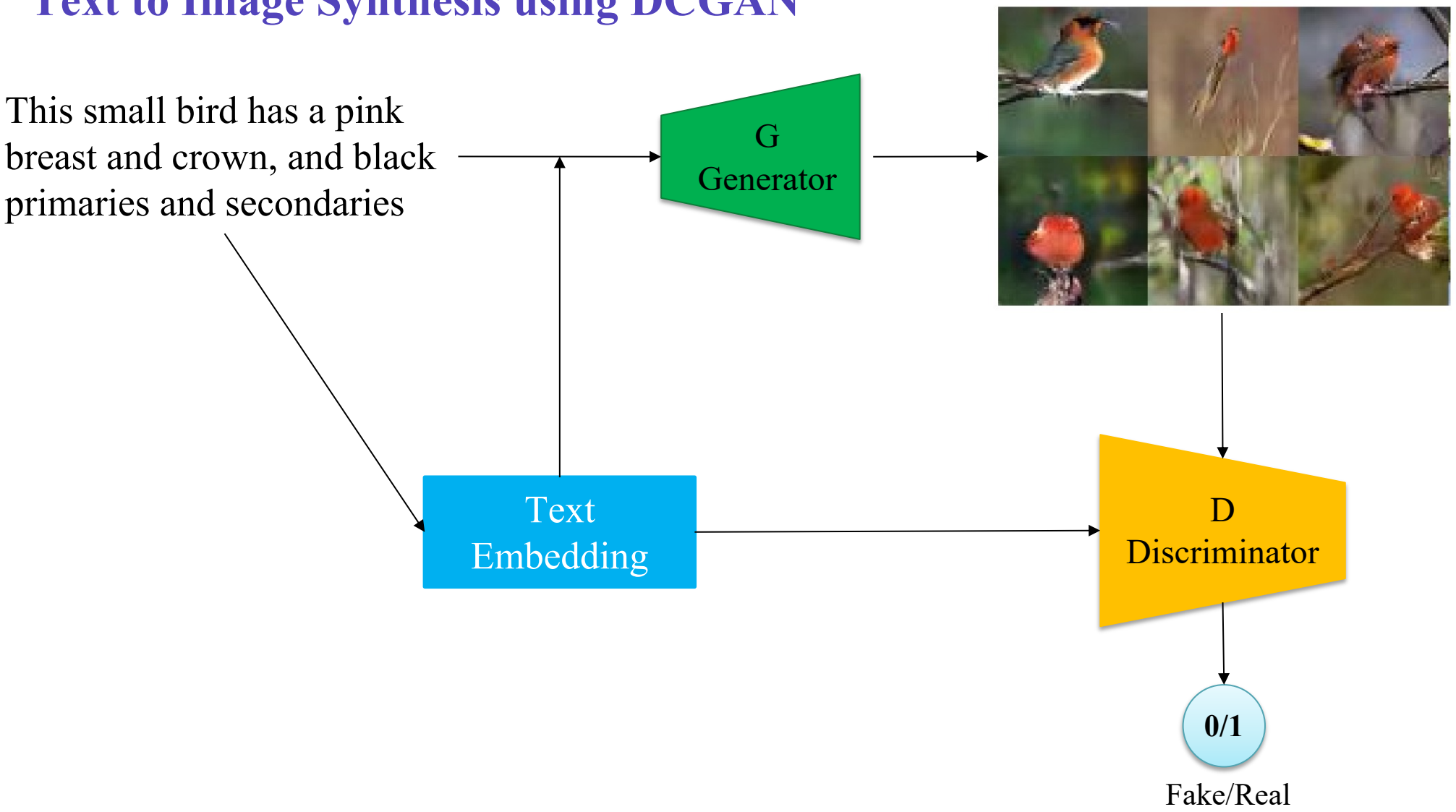
Module 9 - Exercise

Text to Image Synthesis using DCGAN

AI VIET NAM
Nguyen Quoc Thai

Objective

Text to Image Synthesis using DCGAN



Outline

- Introduction
- DCGAN
- Text to Image Synthesis using DCGAN

Introduction

Text to Image

This small bird has a pink breast and crown, and black primaries and secondaries

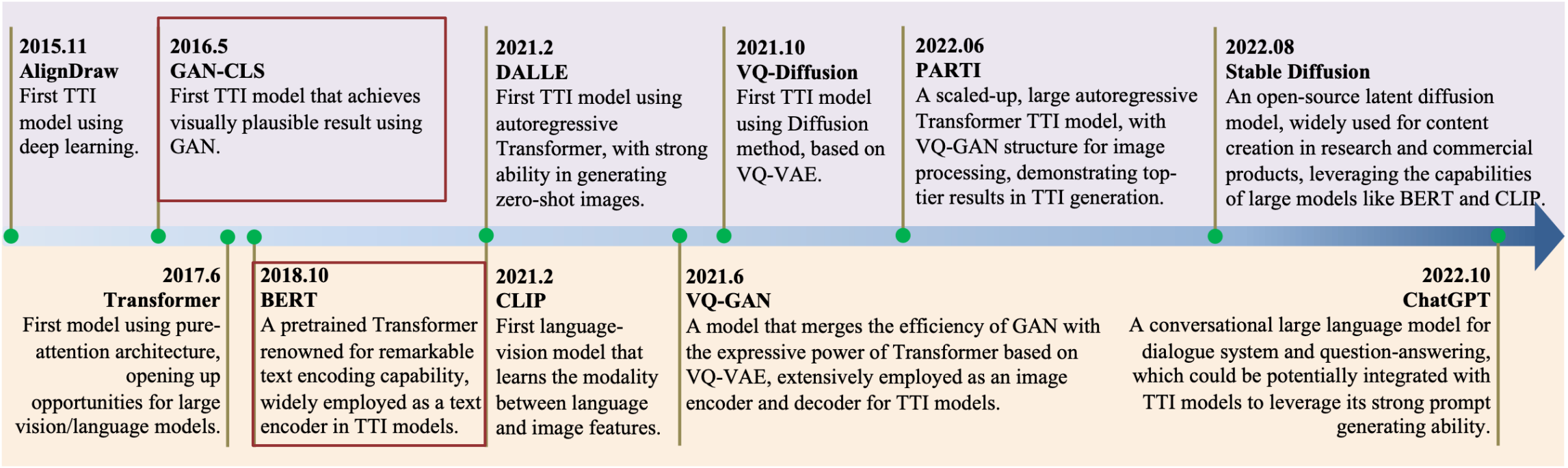
Introduction

Text to Image

