Include 2026 Spring Computer Vision Seminar

Generative Adversarial Networks (GAN)

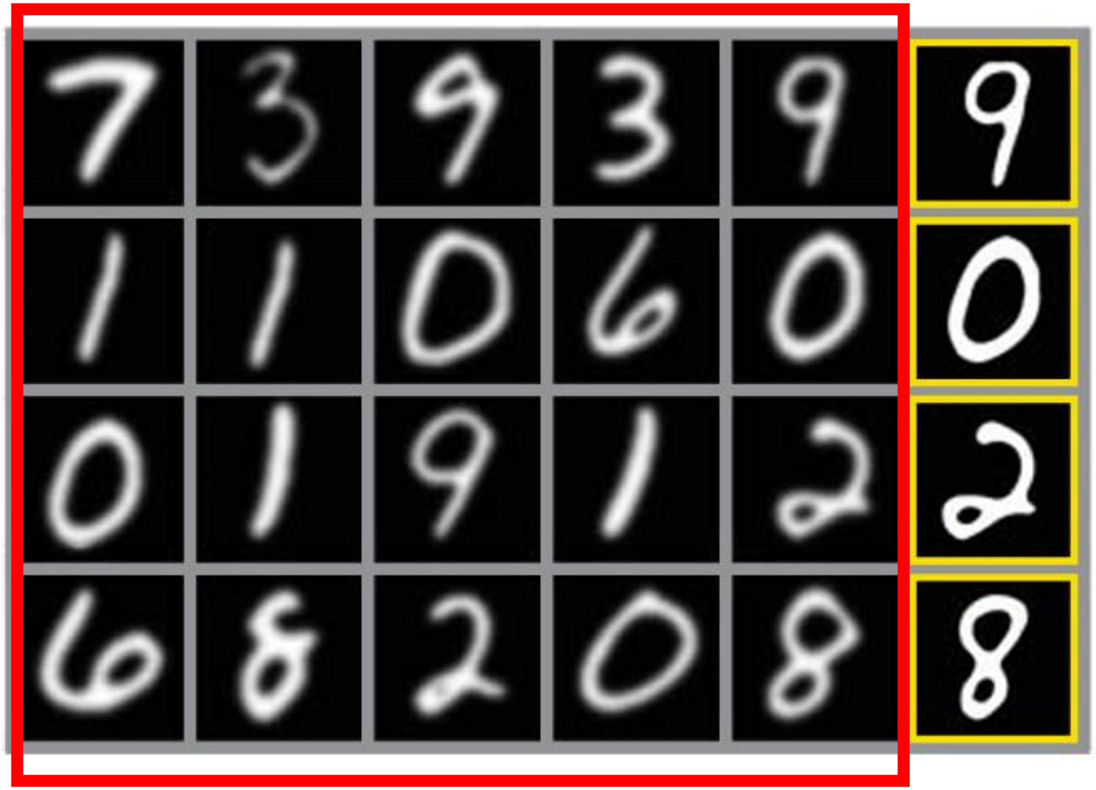
MinJae Choi minjaechoi.cs@gmail.com

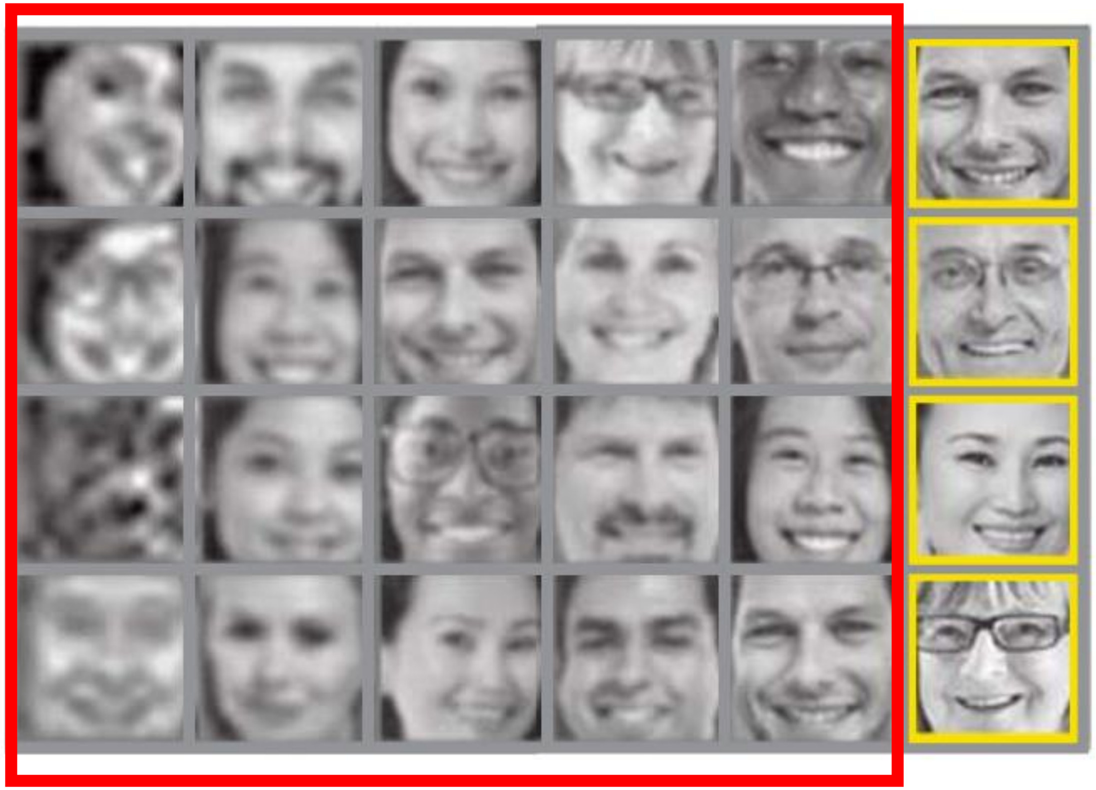
GAN

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie,[*] Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,[†] Aaron Courville, Yoshua Bengio[‡]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

