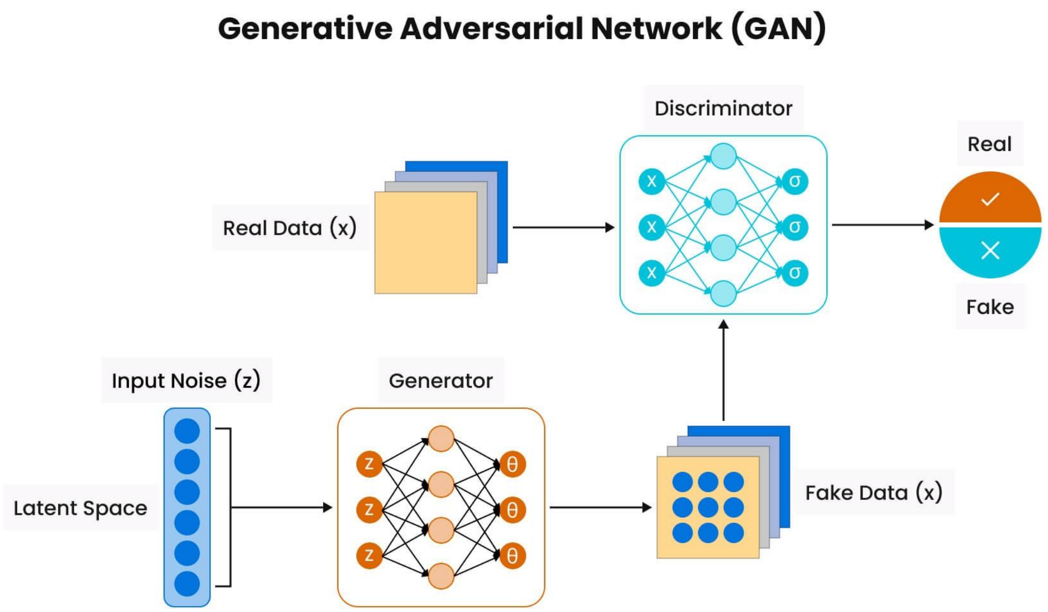
The Generator wants to fool the Discriminator.
The Discriminator wants to correctly identify fakes.
This constant competition forces both to become improve

How GANs Work

The Training Process: A Two-Player Minimax Game

- ❏ Generator (G)
  - Input: Random noise vector z
  - Output: Fake sample G(z)
- ❏ Discriminator (D)
  - Input: Real sample x (from dataset) OR Fake sample G(z)
  - Output: D(x) or D(G(z)) probability input is real
- ❏ Loss Function

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



Advantages

- **High-Quality Output**
  - using the right architecture you can generate high-quality and high-resolution data
- **Data Augmentation**
  - In cases with limited labeled data, GANs can be used to produce synthetic examples.
- **Versatility in Applications**
  - GANs can be used across various data types (e.g. Images, Audio, Video)