The flower has petals that are bright pinkish purple with while stigma

Introduction

The milestones of text-to-image models and large models



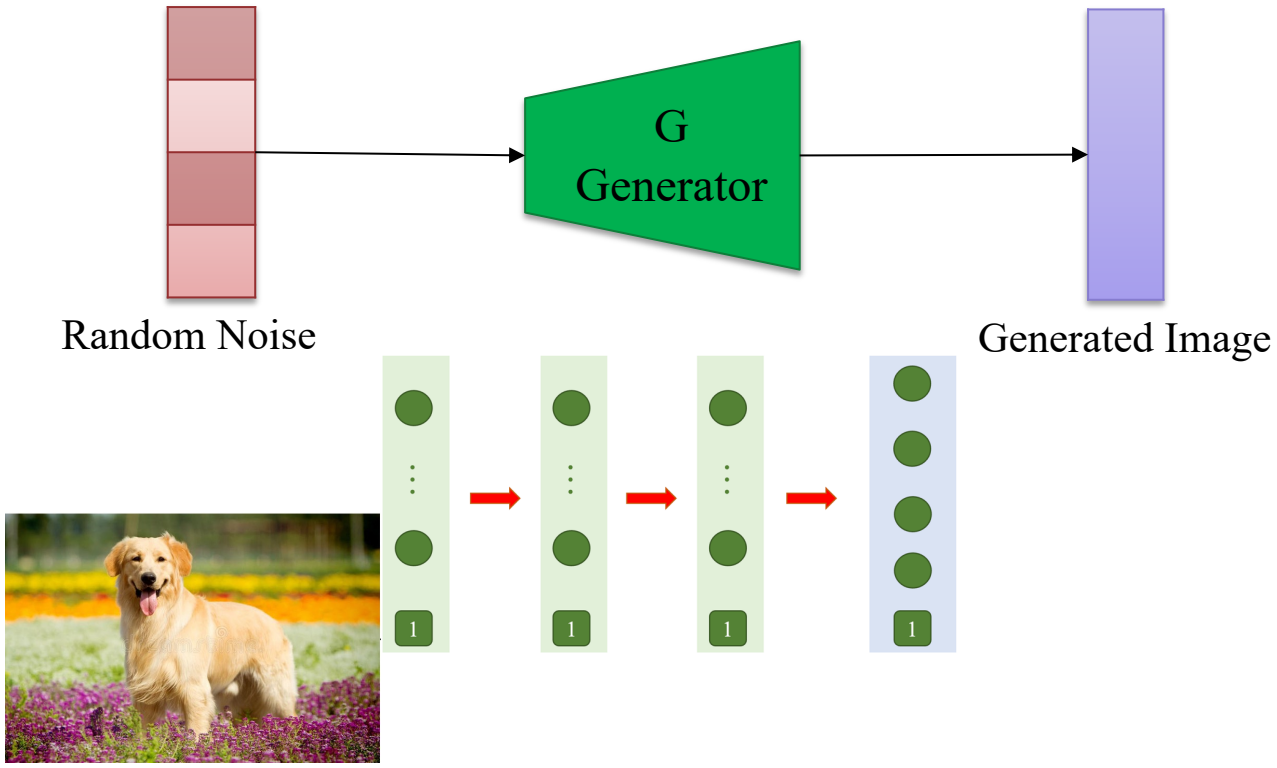
Source

Outline

- Introduction
- DCGAN
- Text to Image Synthesis using DCGAN

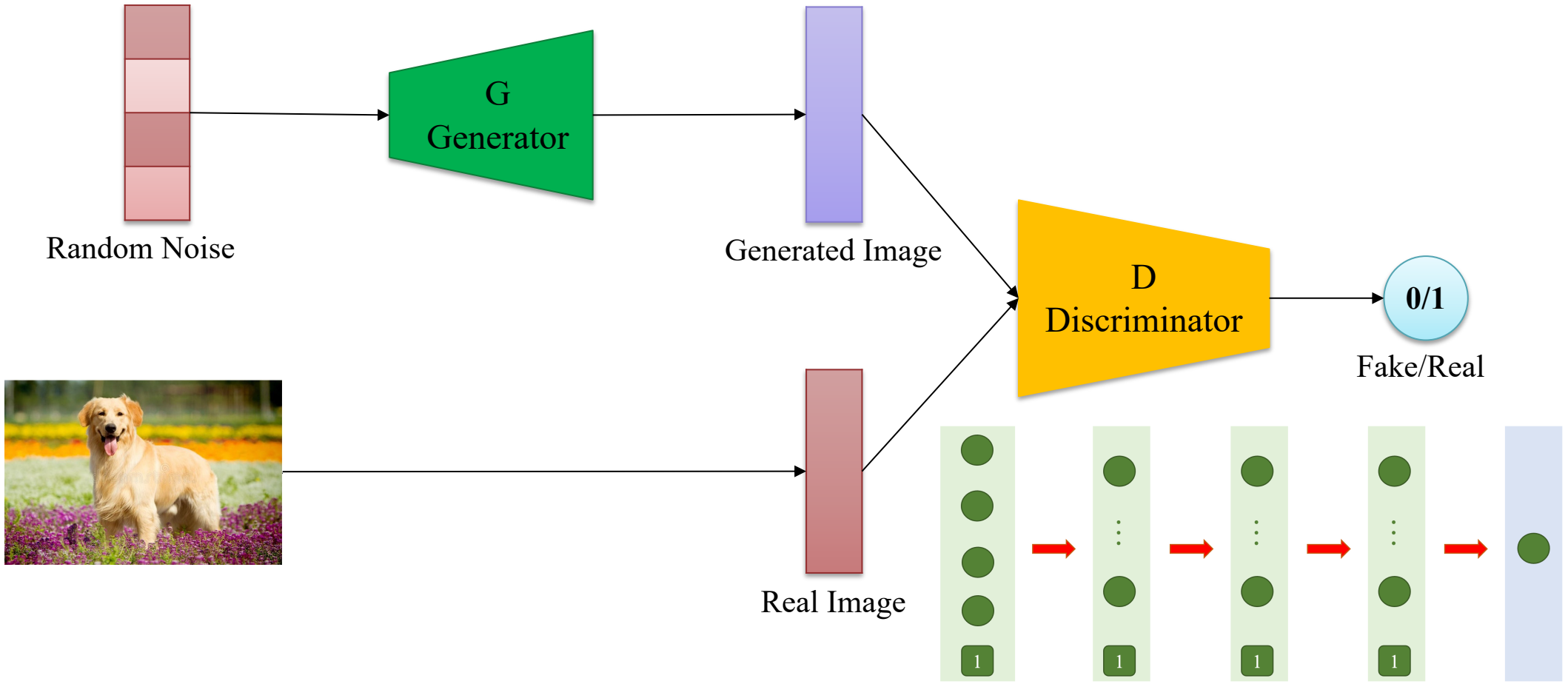
DCGAN

GAN (Generative Adversarial Networks)



DCGAN

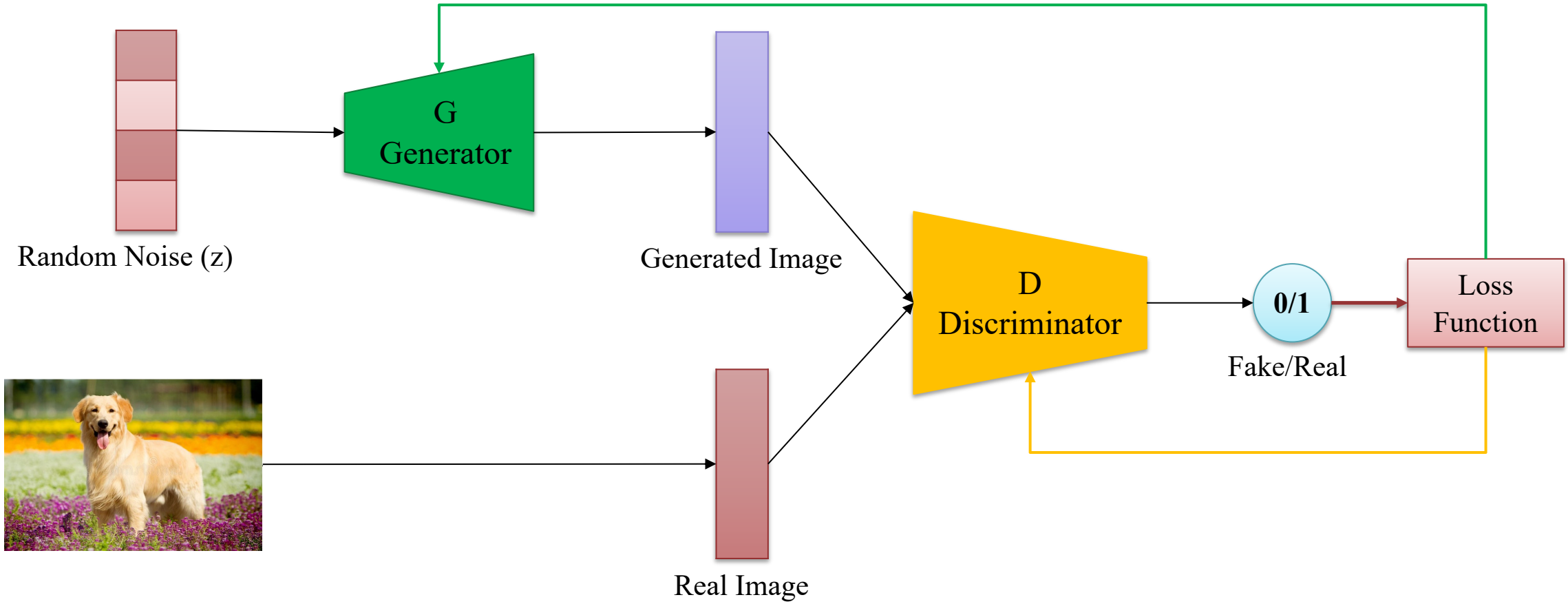
GAN (Generative Adversarial Networks)