GAN





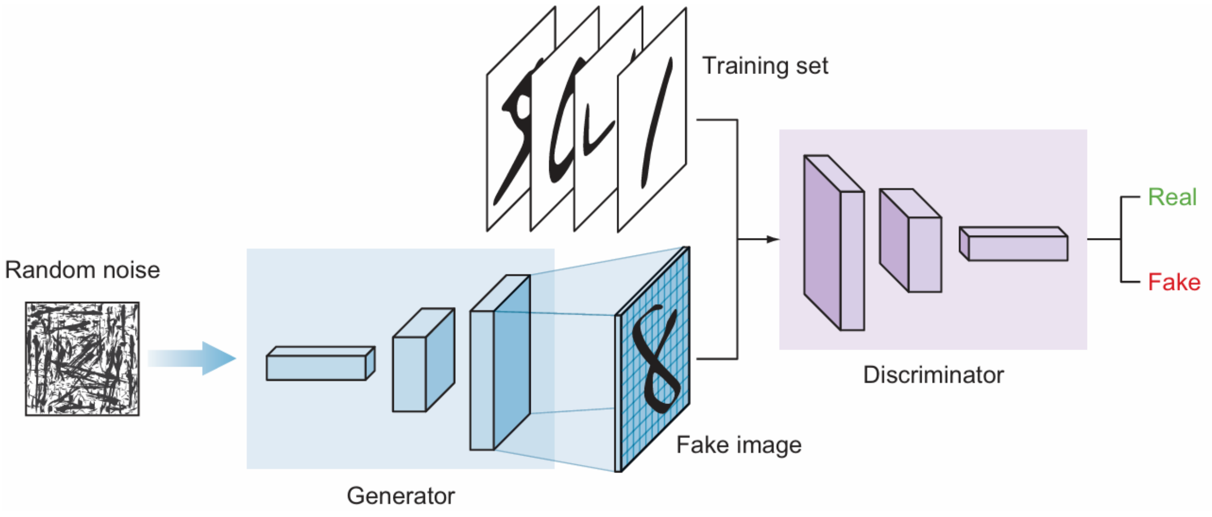
GAN

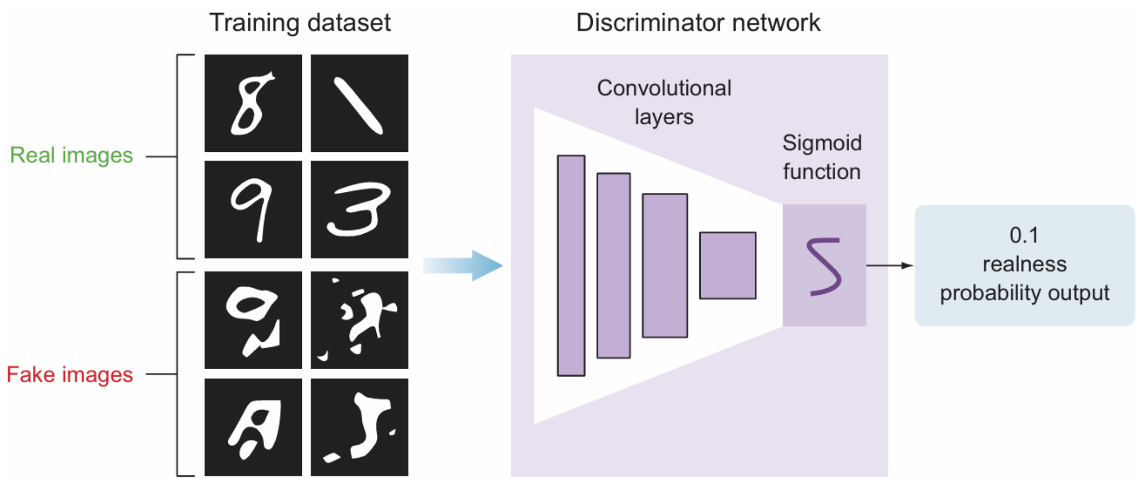
Generator

- Take random noise z and create fake image

Discriminator

- Outputs possibility of whether the image is fake or not





GAN

Objective Function

$$\min_G \max_D V(D, G) = \boxed{\mathbb{E}_x[\log D(x)]} + \boxed{\mathbb{E}_z[\log(1 - D(G(z)))]}$$

진짜 이미지 x를 보면 $D(x)$를 1에 가깝게 만들어야 함

가짜 이미지의 경우, D(G(z))를 0에 가깝게 만들어야 함

생산자 입장에서 최소화해야 함

GAN

Training

Lane1: Training Discriminator (Alone!)

- Label 1: Real Image
- Label 0: Fake Image
- Binary Classification

$$\mathbb{E}_{x \sim p_{data}}[\log D(x)]$$

Lane2: Training Generator (Combined Model!)

- The generator is trained by passing the fake images it creates through the discriminator, so that the **discriminator classifies them as real (label 1).**
- At this point, the discriminator's weights must be **frozen**.

$$\mathbb{E}_{z \sim P_z(z)}[\log(1 - D(G(z)))]$$

GAN

Evaluation

Inception Score (IS)

- Measures the **quality** of generated images using a pretrained **Inception network**
- Considers:
  - **Quality**: Images are clearly classifiable (high confidence)
  - **Diversity**: Images cover a wide range of classes
- **Higher score = better quality and diversity**

Fréchet Inception Distance (FID)

- Measures similarity between **real and generated image distributions**
- Extracts features using an **Inception network** and computes **Fréchet distance**
- **Lower score = closer to real images**
