DL-Seminar
Season#5
AI Lab

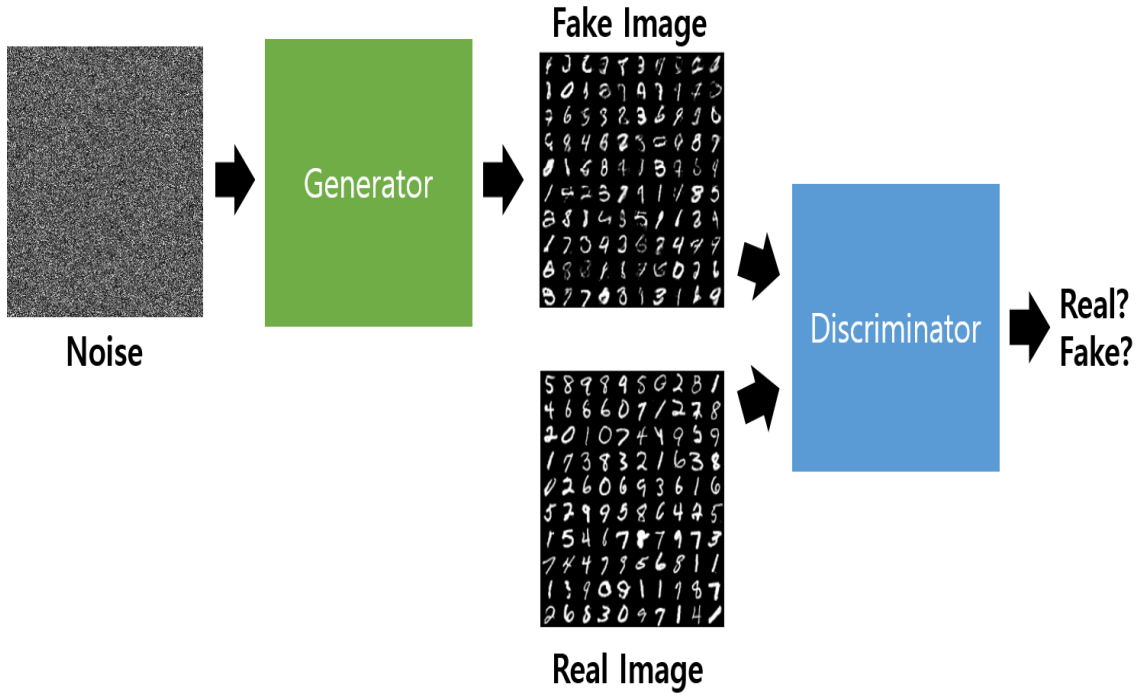
유 재창

StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks

Han Zhang[1], Tao Xu[2], Hongsheng Li[3],
Shaoting Zhang[4], Xiaogang Wang[3], Xiaolei Huang[2], Dimitris Metaxas[1]

[1]Rutgers University [2]Lehigh University [3]The Chinese University of Hong Kong [4]Baidu Research
{han.zhang, dnm}@cs.rutgers.edu, {tax313, xih206}@lehigh.edu
{hsli, xgwang}@ee.cuhk.edu.hk, zhangshaoting@baidu.com

Generalized Adversarial Networks (GAN)



Generator(생성자)
- Random noise z 를 입력으로 받아 Discriminator가 real data와 구분이 어렵도록 학습.

Discriminator(분류기)
- Real data와 fake data를 잘 구분하도록 학습.
- Real -> 1, fake -> 0

Generalized Adversarial Networks (GAN)
Loss Function

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

Problem

Text description

A small bird with varying shades of brown with white under the eyes

A small yellow bird with a black crown and a short black pointed beak

StackGAN

?

Problem

Text description	A small bird with varying shades of brown with white under the eyes	A small yellow bird with a black crown and a short black pointed beak
StackGAN		

Stack Generalized Adversarial Networks (GAN)

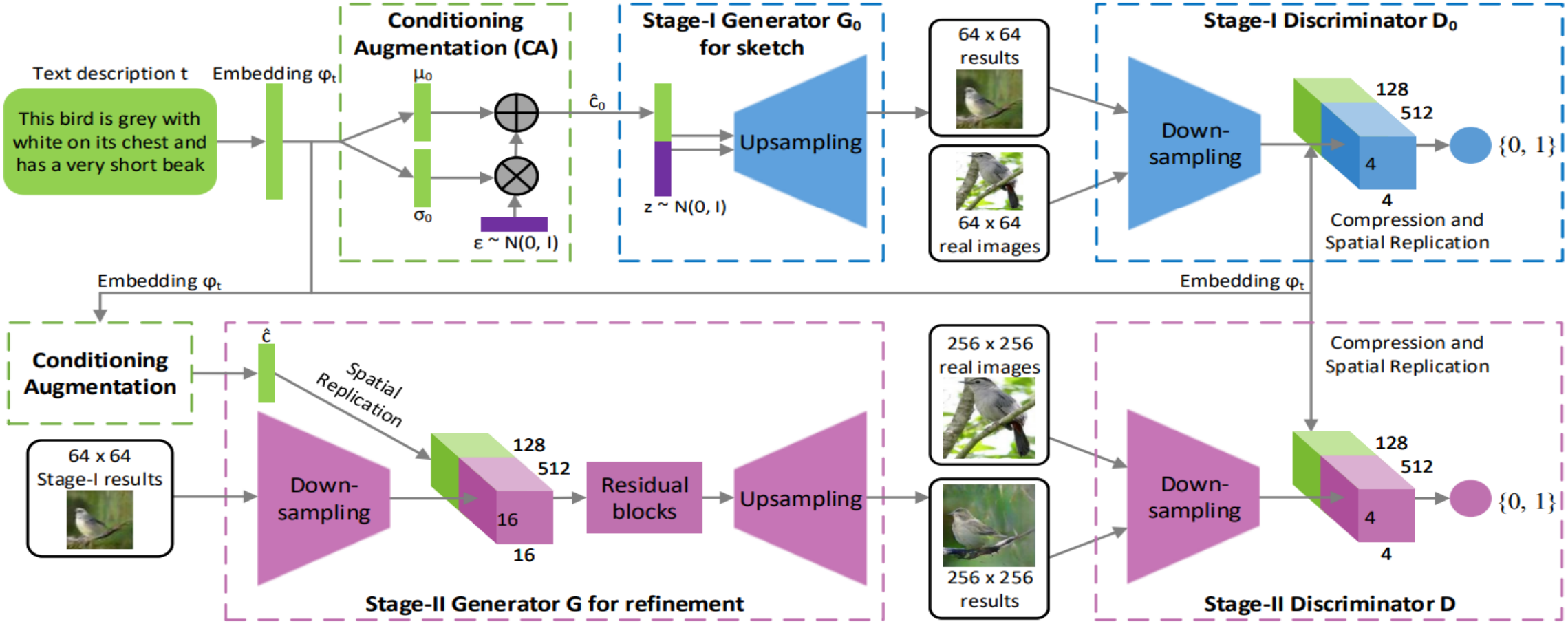
The Stage-I GAN sketches the primitive shape and colors of the object based on the given text description, yielding Stage-I low-resolution images.

