Natural Language Processing
& Artificial Intelligence

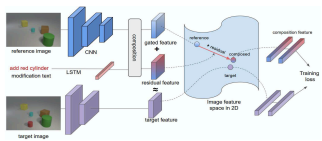
Text to Image Synthesis (T2I)
Text guided Image Manipulation (TGIM)

고려대학교 컴퓨터학과 석사과정
2019011001 박성진

Index

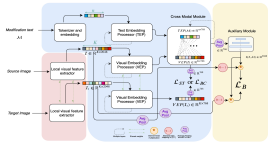
Composing Text and Image for Image Retrieval~
~> Text-guided image manipulation

• Composing Text and Image for Image Retrieval



TIRG; Text Image Residual Gating
(CVPR 2019)

- BASELINE
- No attention

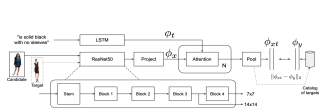


Locally Bounded
(CVPR 2020)

- Self-Attention, Cross-Attention
- Auxiliary Module

VAL;Visualinguistic Attention Learning
(CVPR 2020)

- Self-Attention, Cross-Attention

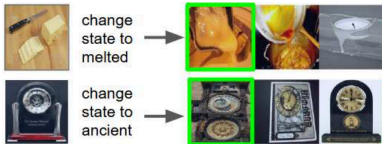


MAAF;Modality-Agnostic Attention Fusion
(Arxiv 2020)

- Self-Attention, Cross-Attention

- Fashion200k
  - train 172k, test 30k, val 3k
  - attribute-like (one word difference)
  - ”replace with ---- “

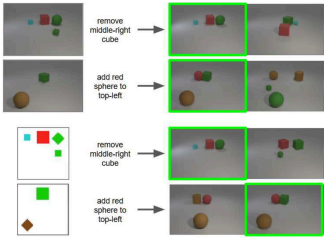
- MIT-state(245noun/115adj)
  - 60k images with object-state label (e.g., “red tomato”, “new camera”)
  - fix noun, change adj



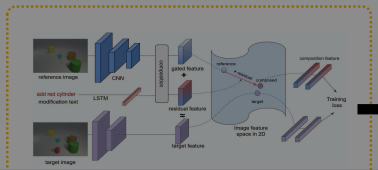
- Fashion IQ
  - Workshop
  - 40k images with modification text

- Shoes (train 10k test 4.7k)
  - more enrich text
  - relative caption/ description

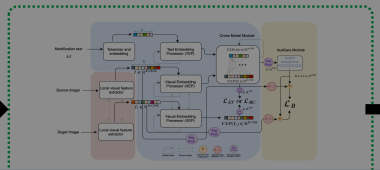
- CSS(color, shape, size)
  - train 46K test 15K
  - add, remove, change object attribute
  - 2d->3D / 3D -> 3D
  - more enrich text



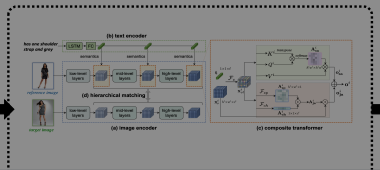
- Composing Text and Image for Image Retrieval



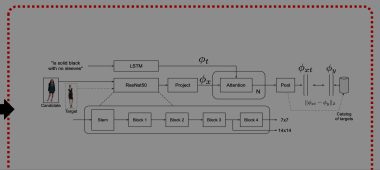
TIRG; Text Image Residual Gating
(CVPR 2019)



Locally Bounded
(CVPR 2020)



VAL;Visualinguistic Attention Learning
(CVPR 2020)



MAAF;Modality-Agnostic Attention Fusion
(Arxiv 2020)

1. Visual-Linguistic Representation
2. How to retrieval

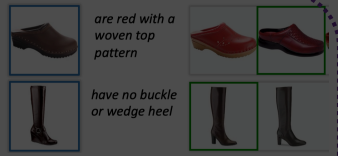
• BASELINE
• No attention

• Auxiliary Module

• Self-Attention, Cross-Attention

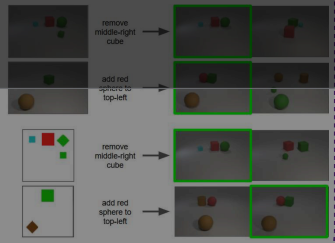
• Fashion200k
- train 172k, test 30k, val 3k
- attribute-like (one word difference)
- "replace with ---- "

