Herbert Wertheim College of Engineering
UNIVERSITY of FLORIDA

POWERING THE NEW ENGINEER TO TRANSFORM THE FUTURE

Large-scale Intelligent Systems Laboratory
NSF I/UCRC Center for Big Learning
Department of Electrical and Computer Engineering
Department of Computer & Information Science & Engineering

StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks

Xiyao Ma

https://arxiv.org/pdf/1710.10916.pdf

StackGAN++

- GAN made some huge success in various tasks, but not good on high quality image generation.
- StackGAN++ consists of multiple generators and discriminators in a tree-like structure; images at multiple scales corresponding to the same scene are generated from different branches of the tree.

Generative Adversarial Network

- GAN comprises a Generative model G and a Discriminator D that they are trained alternatively to compete with each other.
  - The generator G is optimized to reproduce the true data distribution pdata by generating images that are difficult for the discriminator D to differentiate from real images.
  - D is optimized to distinguish real images and synthetic images generated by G

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

where x is a real image from the true data distribution pdata, and z is a noise vector sampled from distribution pz (e.g., uniform or Gaussian distribution).

Conditional GAN is an extension of GAN where both the generator and discriminator receive additional conditioning variables c, yielding G(z,c) and D(x,c).

Task

- Description: Generate images with text t.
- Dataset: CUB, Oxford-102 and MS COCO.
- Requirement: Generated images should be real and match with given context.

StackGAN-v2

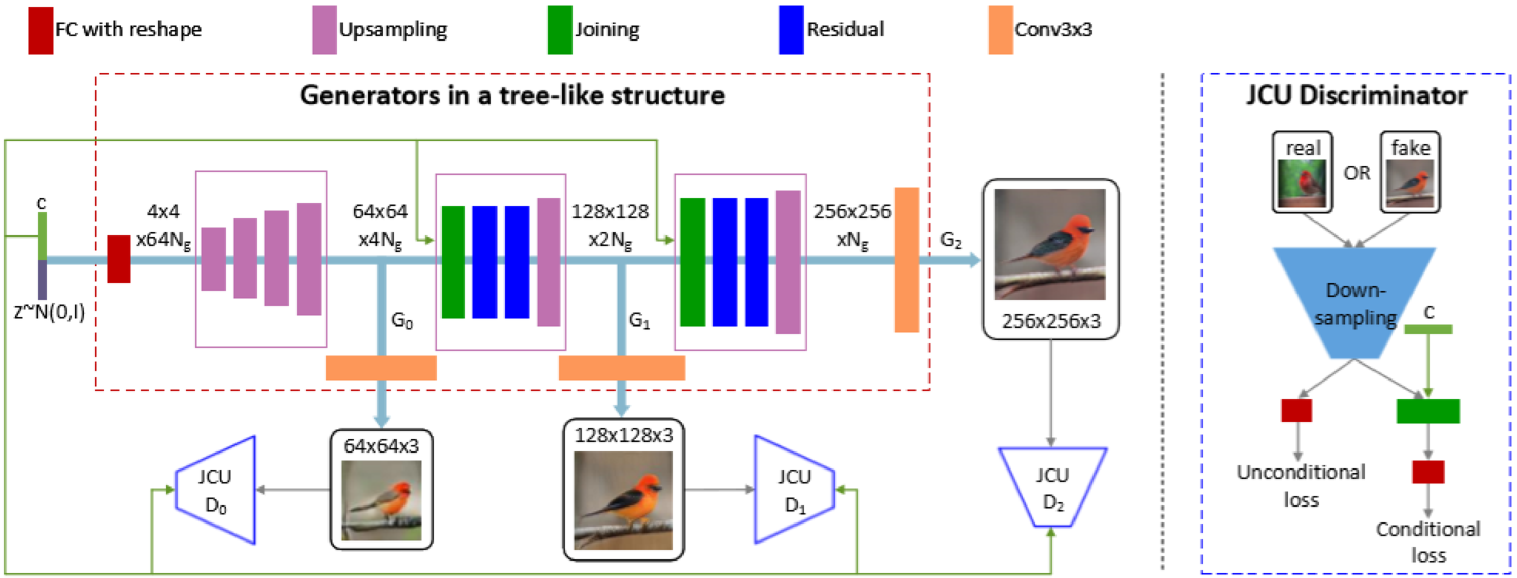

Fig. 2: The overall framework of our proposed StackGAN-v2 for the conditional image synthesis task. c is the vector of conditioning variables which can be computed from the class label, the text description, *etc.*. N_g and N_d are the numbers of channels of a tensor.

Loss Function

- Joint conditional and unconditional Discriminator
  - The unconditional loss determines whether the image is real or fake
  - The conditional one determines whether the image and the condition match or not.

i_{th} Discriminator:

$$\mathcal{L}_{D_i} = \underbrace{-\frac{1}{2}\mathbb{E}_{x_i \sim p_{data_i}}\left[\log D_i(x_i)\right] - \frac{1}{2}\mathbb{E}_{s_i \sim p_{G_i}}\left[\log(1 - D_i(s_i)\right]}_{\text{unconditional loss}} + \underbrace{-\frac{1}{2}\mathbb{E}_{x_i \sim p_{data_i}}\left[\log D_i(x_i, c)\right] - \frac{1}{2}\mathbb{E}_{s_i \sim p_{G_i}}\left[\log(1 - D_i(s_i, c)\right]}_{\text{conditional loss}}.$$

i_{th} Generator:

$$\mathcal{L}_{G_i} = \underbrace{\frac{1}{2}\mathbb{E}_{s_i \sim p_{G_i}}\left[\log(1 - D_i(s_i)\right]}_{\text{unconditional loss}} + \underbrace{\frac{1}{2}\mathbb{E}_{s_i \sim p_{G_i}}\left[\log(1 - D_i(s_i, c)\right]}_{\text{conditional loss}}.$$