Stack Generalized Adversarial Networks (GAN)

The Stage-I GAN sketches the primitive shape and colors of the object based on the given text description, yielding Stage-I low-resolution images.

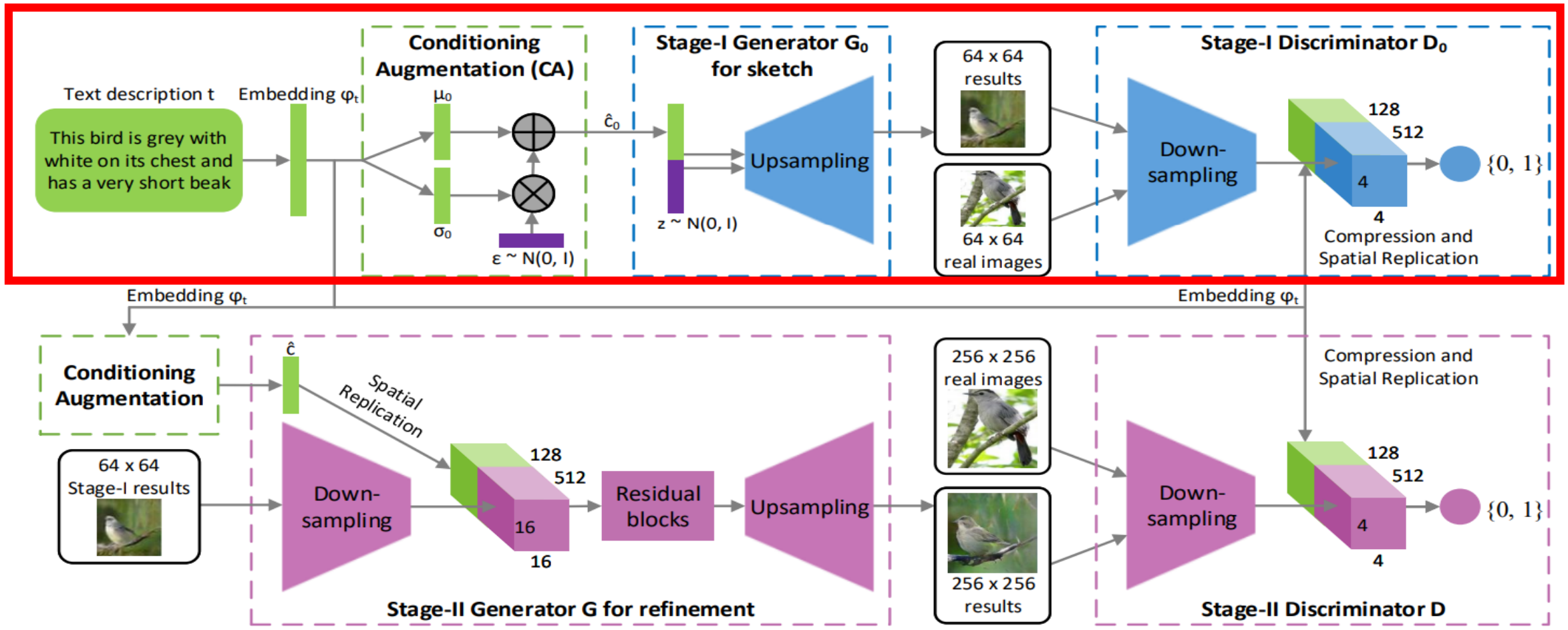
The Stage-II GAN takes Stage-I results and text descriptions as inputs, and generates high-resolution images with photo-realistic details.