• Shoes (train 10k test 4.7k)
- more enrich text
- relative caption/ description



• MIT-state(245noun/115adj)
- 60k images with object-state label
(e.g., "red tomato", "new camera")
- fix noun, change adj

• CSS(color, shape, size)
- train 46K test 15K
- add, remove, change object attribute
- 2d->3D / 3D -> 3D
- more enrich text

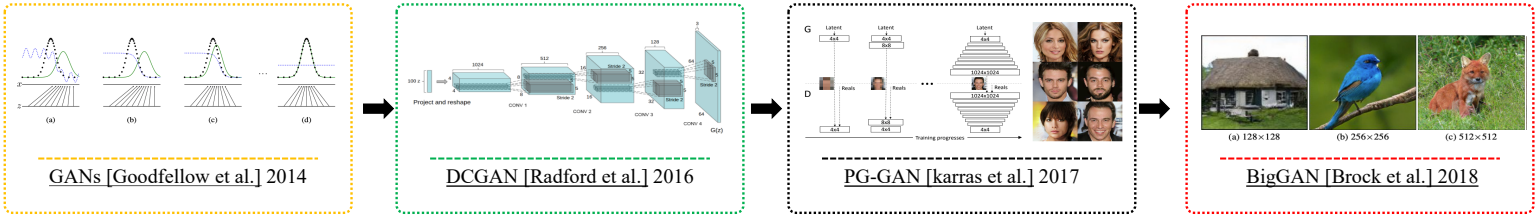


• Fashion IQ
- Workshop
- 40k images with modification text

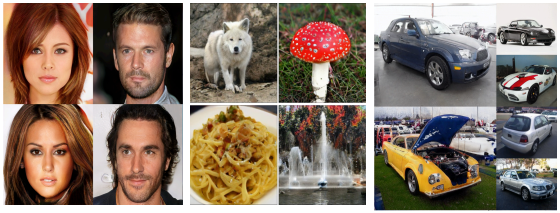
- Composing Text and Image for Image Retrieval

➡ What about Generative models?

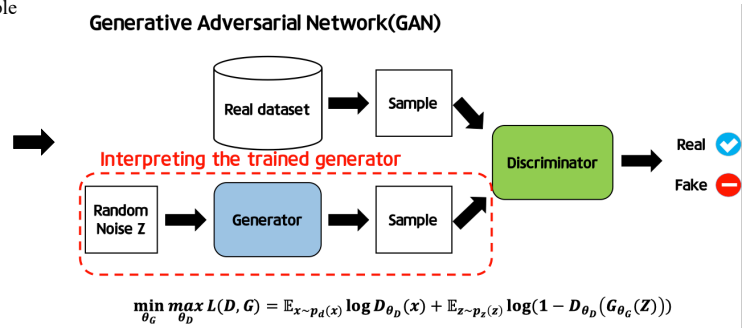
GANs for Synthesizing Images Since 2014…



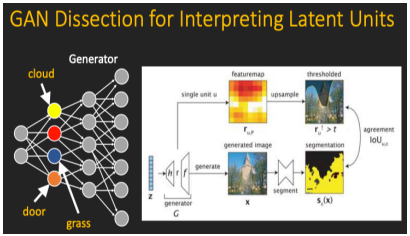
- Toward more high-resolution , super-resolution images with **high variation** such as age, gender, color, hair (in Face datasets)
- What is GAN learned? -> If we know that, Image Manipulation is possible