DCGAN

GAN Training – Update G

$$L_G(z) = -\log(D(G(z)))$$

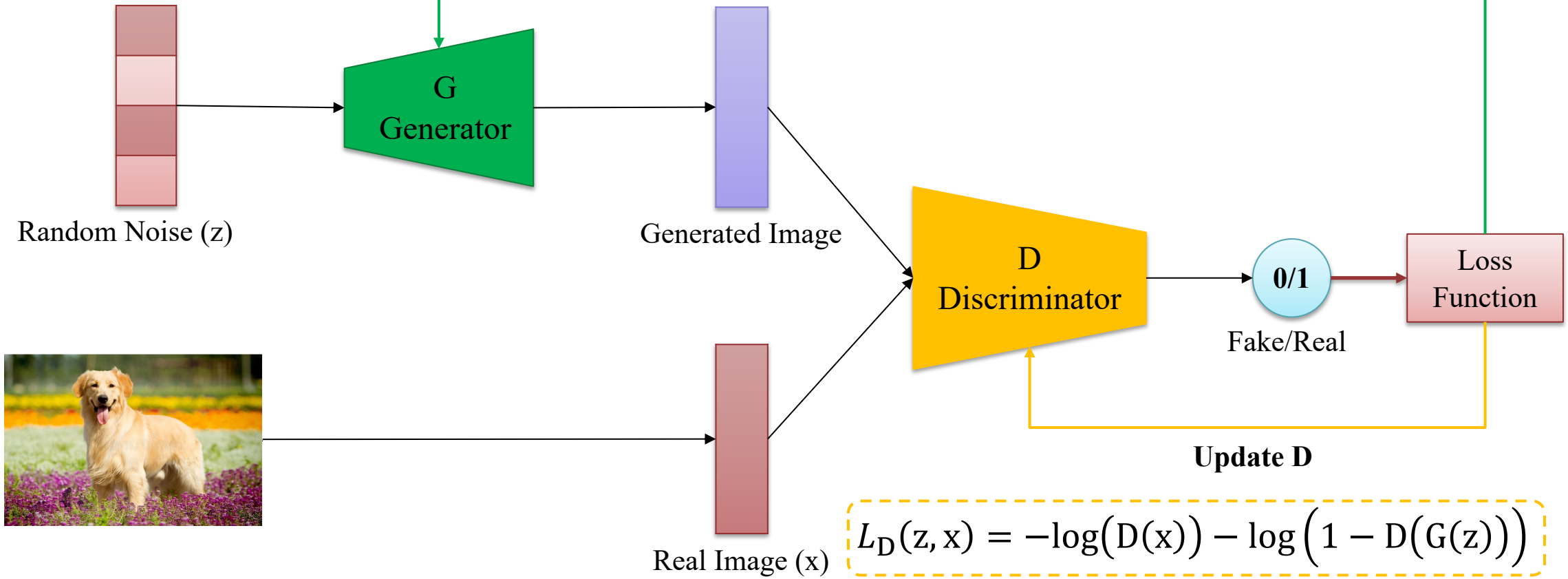


DCGAN

GAN Training – Update D

$$L_G(z) = -\log(D(G(z)))$$

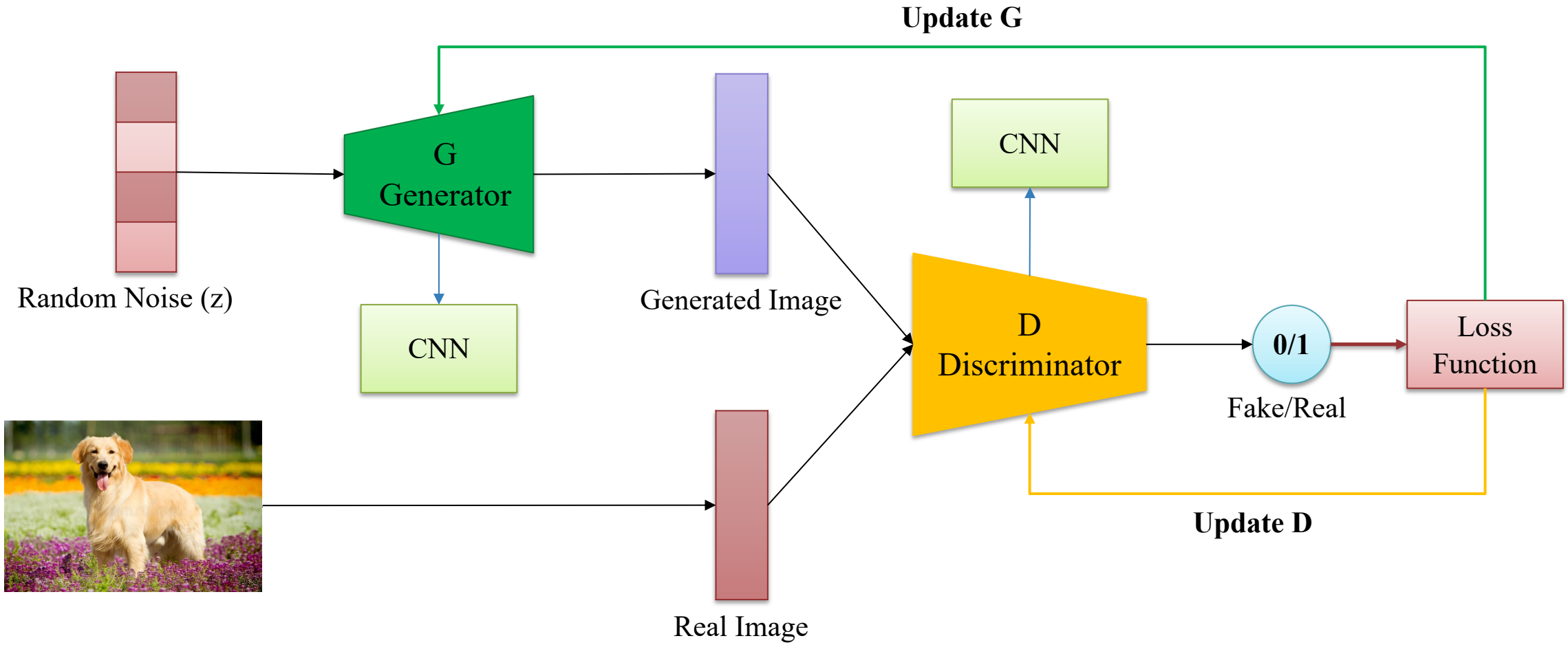
Update G



$$L_D(z, x) = -\log\big(D(x)\big) - \log\Big(1 - D\big(G(z)\big)\Big)$$

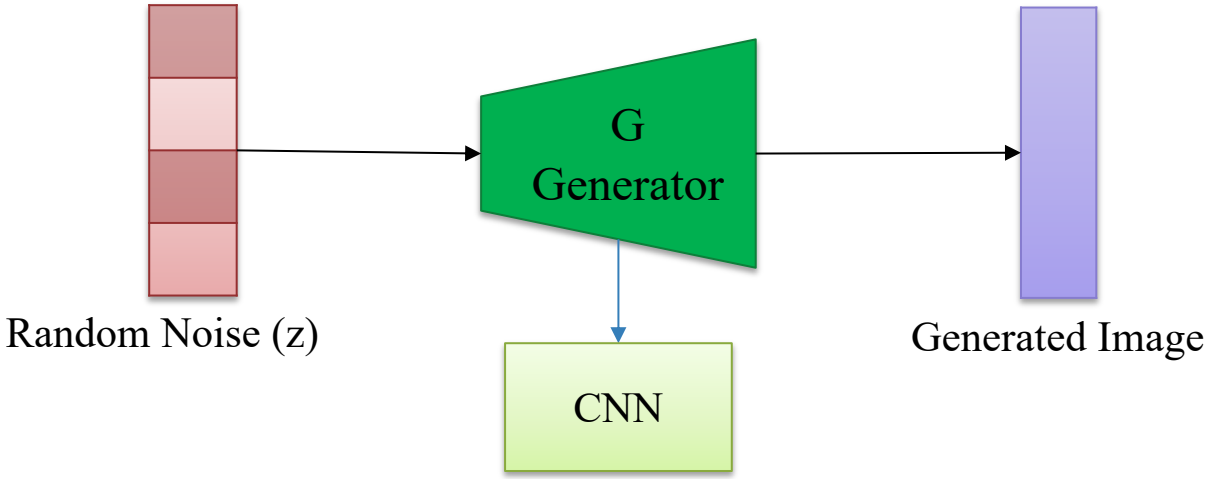
DCGAN

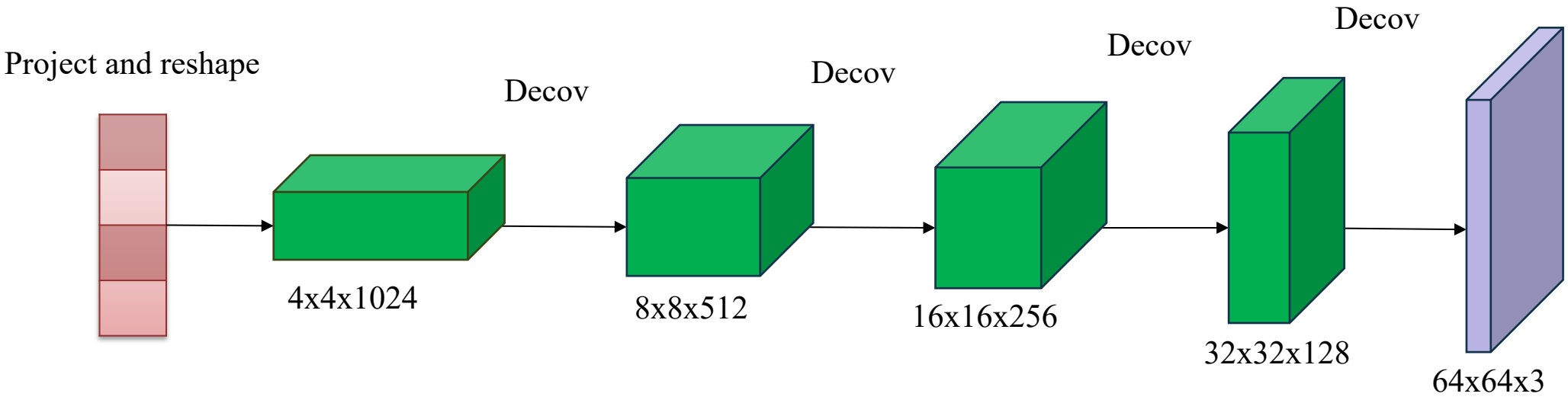
Deep Convolution GAN (DCGAN)