Stack GAN Model Architecture

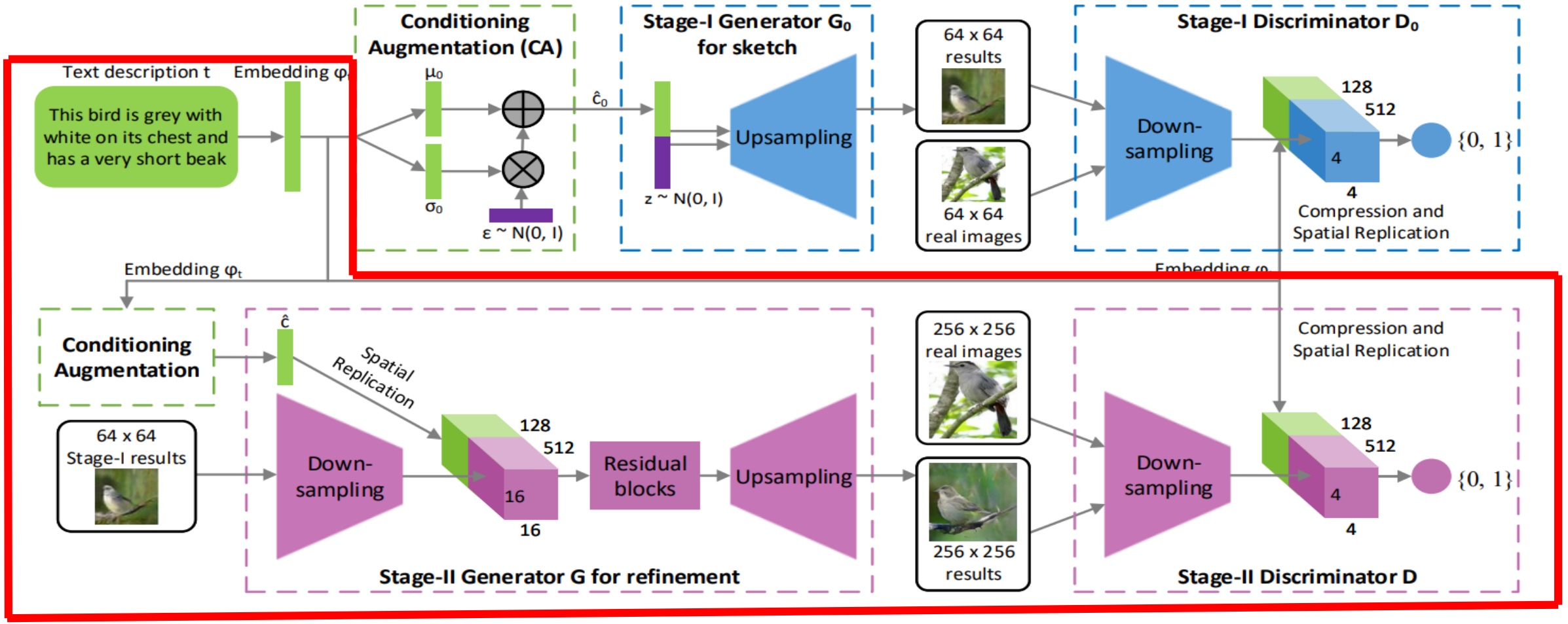


Stack GAN Model Architecture

Stage-1 GAN



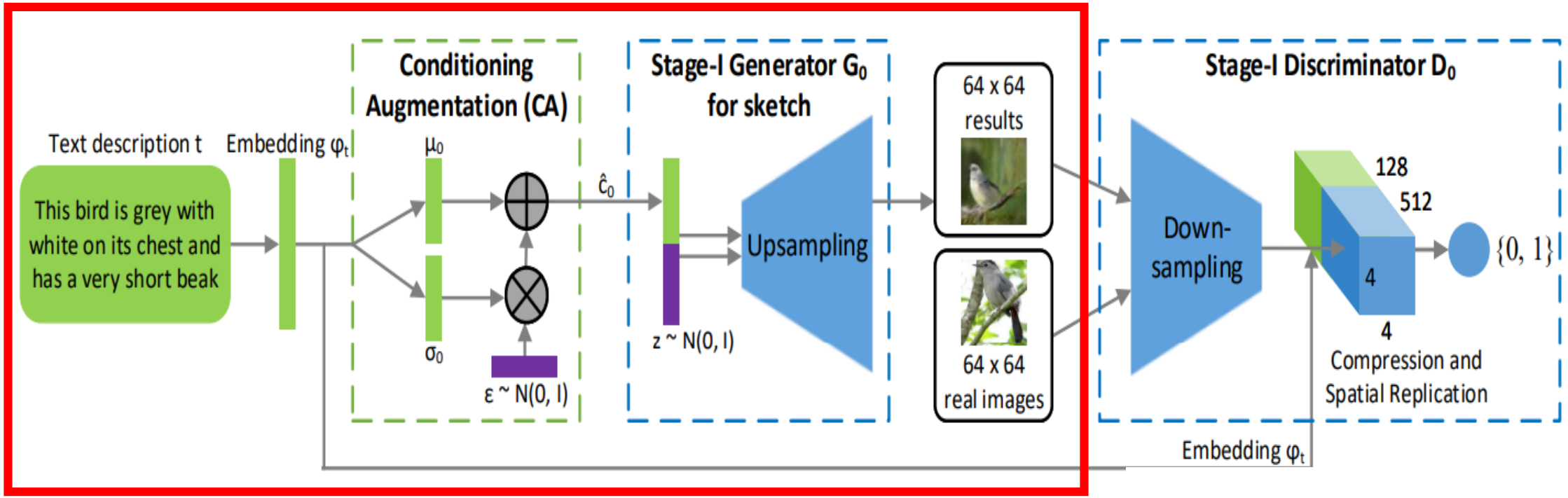
Stack GAN Model Architecture



Stage-2 GAN

Stage-1 GAN / Generator

Pre-trained Encoder -> Embedding vector 생성.
CA를 통해 conditional vector $\hat{c}_0$를 구함.

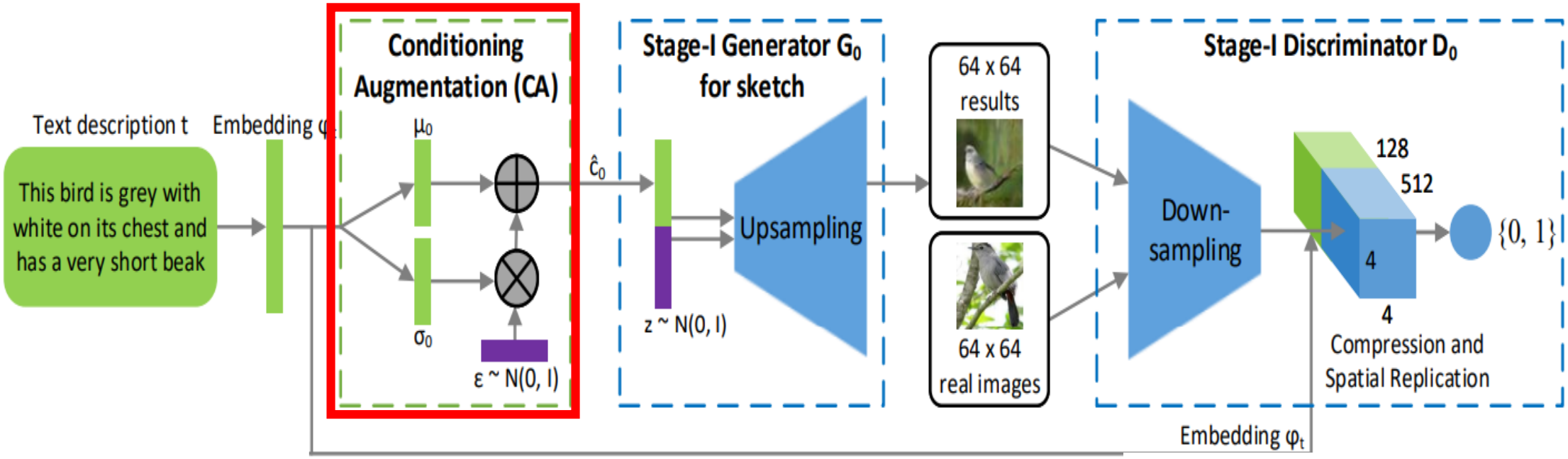


the element-wise multiplication

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \odot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 \cdot b_1 \\ a_2 \cdot b_2 \end{bmatrix}$$