- More **robust and consistent** than Inception Score

GAN

Application

Pix2Pix (Image-to-Image Translation GAN)

- Transforms one image into another (e.g., day → night, sketch → photo)
- A type of **Conditional GAN (cGAN)**
- Uses an input image as a condition to generate a corresponding output image

SRGAN (Super-Resolution GAN)

- Converts **low-resolution images → high-resolution images**
- Generates realistic textures instead of simple pixel upscaling
- Used in **medical imaging, satellite analysis, and photo restoration**

DCGAN (Deep Convolutional GAN)

- Replaces the original GAN's **MLP with CNN**

Key design changes:
- Remove pooling → use **strided convolutions**
- Apply **Batch Normalization** in both generator and discriminator
- Use **LeakyReLU** (discriminator) and **tanh** (generator output)
- Minimize fully connected layers

Architecture:
- **Generator**: Starts from a small feature map (e.g., 7×7×128) and upsamples to generate images (inverted CNN)
- **Discriminator**: Downsamples the image and outputs a single probability (real/fake)

Challenges in Training GANs

- GAN training is a **non-convex optimization problem** in a **high-dimensional parameter space**

- The goal is to reach a **Nash equilibrium**, where:
  - The generator produces realistic data
  - The discriminator cannot distinguish real from fake

- However, standard **gradient descent** is designed to:
  - Minimize a **single objective function**, not solve a game