DCGAN

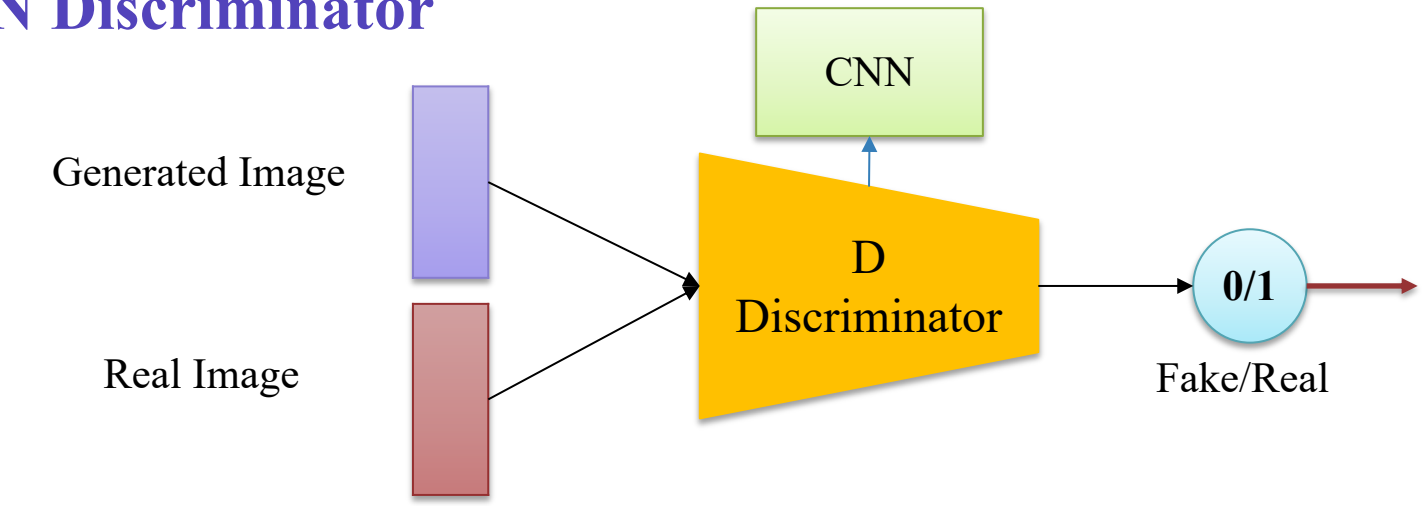
DCGAN Generator

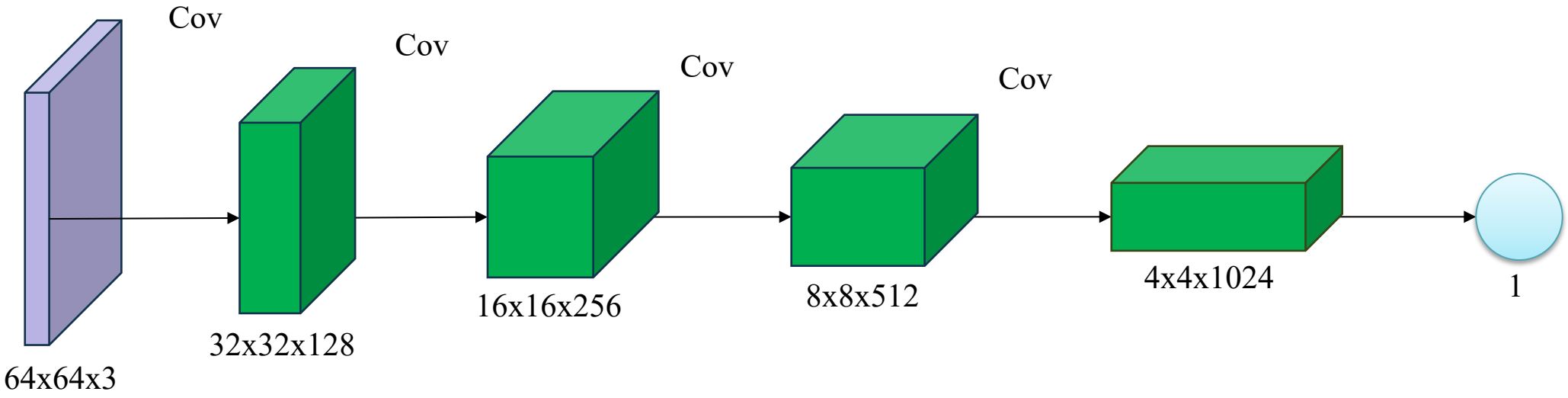




DCGAN

DCGAN Discriminator





Outline

- Introduction
- DCGAN
- Text to Image Synthesis using DCGAN

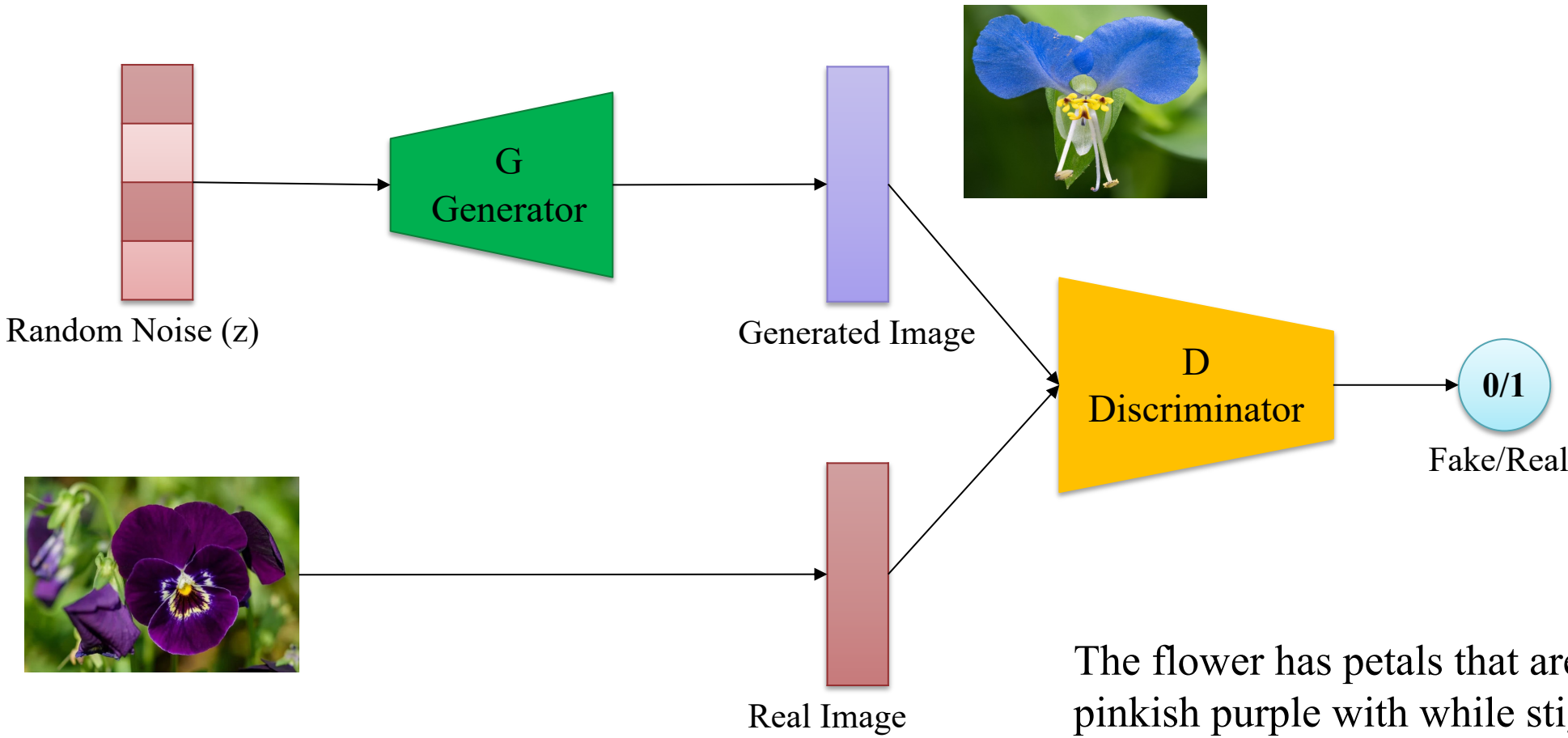
Text to Image Synthesis

Example

The flower has petals that are bright pinkish purple with while stigma

Text to Image Synthesis

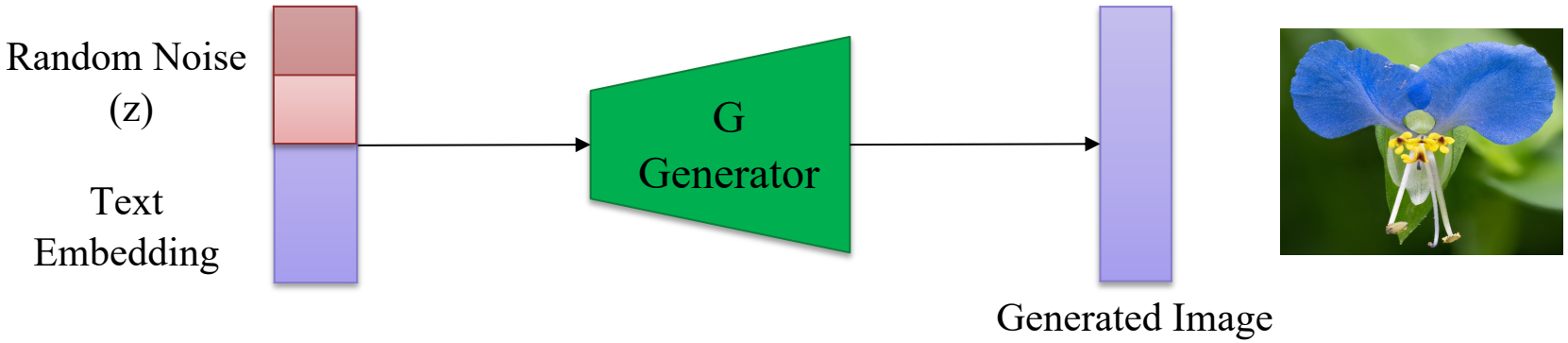