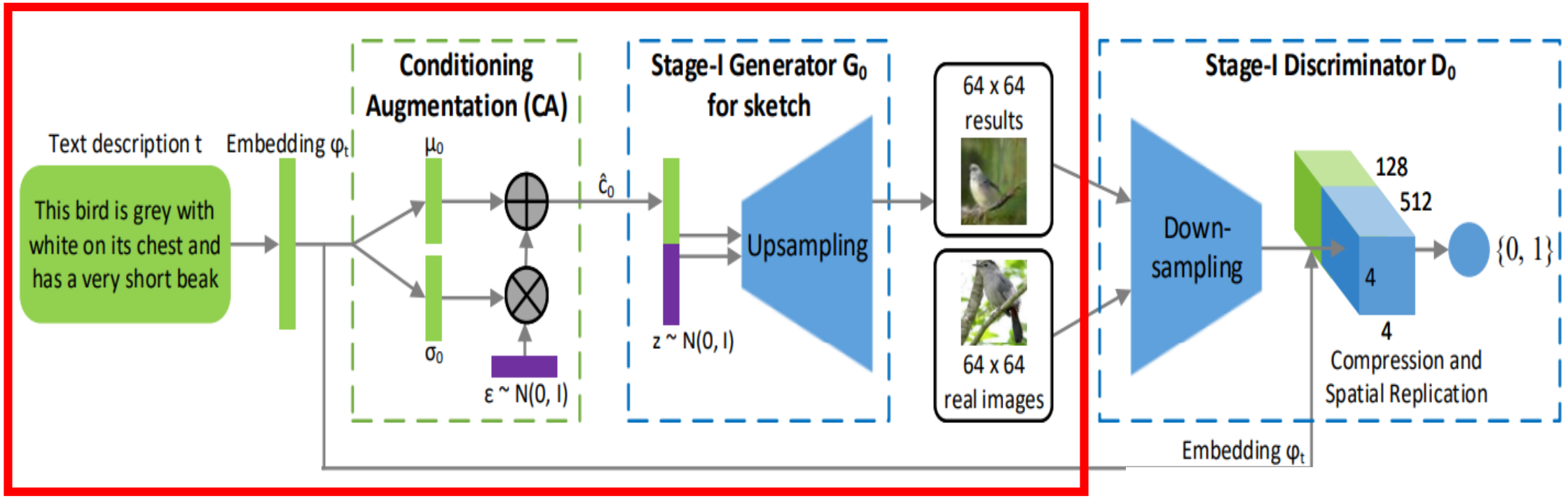
Conditioning Augmentation technique

Conditional GAN -> fixed Condition variable c를 input으로 사용.
Embedding vector -> Fully Connected Layer -> μ_0 , σ_0 (independent Gaussian distribution)
$\hat{c_0} = \mu_0 + \sigma_0 \odot \epsilon$ ($\epsilon \sim \mathcal{N}(0, I)$) -> conditioning vector (not fixed)
적은 text, image pair에도 많은 training pair 효과. -> 학습 good~



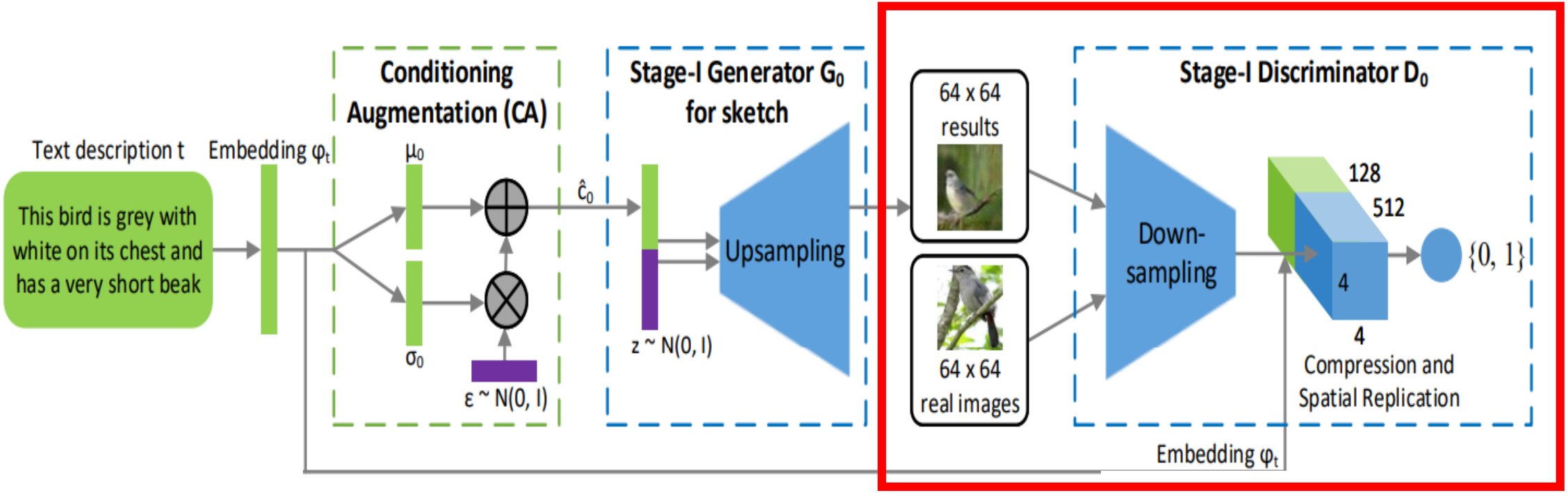
Stage-1 GAN / Generator

Pre-trained Encoder -> Embedding vector 생성.
CA를 통해 conditional vector $\hat{c}_0$를 구함.
z(random noise vector)와 concatenate.
Deconvolution(upsampling) -> 64x64 low image-resolution.