- As a result:
  - **Convergence is not guaranteed**
  - Training may exhibit **instability, oscillation, or mode collapse**

Improved Techniques for Training GANs

Improved Techniques for Training GANs

Tim Salimans tim@openai.com
Ian Goodfellow ian@openai.com
Wojciech Zaremba woj@openai.com
Vicki Cheung vicki@openai.com
Alec Radford alec.radford@gmail.com
Xi Chen peter@openai.com

Abstract

We present a variety of new architectural features and training procedures that we apply to the generative adversarial networks (GANs) framework. We focus on two applications of GANs: semi-supervised learning, and the generation of images that humans find visually realistic. Unlike most work on generative models, our primary goal is not to train a model that assigns high likelihood to test data, nor do we require the model to be able to learn well without using any labels. Using our new techniques, we achieve state-of-the-art results in semi-supervised classification on MNIST, CIFAR-10 and SVHN. The generated images are of high quality as confirmed by a visual Turing test: our model generates MNIST samples that humans cannot distinguish from real data, and CIFAR-10 samples that yield a human error rate of 21.3%. We also present ImageNet samples with unprecedented resolution and show that our methods enable the model to learn recognizable features of ImageNet classes.

Improved Techniques for Training GANs

1. **Feature Matching**

- Instead of directly **fooling the discriminator**(Too simple!), the generator is trained to
  match the **statistics of intermediate features** from real data

- **Objective**: Match the mean of discriminator features for real vs. generated data

- **Effect**:
  - Prevents overfitting to the current discriminator
  - Leads to **more stable training**

Improved Techniques for Training GANs

2. Minibatch Discrimination

- Standard discriminator evaluates images independently → causes mode collapse
- This method allows the discriminator to look at the entire batch
- It checks whether generated samples are too similar to each other

- **Effect**: Encourages diversity in generated images

Improved Techniques for Training GANs

3. Historical Averaging

- Adds a penalty if current parameters deviate too much from their historical average

- **Effect**:
  - Prevents rapid, unstable updates
  - Improves training stability

Improved Techniques for Training GANs

4. One-sided Label Smoothing

- Replace real label 1.0 → 0.9 (fake remains 0.0)
- Prevents the discriminator from becoming overconfident

- **Effect**
  - Produces smoother gradients
  - Stabilizes training

Improved Techniques for Training GANs

5. Virtual Batch Normalization

- Standard BatchNorm depends on the current batch → introduces unwanted correlations
- Uses a fixed reference batch for normalization

- **Effect**:
  - Removes inter-sample dependency
  - Improves consistency and stability

Self-Attention GAN (SAGAN)

Limitation of DCGAN

1. CNN Focuses on Local Regions

- CNN looks at **small patches** (e.g., 3×3 kernel)
- It processes images **piece by piece**, not all at once
- Even with deeper layers → Hard to directly connect **distant regions**
- **Simple idea**: “CNN sees parts, not the whole picture”

2. Works Well for Simple Structures

- CNN-based GANs perform well on: Sky, ocean, grass (simple textures)
- But struggle with:
  - **Complex structures**
  - **Geometric relationships**