Approach

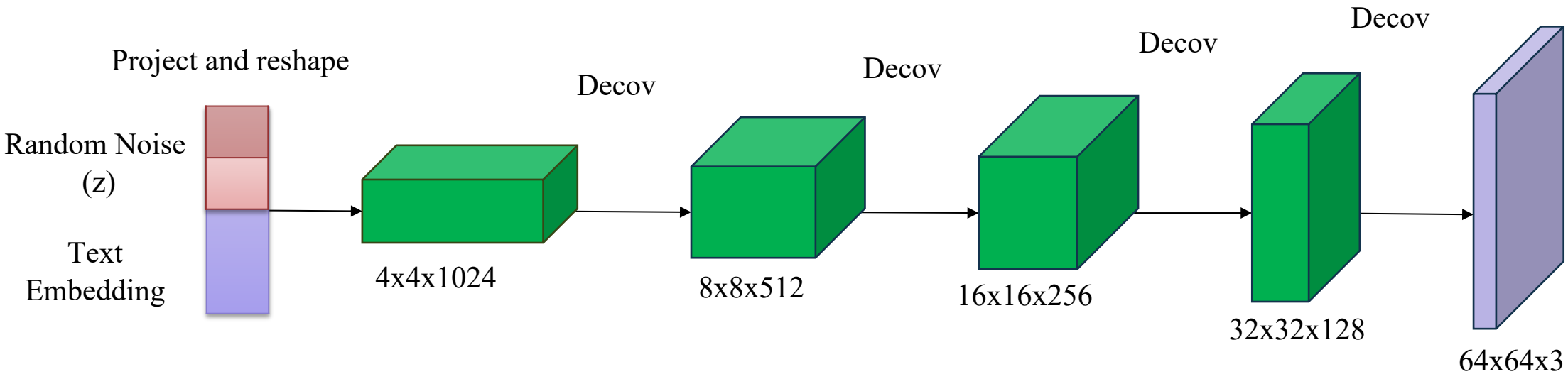


The flower has petals that are bright pinkish purple with while stigma

Text to Image Synthesis

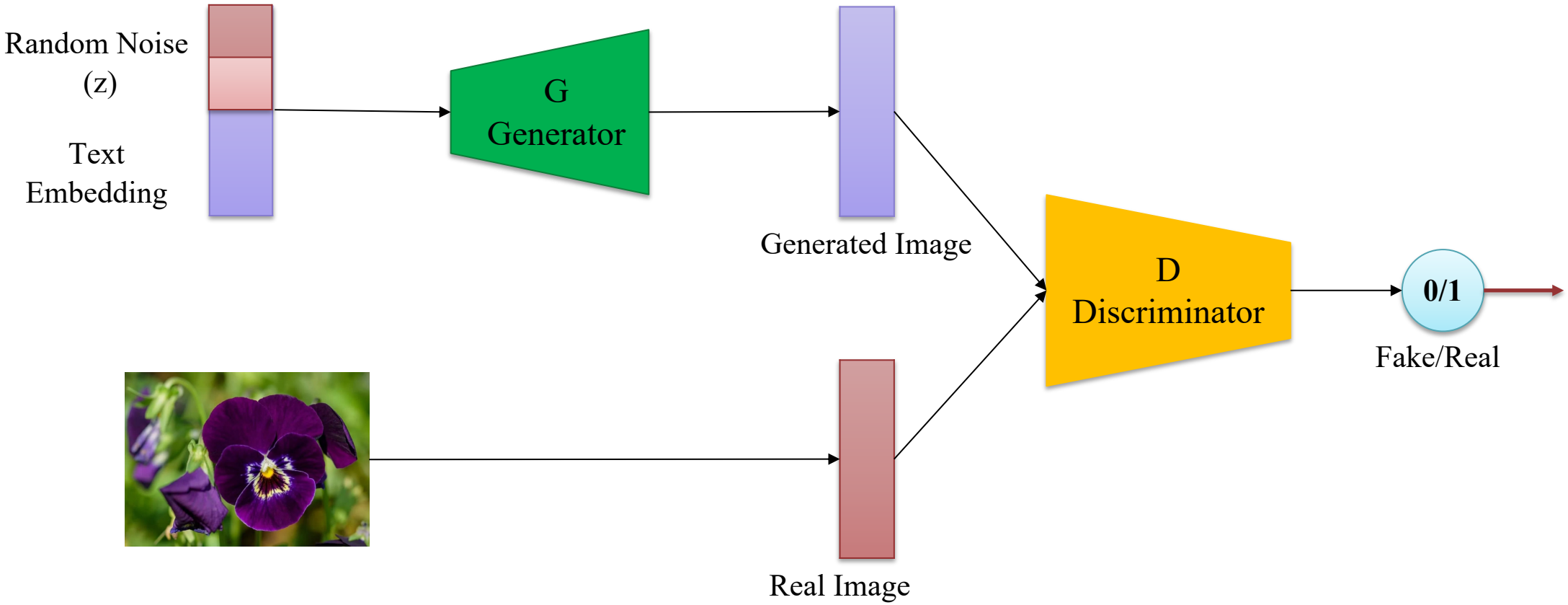
GANs with Join Distributions





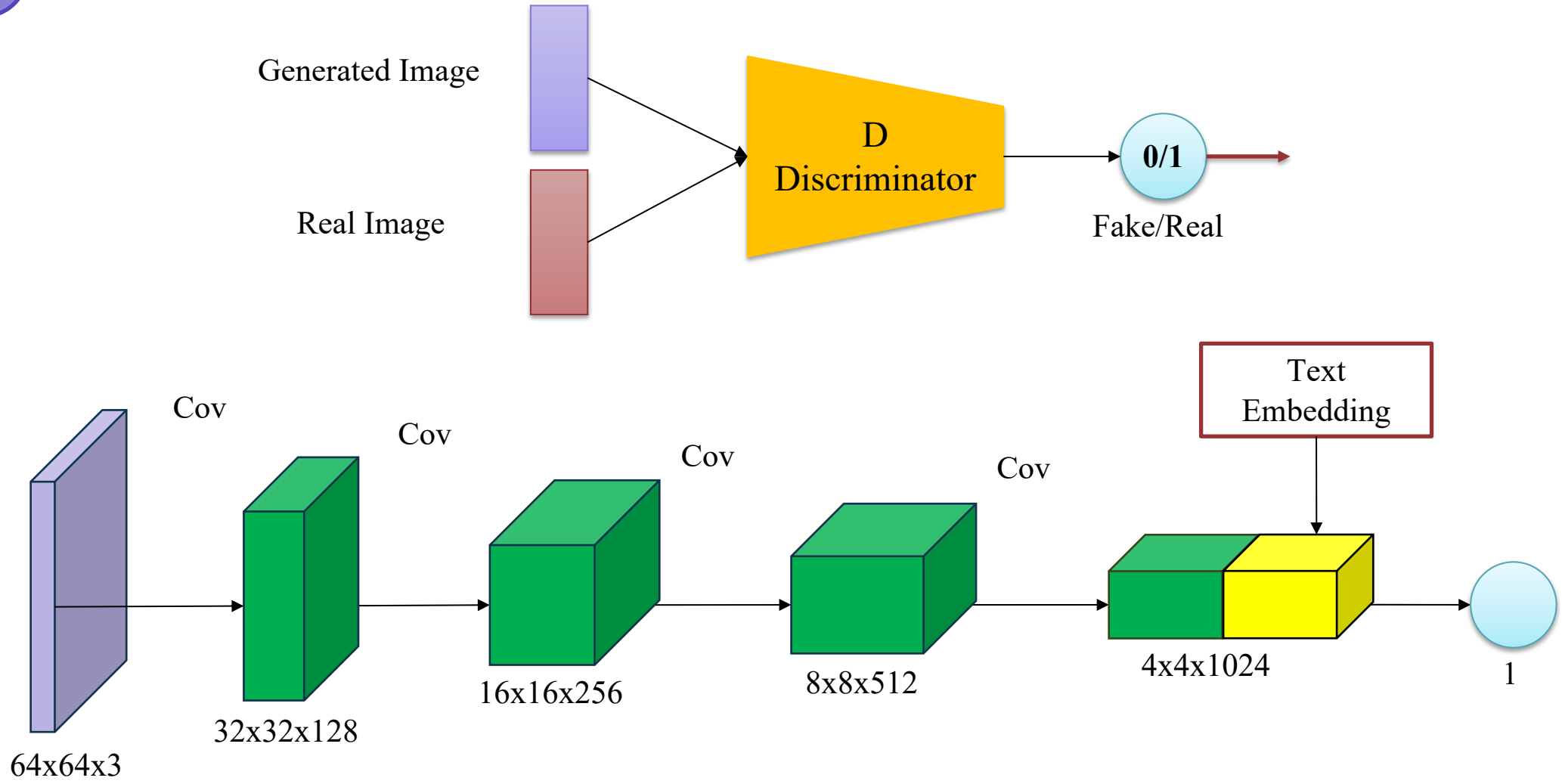
Text to Image Synthesis

GANs with Join Distributions



Text to Image Synthesis

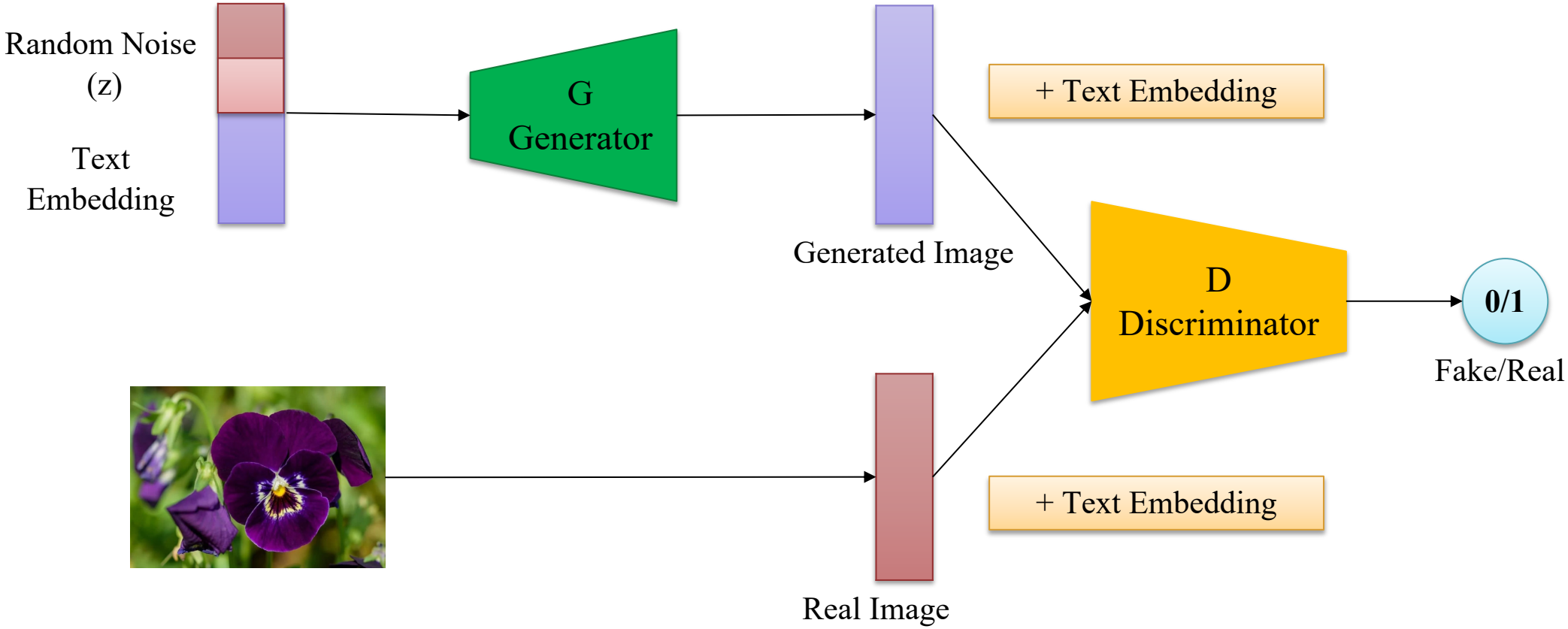