Stage-1 GAN / Discriminator

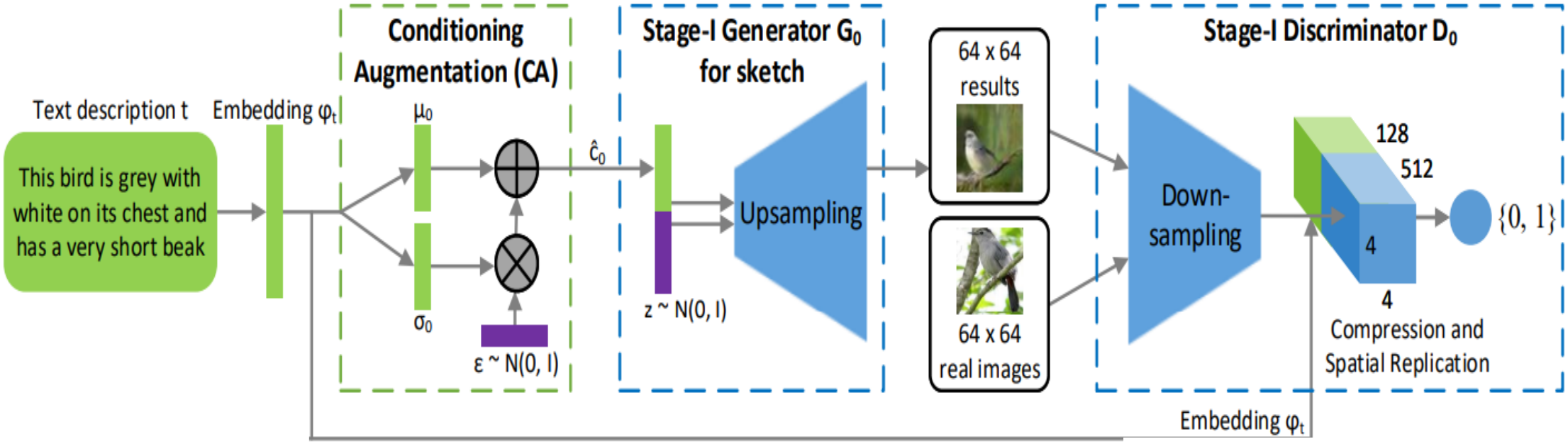
G_0 로부터 생성된 64x64 low image-resolution
Input 으로 들어온 image를 Down-sampling -> 4x4x512 image map 생성
Embedding vector(128 dimension)를 Replication -> 4x4x128의 tensor 생성
1x1 convolutional layer -> image, text feature 관계 학습.
FCL with 1 node



Stage-1 GAN Loss Function

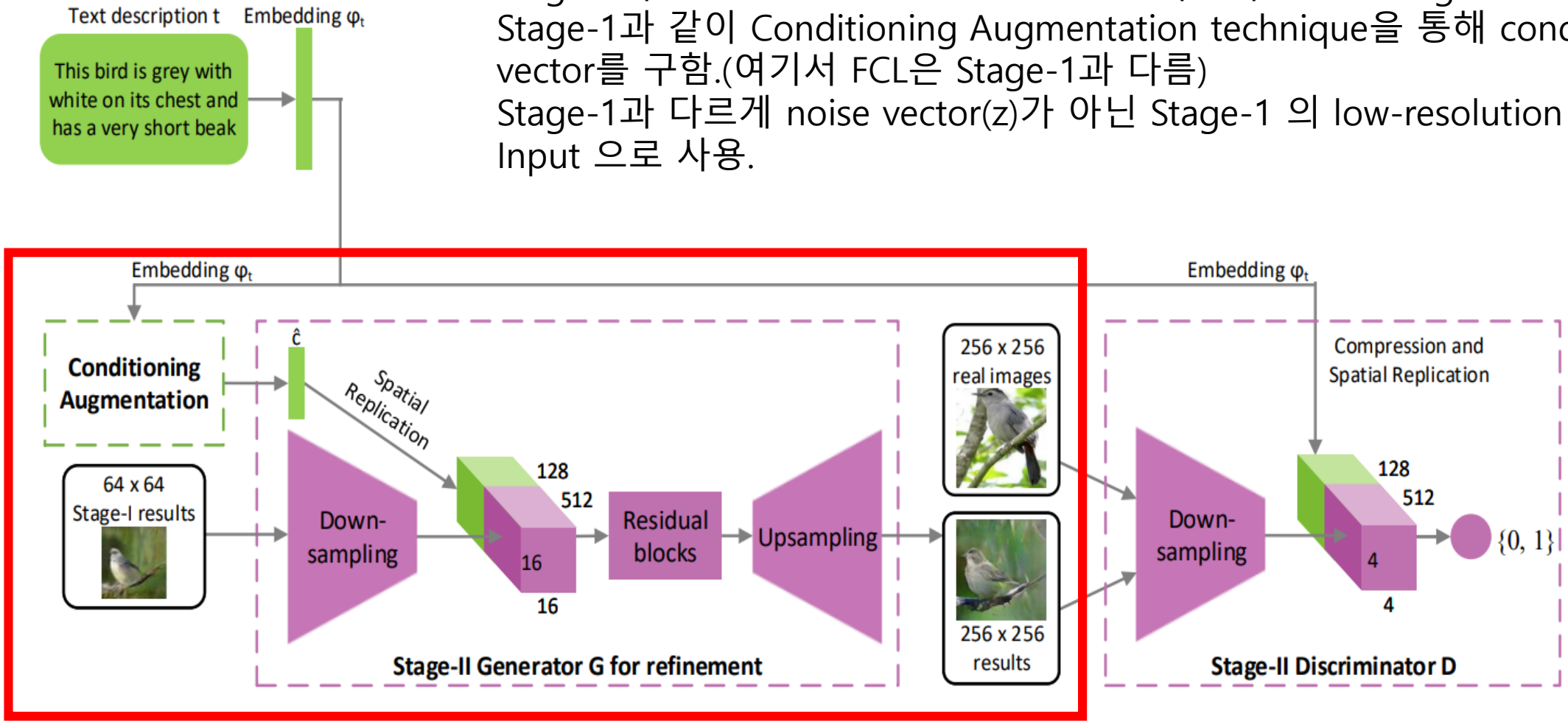
$$\mathcal{L}_{D_0} = \mathbb{E}_{(I_0,t)\sim p_{data}}[\log D_0(I_0, \varphi_t)] + \mathbb{E}_{z\sim p_z, t\sim p_{data}}[\log(1 - D_0(G_0(z, \hat{c}_0), \varphi_t))],$$

$$\mathcal{L}_{G_0} = \mathbb{E}_{z\sim p_z, t\sim p_{data}}[\log(1 - D_0(G_0(z, \hat{c}_0), \varphi_t))] + \lambda D_{KL}(\mathcal{N}(\mu_0(\varphi_t), \Sigma_0(\varphi_t)) \,||\, \mathcal{N}(0, I)),$$