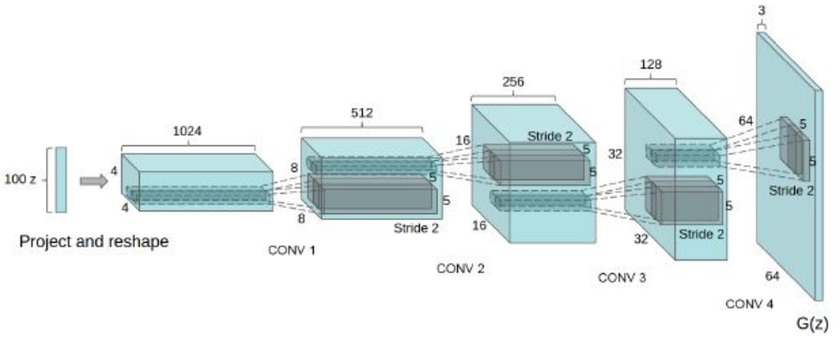
Disadvantages

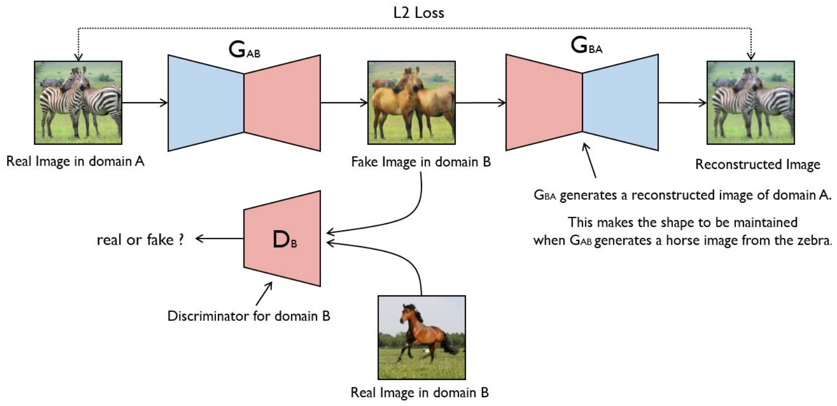
- **Training Instability**
  - Model Collapse (G produces limited variety of outputs)
  - Non-Convergence (G and D don’t reach equilibrium)
  - Vanishing Gradient (D gets too strong)
- **Sensitivity**
  - GANs are sensitive to architecture, loss functions, and hyperparameters
- **Data-Hungry**
  - requires large amount of training data
- **Uncontrolled Generation**
  - There’s no way to generate a desired data

The GAN Family

To address some of the disadvantages, different GAN architectures have been designed over the years:

- DCGAN (Deep Convolutional GAN)
  - Used Convolutional Layers instead of Fully Connected ones
  - More Stable training and better image quality
- CGAN (Conditional GAN)
  - Adds a condition to both G and D
  - Allows for controlled generation
- CycleGAN
  - Performs unpaired image-to-image translation
- StyleGAN
  - High resolution Images
  - Introduce style-based control





Our Experiment

Experiment Goal

Following the DCGAN original paper, we want to successfully train a DCGAN to generate novel and plausible animal faces

The DCGAN paper provide some Key Architectural Guidelines w.r.t original GAN architecture:

- No pooling layers: strided convolution for D and transposed convolution for G
- Batch Normalization: in both G and D except G’s output and D’s input
- Activation Function: ReLU in Generator, LeakyReLU in Discriminator