GANs with Join Distributions



Text to Image Synthesis

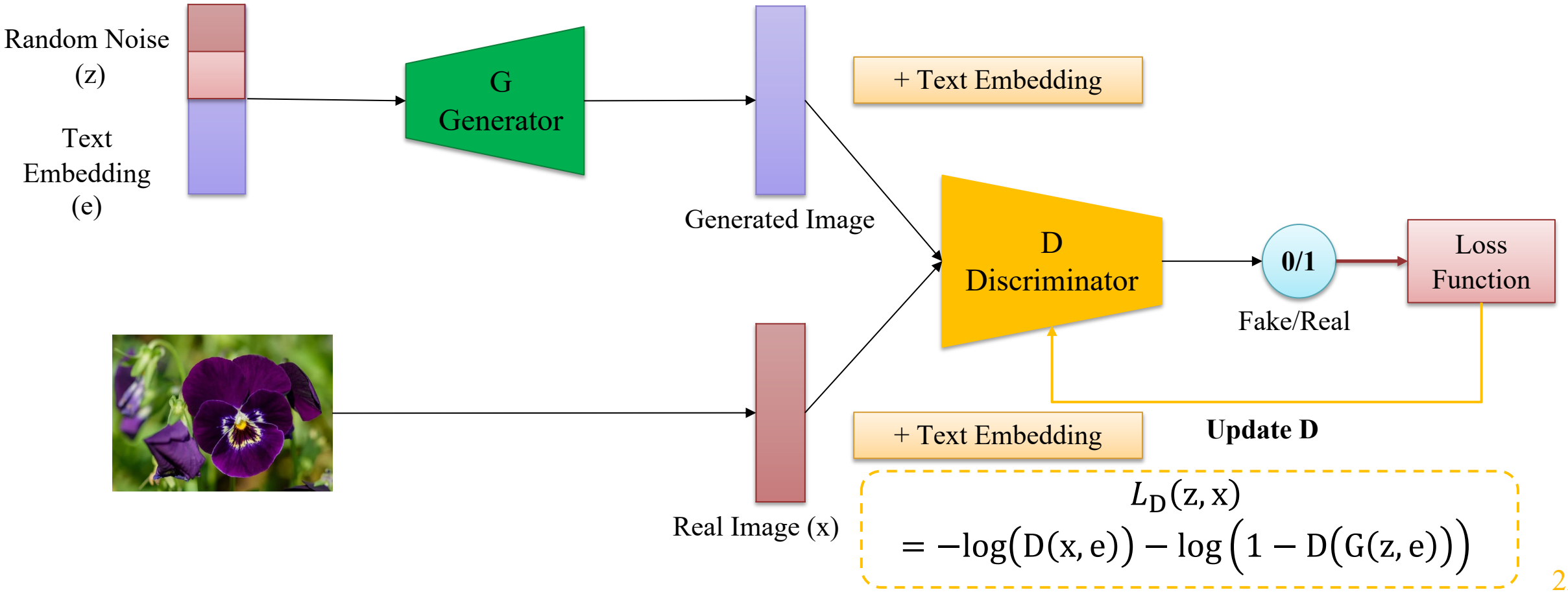
GANs with Join Distributions

The flower has petals that are bright pinkish purple with while stigma



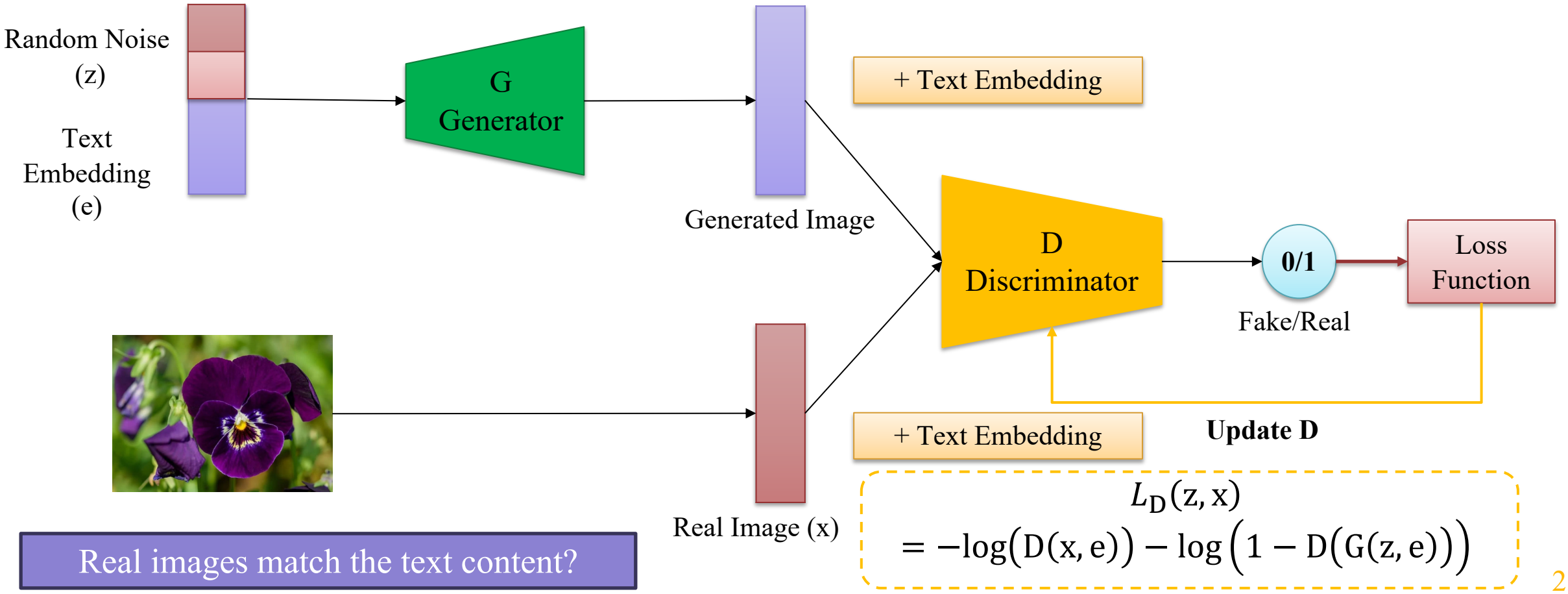
Text to Image Synthesis

GANs with Join Distributions



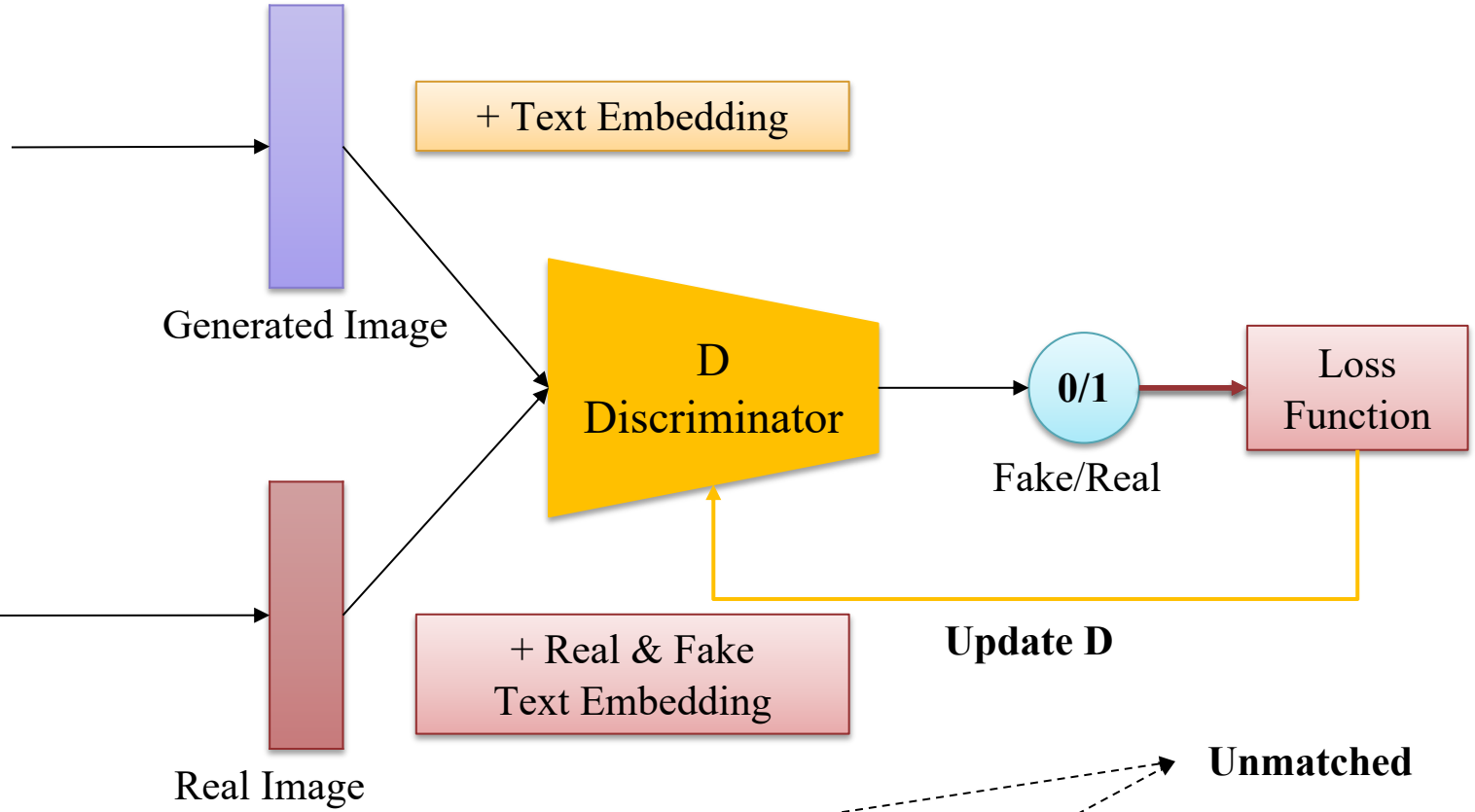
Text to Image Synthesis

GANs with Join Distributions



Text to Image Synthesis