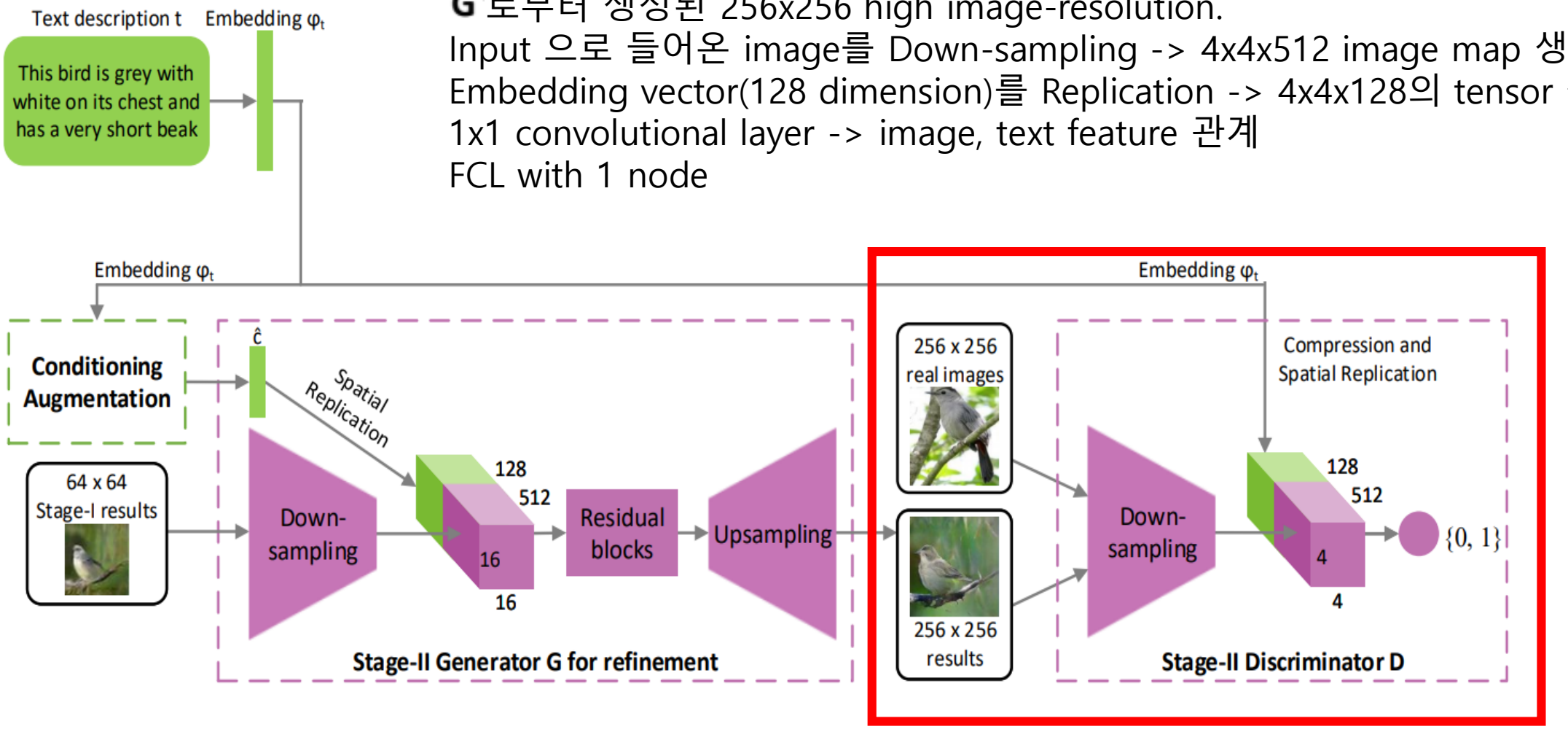


Stage-2 GAN / Generator

Stage-1과 동일한 Pre-trained Encoder를 이용해 Embedding vector 생성.
Stage-1과 같이 Conditioning Augmentation technique을 통해 conditioning vector를 구함.(여기서 FCL은 Stage-1과 다름)
Stage-1과 다르게 noise vector(z)가 아닌 Stage-1 의 low-resolution image를 Input 으로 사용.



Stage-2 GAN / Discriminator



G 로부터 생성된 256x256 high image-resolution.
Input 으로 들어온 image를 Down-sampling -> 4x4x512 image map 생성
Embedding vector(128 dimension)를 Replication -> 4x4x128의 tensor 생성
1x1 convolutional layer -> image, text feature 관계
FCL with 1 node

Stage-2 GAN Loss Function

$$\mathcal{L}_D = \mathbb{E}_{(I,t)\sim p_{data}}[\log D(I, \varphi_t)] + \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{data}}[\log(1 - D(G(s_0, \hat{c}), \varphi_t))],$$

$$\mathcal{L}_G = \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{data}}[\log(1 - D(G(s_0, \hat{c}), \varphi_t))] + \lambda D_{KL}(\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t)) \,||\, \mathcal{N}(0, I)),$$