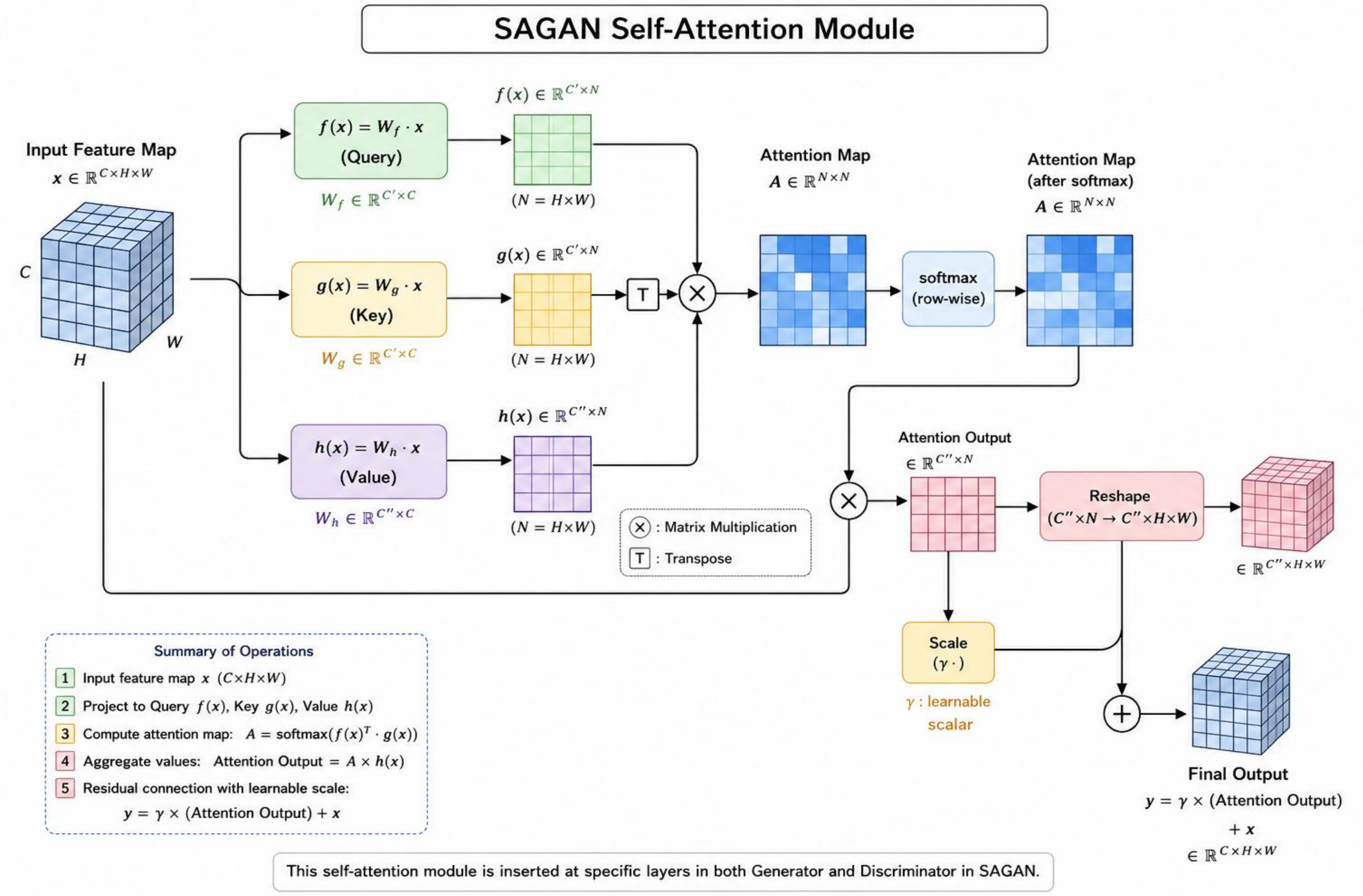
Self-Attention GAN (SAGAN)

Limitation of DCGAN

3. Difficulty Capturing Global Relationships

- Hard to model relationships between **far-apart features**
- Example:
  - A dog's **left eye and right eye should be symmetric**
  - CNN cannot easily connect these distant parts
- Result:
  → Generated images may look **structurally incorrect**
  → e.g., distorted or misaligned eyes

Self-Attention GAN (SAGAN)



GMAN

Why is a Singel Discriminator a Problem?