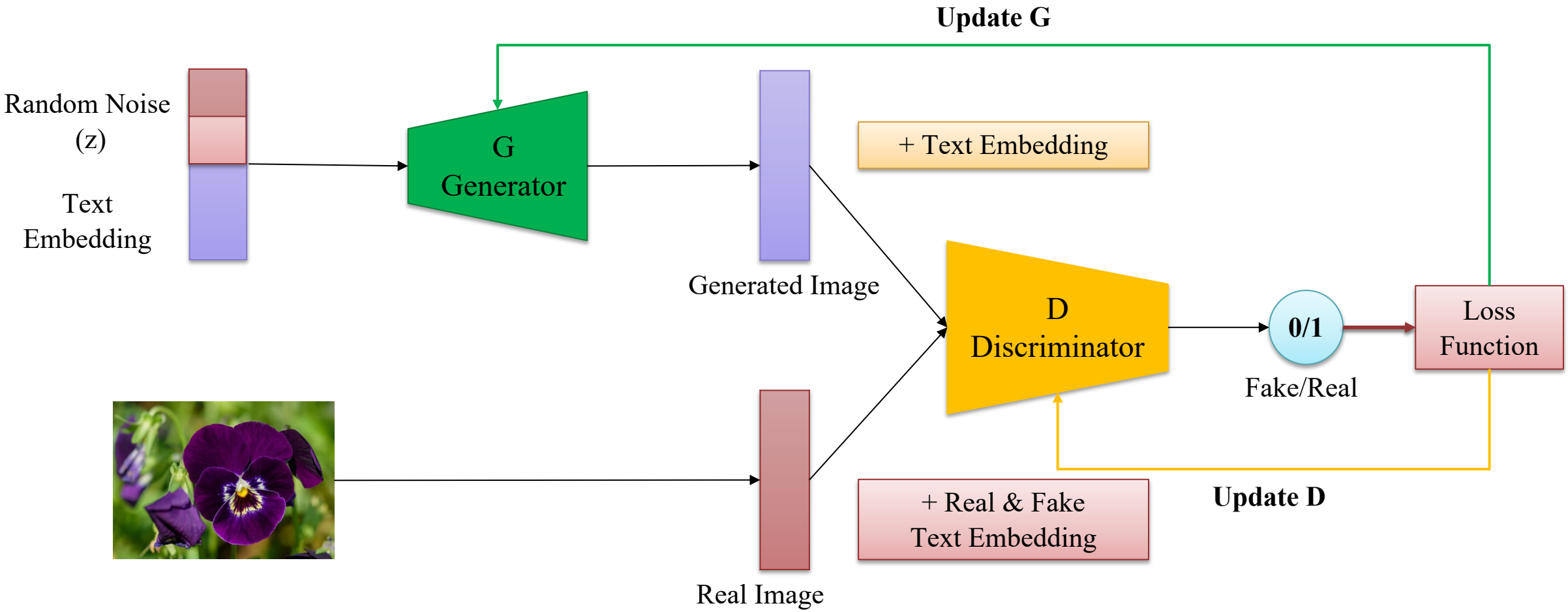
Matching-aware Discriminator (GAN-CLS)



$$L_D(z, x) = -\log\big(D(x, e)\big) - \frac{\log\Big(1 - D\big(G(z, e)\big)\Big) + \log\Big(1 - D\big(G(x, e_k)\big)\Big)}{2}$$

Text to Image Synthesis

Matching-aware Discriminator (GAN-CLS)



Text to Image Synthesis

Generator

$$L_G(z) = -\log\left(D\left(G(z, e)\right)\right)$$

Update G

Random Noise (z)