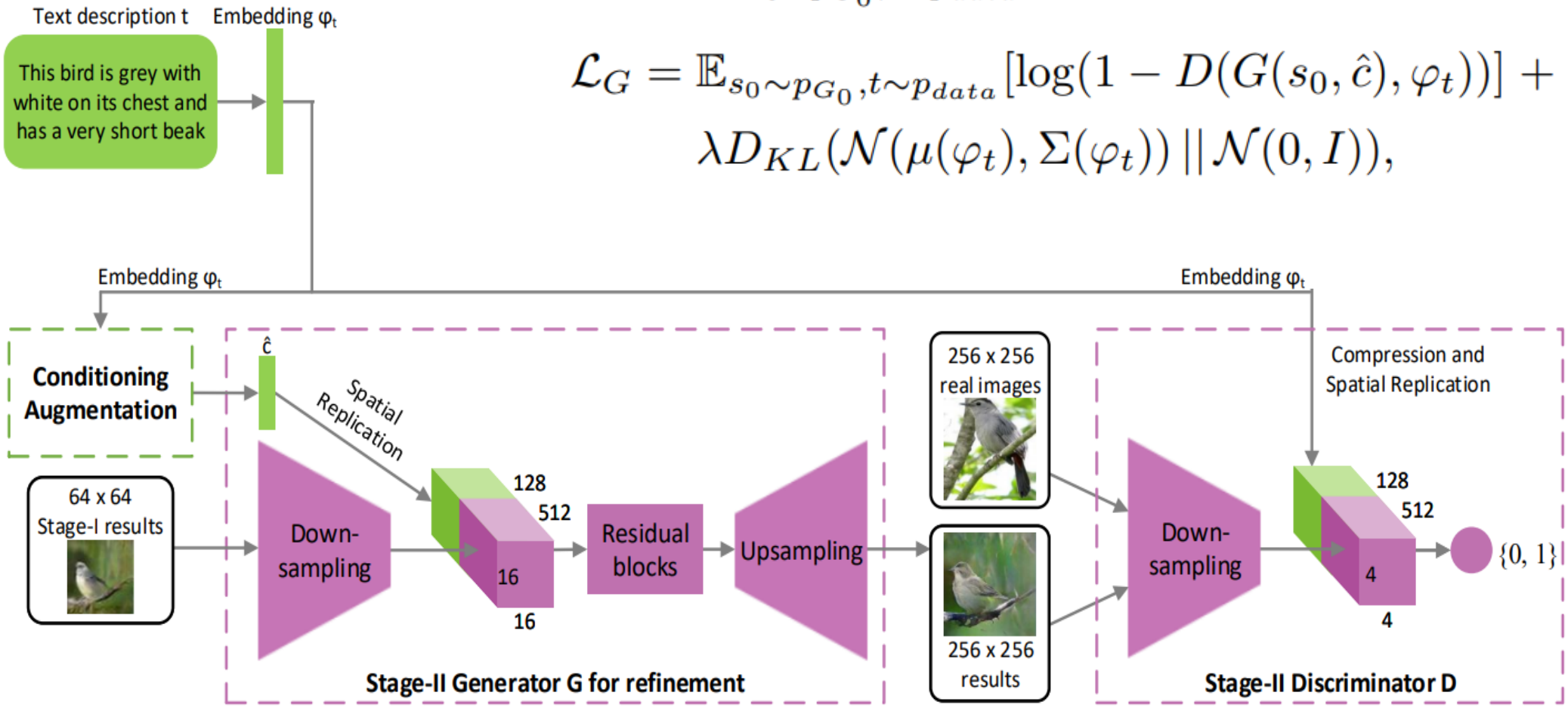
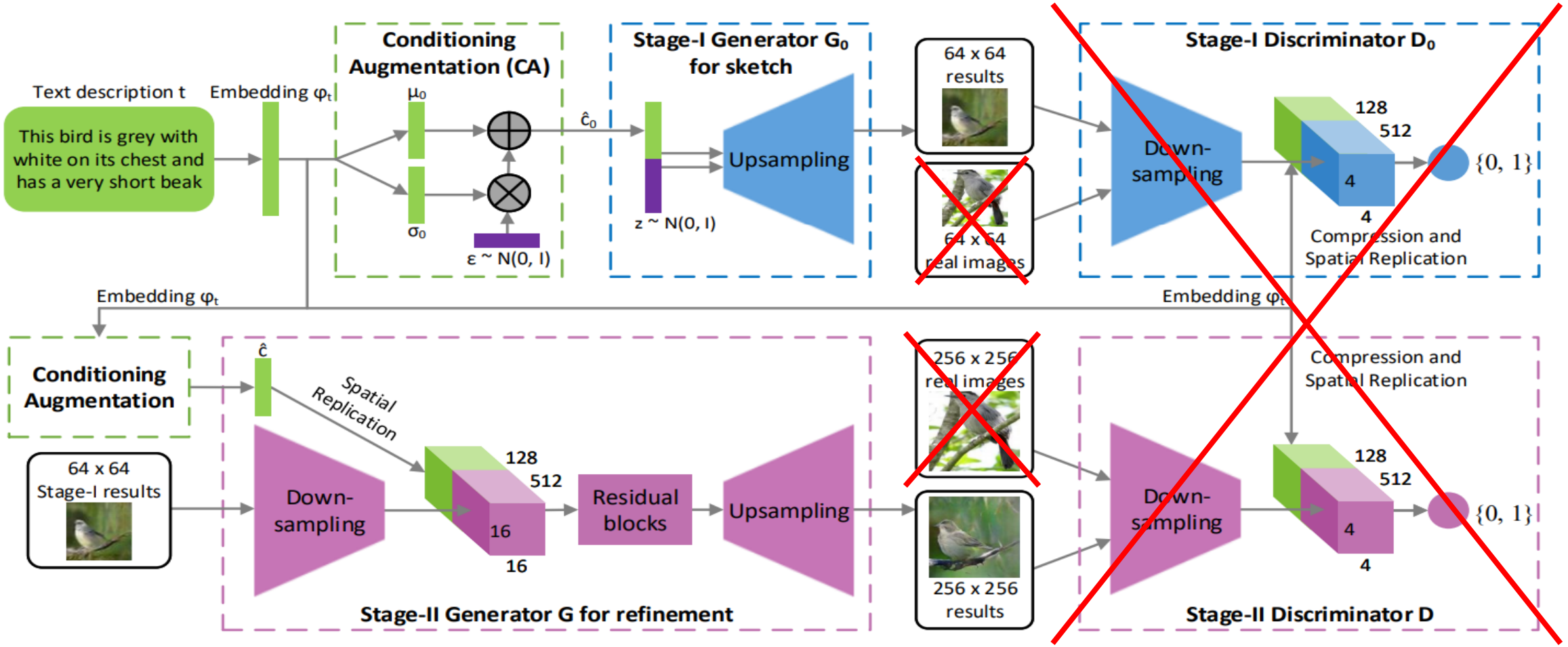