Color-consistency regularization

- Motivation: Generated images at different generators should share similar basic structure and colors.
- Let x_k = (R,G,B)T represent a pixel in a generated image, then the mean and covariance of pixels of the given image can be defined by $\boldsymbol{\mu} = \sum_k \boldsymbol{x_k}/N$ and $\boldsymbol{\Sigma} = \sum_k (\boldsymbol{x_k} - \boldsymbol{\mu})(\boldsymbol{x_k} - \boldsymbol{\mu})^T/N$, where N is the number of pixels in the image. The color-consistency regularization term aims at minimizing the differences of µ and Σ between different scales to encourage the consistency, which is defined as

$$\mathcal{L}_{C_i} = \frac{1}{n}\sum_{j=1}^{n}\left(\lambda_1 \|\boldsymbol{\mu}_{s_i^j} - \boldsymbol{\mu}_{s_{i-1}^j}\|_2^2 + \lambda_2 \|\boldsymbol{\Sigma}_{s_i^j} - \boldsymbol{\Sigma}_{s_{i-1}^j}\|_F^2\right)$$

where n is the batch size, µsji and Σsji are mean and covariance for the jth sample generated by the ith generator.

Loss function of the ith generator: $\mathcal{L}'_{G_i} = \mathcal{L}_{G_i} + \alpha * \mathcal{L}_{C_i}$

Performance

Metric	Dataset	GAN-INT-CLS	GAWWN	Our StackGAN-v1	Our StackGAN-v2
Inception score	CUB	2.88 ± .04	3.62 ± .07	3.70 ± .04	**4.04 ± .05**
	Oxford	2.66 ± .03	/	**3.20 ± .01**	/
	COCO	7.88 ± .07	/	**8.45 ± .03**	/
Human rank	CUB	2.81 ± .03	1.99 ± .04	**1.37 ± .02**	/
	Oxford	1.87 ± .03	/	**1.13 ± .03**	/
	COCO	1.89 ± .04	/	**1.11 ± .03**	/

TABLE 2: Inception scores and average human ranks of our StackGAN-v1, StackGAN-v2, GAWWN [29], and GAN-INT-CLS [31] on CUB, Oxford-102, and MS-COCO datasets.

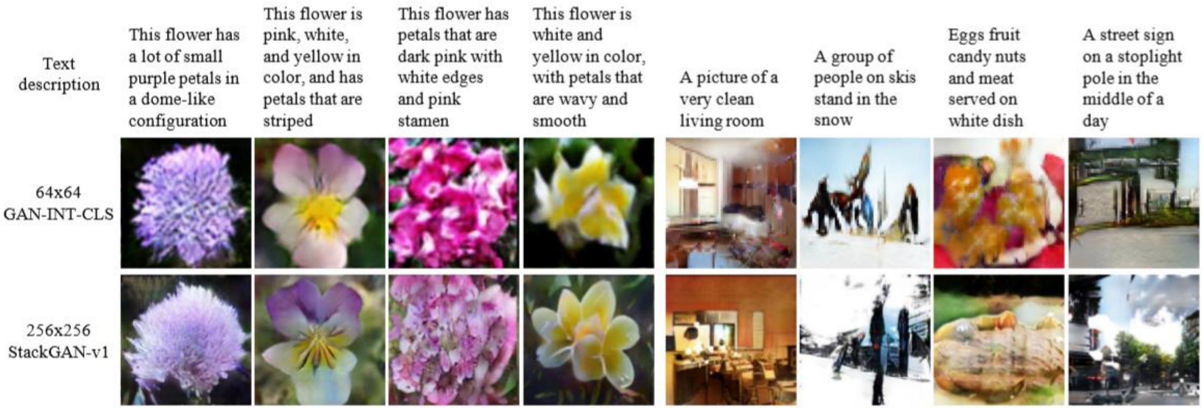

Fig. 4: Example results by our StackGAN-v1 and GAN-INT-CLS [31] conditioned on text descriptions from Oxford-102 test set (leftmost four columns) and COCO validation set (rightmost four columns).

Performance

Model	branch G_1	branch G_2	branch G_3	JCU	inception score
StackGAN-v2	64×64	128×128	256×256	yes	4.04 ± .05
StackGAN-v2-no-JCU	64×64	128×128	256×256	no	3.77 ± .04
StackGAN-v2-G_3	removed	removed	256×256	yes	3.49 ± .04
StackGAN-v2-3G_3	removed	removed	three 256×256	yes	3.22 ± .02
StackGAN-v2-all256	256×256	256×256	256×256	yes	2.89 ± .02

TABLE 4: Inception scores by our StackGAN-v2 and its baseline models on CUB test set. "JCU" means using the proposed discriminator that jointly approximates conditional and unconditional distributions.

(a) StackGAN-v2-all256 (b) StackGAN-v2-G_3 (c) StackGAN-v2-3G_3 (d) StackGAN-v2

This black and white and grey bird has a black bandit marking around it's eyes

(e) StackGAN-v2-all256 (f) StackGAN-v2-G_3 (g) StackGAN-v2-no-JCU (h) StackGAN-v2

Fig. 13: Example images generated by the StackGAN-v2 and its baseline models on LSUN bedroom (top) and CUB (bottom) datasets.

Thank You!

NSF

NIH
National Institutes of Health

U.S. Department of Homeland Security

Department of Energy · United States of America