Text Embedding

G Generator

Generated Image

+ Text Embedding

D Discriminator

0/1

Fake/Real

Loss Function

Real Image

+ Real & Fake Text Embedding

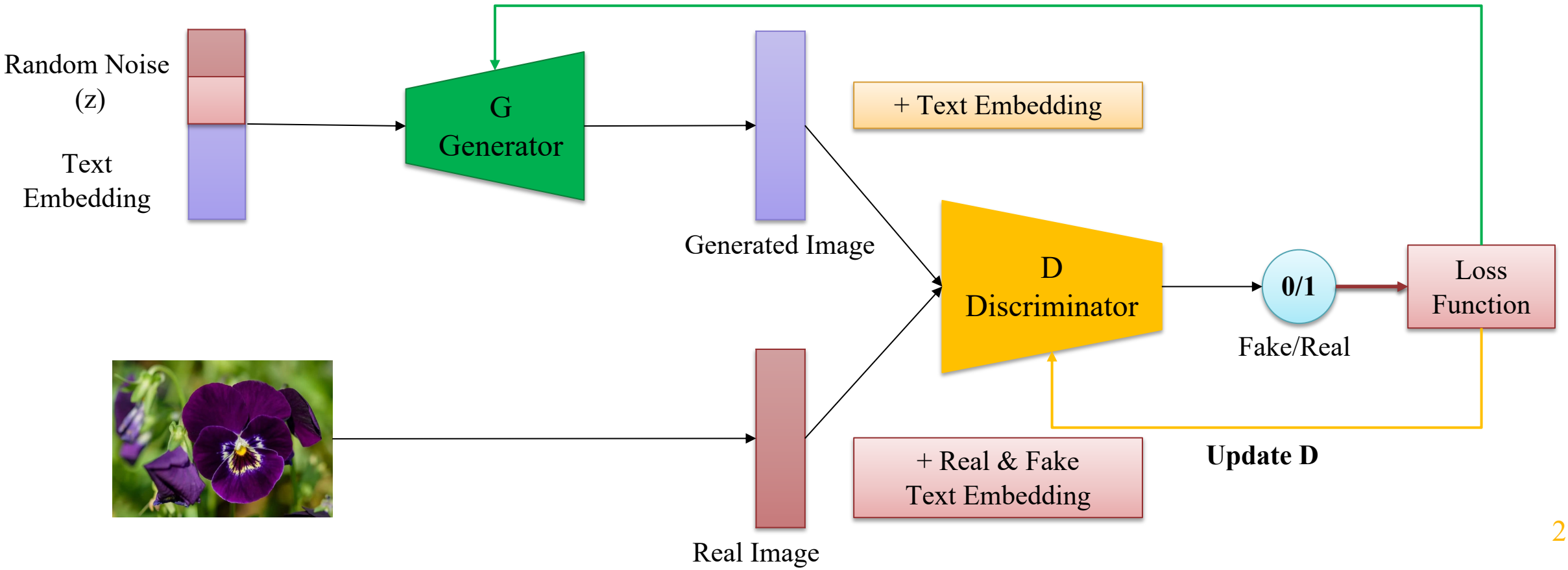
Update D

Text to Image Synthesis

Generator – Pixel-wise Matching Loss

$$L_G(z) = -\log\Big(D\big(G(z,e)\big)\Big) - \alpha L_1(G(z,e),x)$$

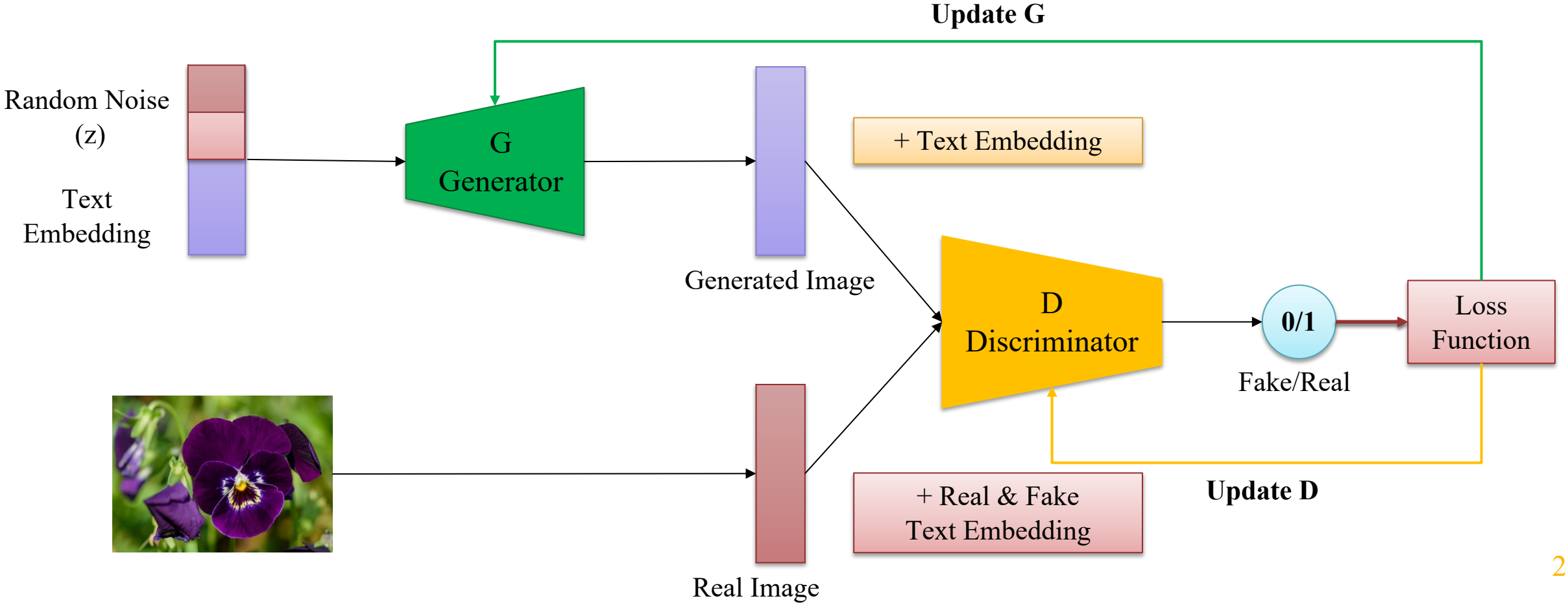
Update G



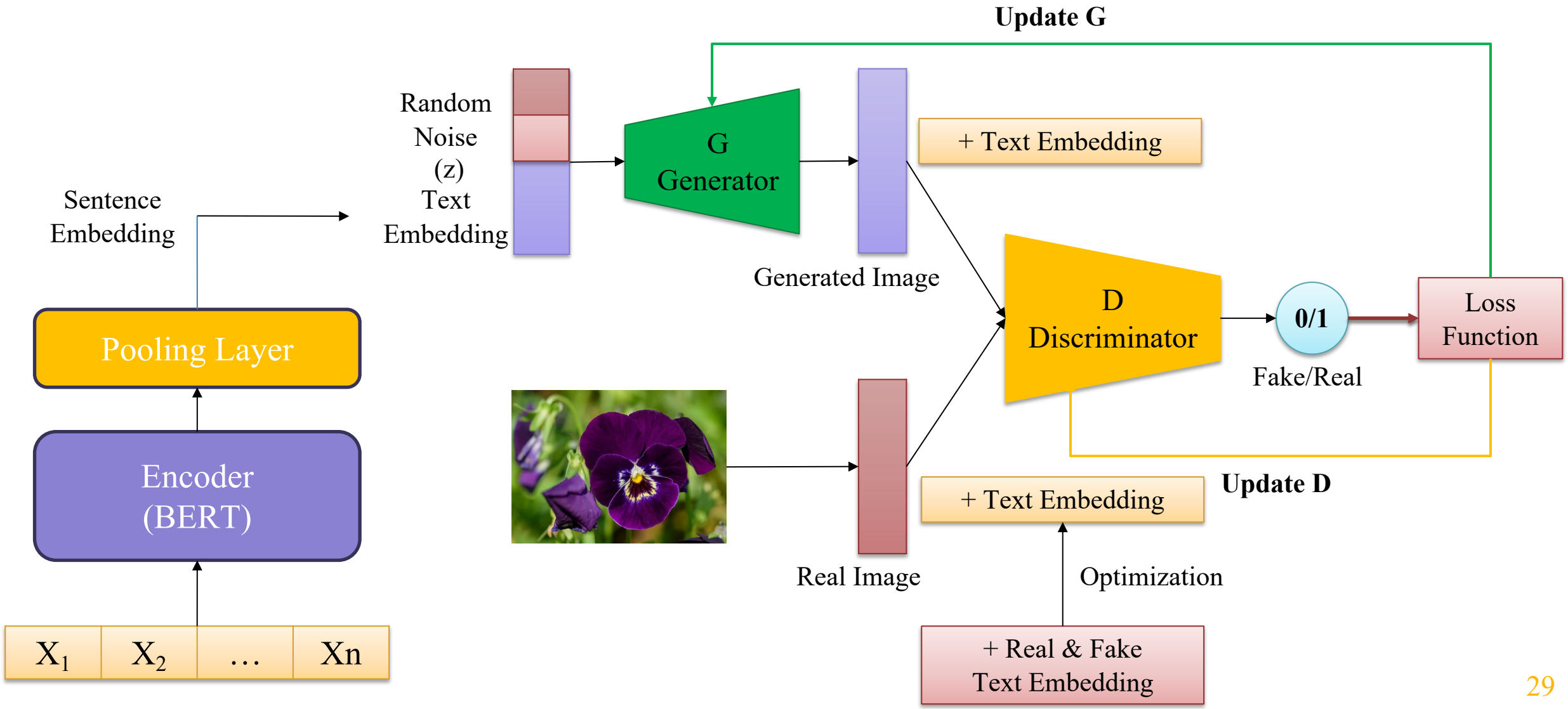
Text to Image Synthesis

Generator – Feature Matching Loss

$$L_G(z) = -\log\Big(D\big(G(z,e)\big)\Big) - \alpha L_1(G(z,e),x) - \beta L_2\Big(D(x), D\big(G(z)\big)\Big)$$



Summary


Update G
Random
Noise
(z)
Text
Embedding
G
Generator
+ Text Embedding
Sentence
Embedding
Generated Image
D
Discriminator
0/1
Loss
Function
Fake/Real
Pooling Layer
Encoder
(BERT)
Update D
+ Text Embedding
Real Image
Optimization
+ Real & Fake
Text Embedding
X1
X2
…
Xn

Thanks!

Any questions?