Photo-realistic image generation



Experiments and Comparison

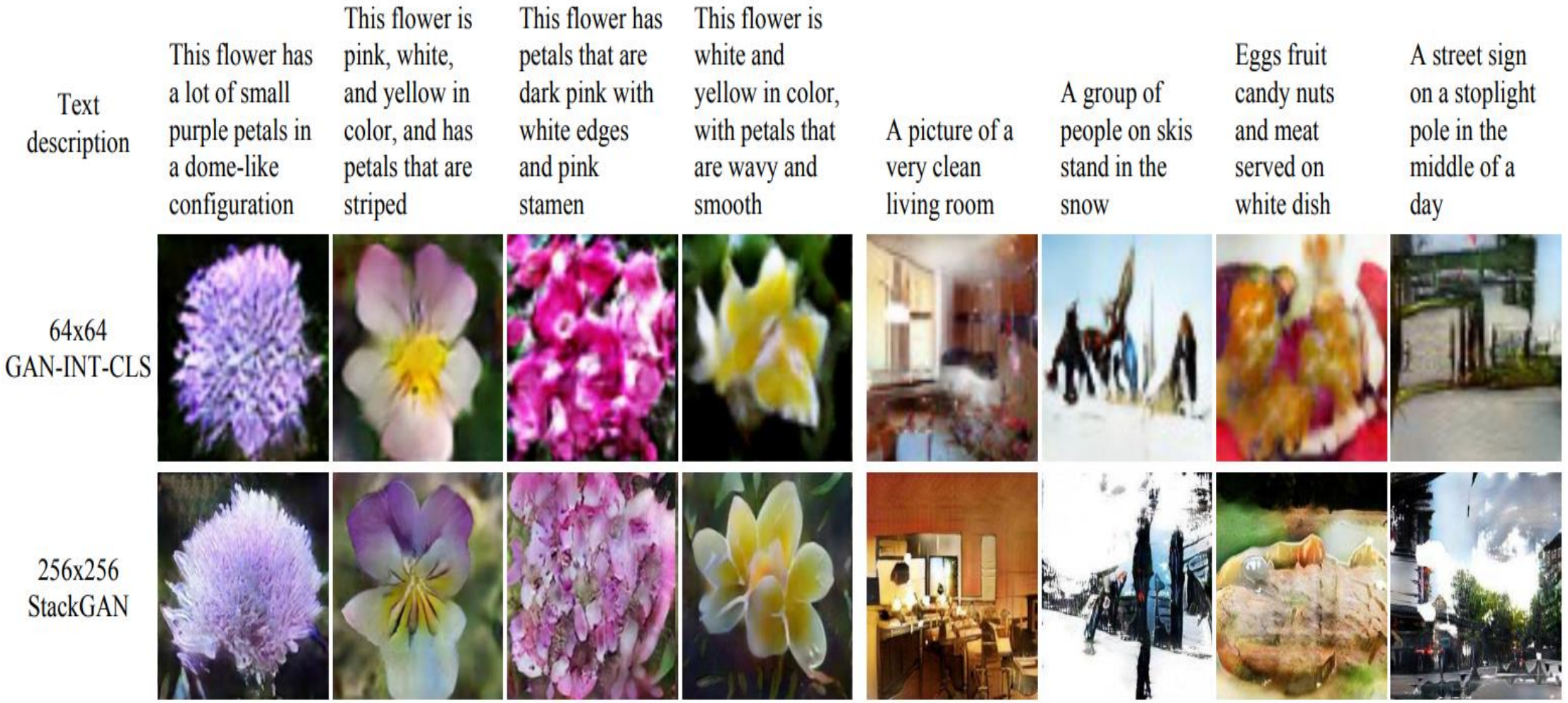
Experiments and Comparison

Experiments and Comparison

Experiments and Comparison

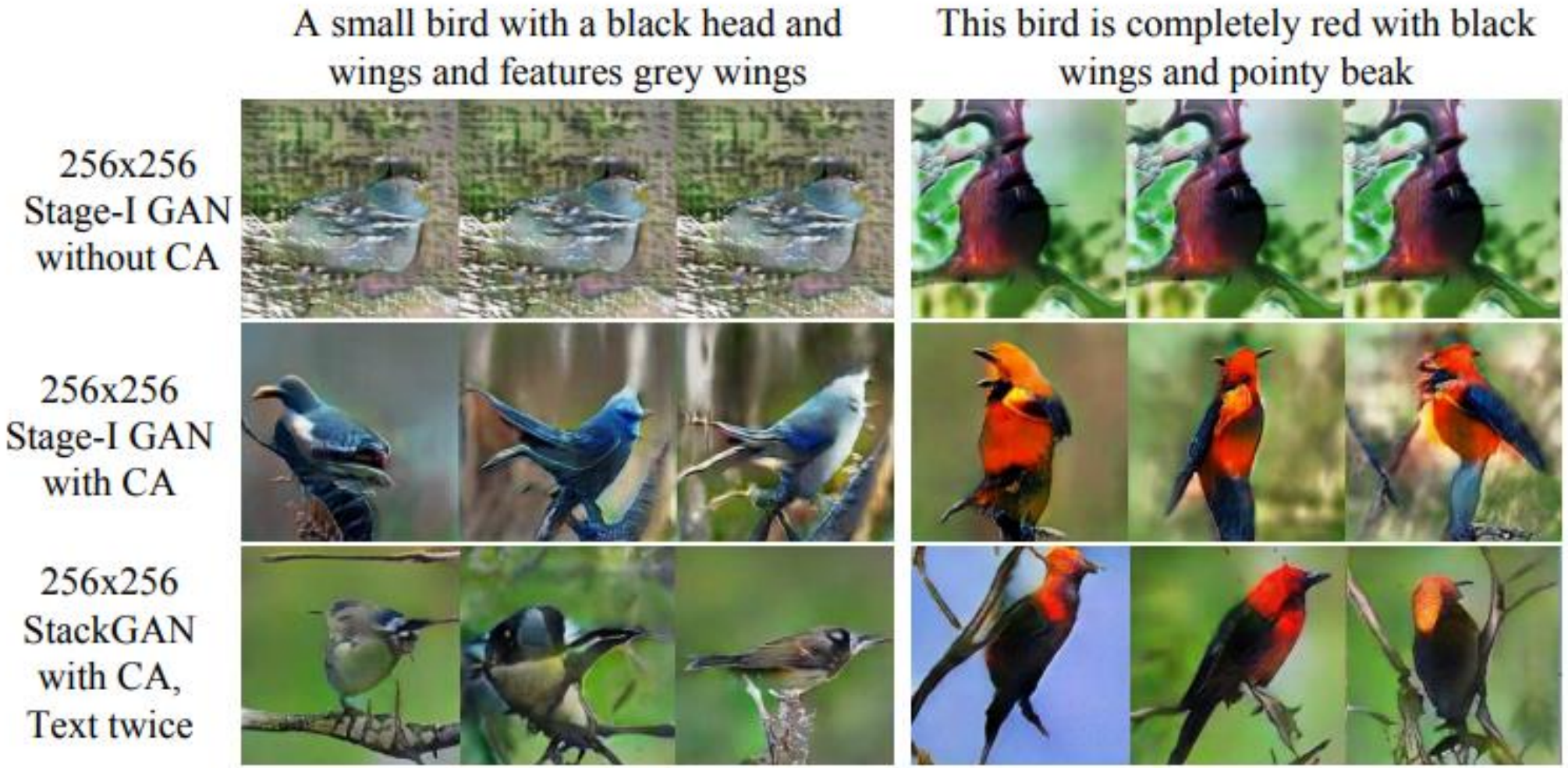
Experiments and Comparison

Experiments and Comparison



Experiments and Comparison

Experiments and Comparison



끝