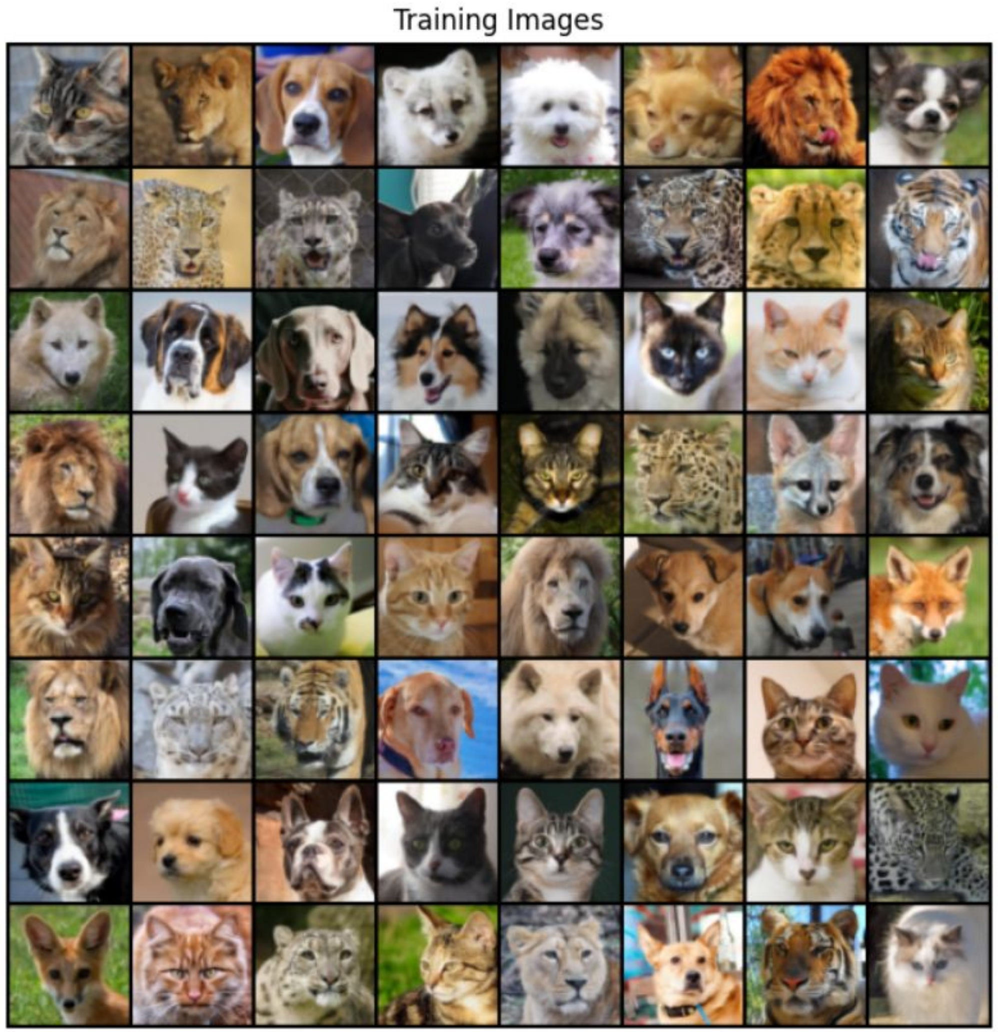
Experiment Setup

Dataset

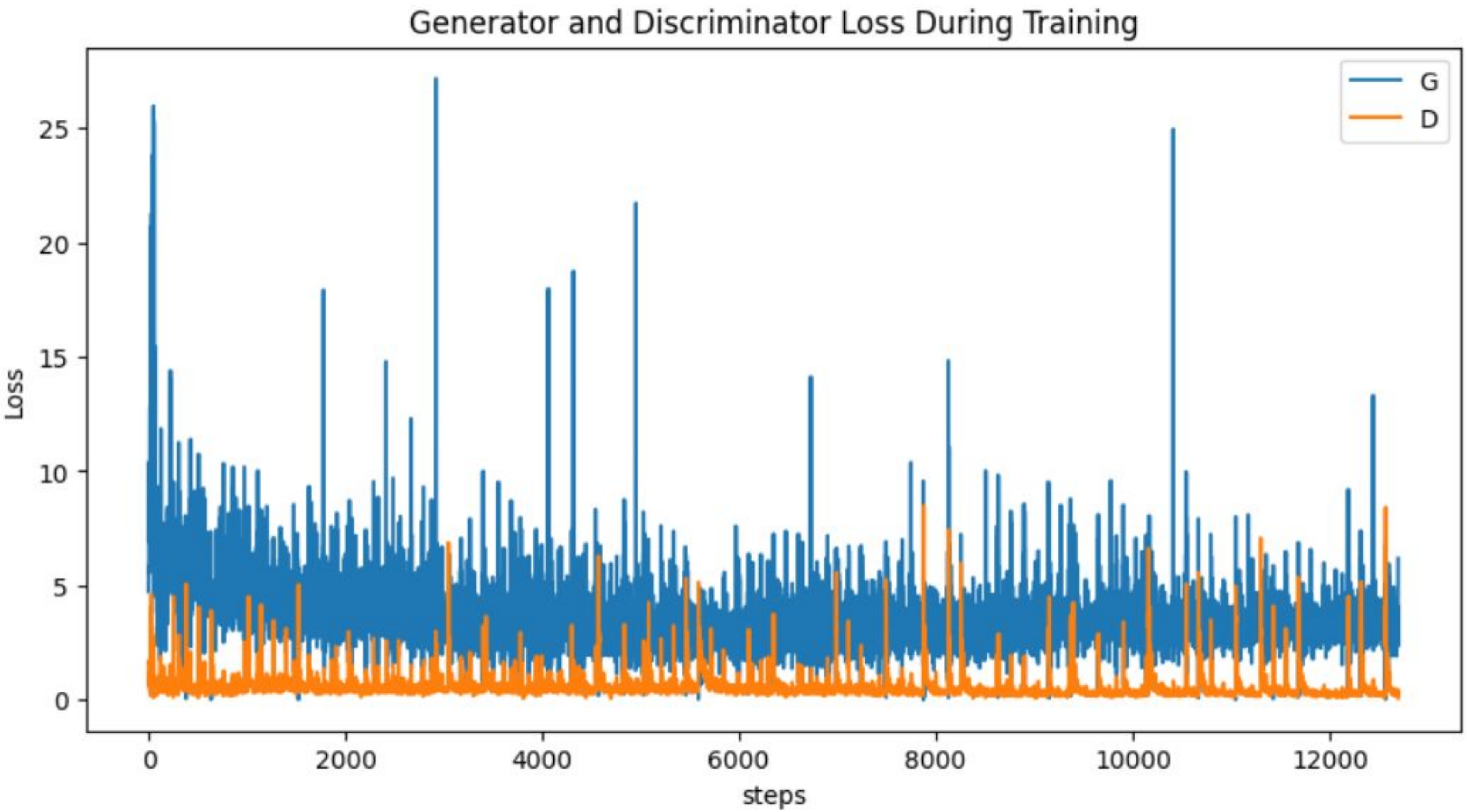
- 16 130 images (512x512)
- resized to 64x64 pixels
- normalized pixel values to [-1, +1]

Training Details

- Framework: Pytorch
- Optimizer: Adam [lr = 0.002, beta1=0.5]
- Batch size: 128
- Number of Epochs: 100



Training Loss



Generated Images

Thank You

for your attention