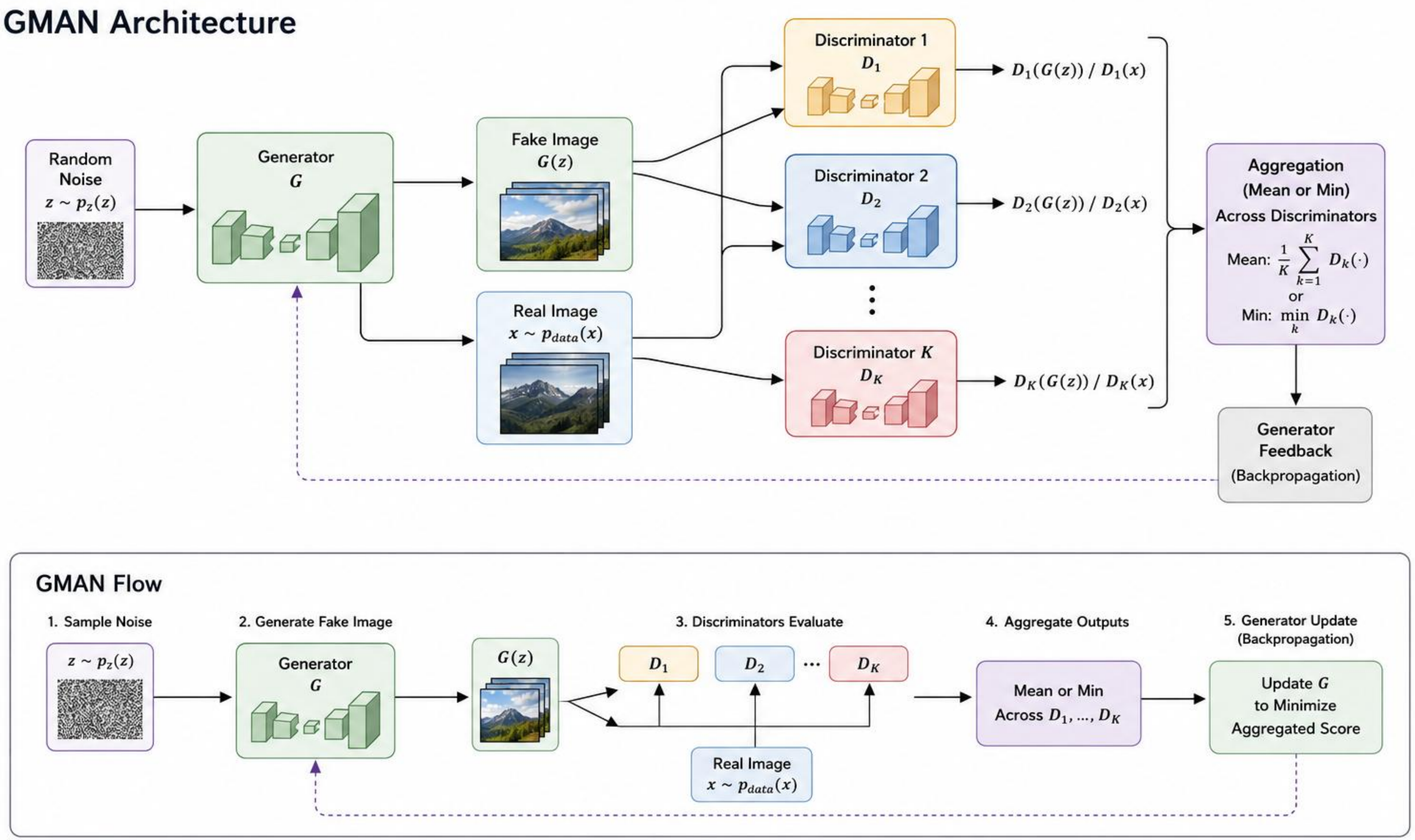
1. Too Strong or Too Weak

- If the discriminator is **too strong**:
  → Generator gets **almost zero gradients**
  → Cannot learn
- If the discriminator is **too weak**:
  → Provides **poor feedback**
  → Generator does not improve
- Simple analogy: “If the exam is too hard or too easy, students can’t learn properly”

2. Training Becomes Unstable

- GAN relies on **one discriminator only**
- Entire training depends on **its condition**
- Result: Training can **easily collapse or oscillate**

GMAN



- Multiple discriminators provide diverse perspectives.
- Aggregation (mean or min) gives a robust signal to the generator.

Mean: encourages overall fooling of all discriminators.
Min: focuses on fooling the hardest discriminator.

GMAN

GMAN: Use Multiple Discriminators

- Uses multiple discriminators instead of one
- **Key idea**: Generator learns from multiple critics
- **Benefits**:
    - More stable training
    - More diverse and informative feedback
    - No need to modify the original minimax objective

GMAN

GMAN: Use Multiple Discriminators

Strong Adversary (Minimum Strategy)

- Use the **minimum output** among multiple discriminators
- Idea: Follow the **most strict discriminator**
- Effect:
  - Generator must satisfy **all discriminators**
  - Produces more **realistic and diverse images**
- Simple idea: “Learn from the toughest critic”

Forgiving Teacher (GMAN - Average Strategy)

- Use the **average output** of multiple discriminators
- Idea: Combine **easy + strict feedback**
- Effect:
  - Provides **balanced guidance**
  - Enables **automatic curriculum learning**
    (start easy → gradually harder)
- Simple idea: “Learn step by step from multiple teachers”

Thank You