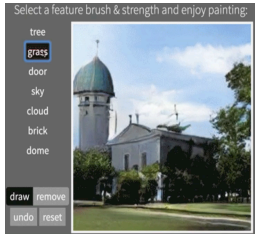


PG-GAN BigGAN StyleGAN

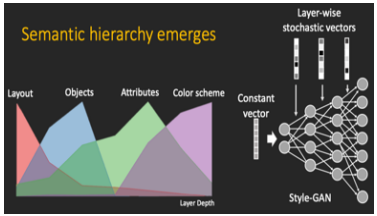


- Interpreting the trained Generator; latent unit, latent space

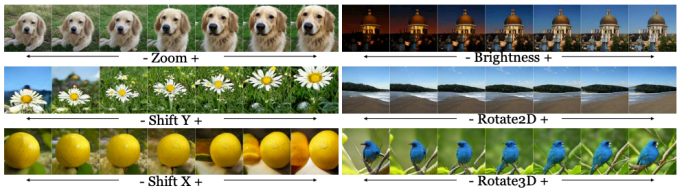




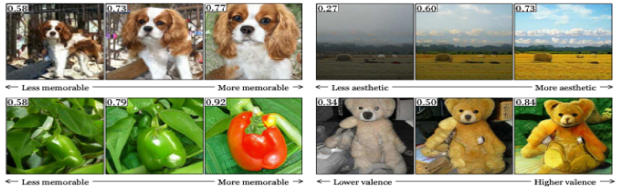
GAN Dissection for Interpreting Latent Units (ICLR 2019)



HiGAN; Semantic Hierarchy Emerges in Deep Generative Representations for Scene Synthesis (Arxiv 2020)



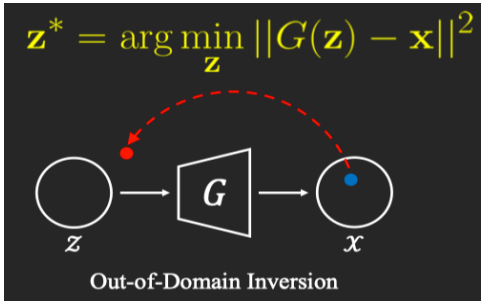
On the "steerability" of generative adversarial networks (Arxiv 2020)

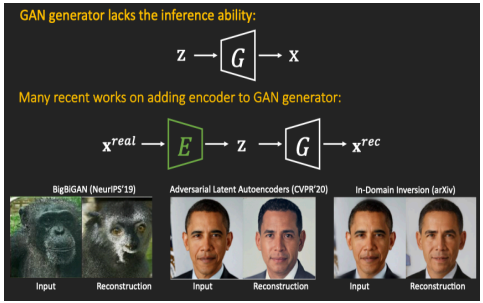


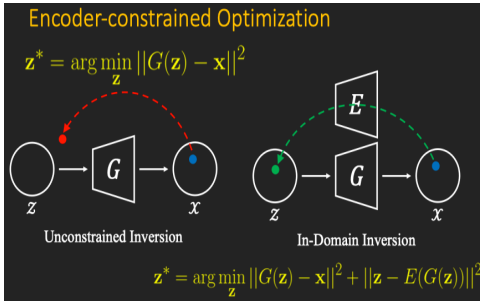
GANalyze: Toward Visual Definitions of Cognitive Image Properties (ICCV 2019)

➡ Manipulate my own image not from Z; GAN Inversion

- GAN Inversion (not from z but my own image)







Algorithm 1: Latent Space Embedding for GANs

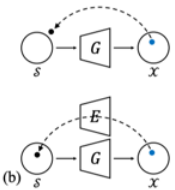
Input: An image $I \in \mathbb{R}^{n \times m \times 3}$ to embed; a pre-trained generator $G(\cdot)$.

Output: The embedded latent code w^* and the embedded image $G(w^*)$ optimzed via F'.

1 Initialize latent code $w^* = w$;
2 **while** *not converged* **do**
3 $\quad L \leftarrow L_{percept}(G(w^*), I) + \frac{1}{n}\|G(w^*) - I\|_2^2$;
4 $\quad w^* \leftarrow w^* - \eta F'(\nabla_{w^*} L)$;
5 **end**

- optimization-based approach
- StyleGAN (FFHQ)
- Reconstruction OK
- Manipulation -> only Face
- G와 다른 도메인 사진 x

Image2StyleGAN [Brock et al.] 2019



- domain-guided encoder
- StyleGAN (FFHQ): W
- domain-guided encoder
- Reconstruction OK
- G와 다른 도메인 사진 x

In-Domain GAN Inversion [Zhu et al.] 2020

➡ Text-Guided Image Manipulation

Index

Composing Text and Image for Image Retrieval~

~> Text-guided image manipulation

Text Guided Image Manipulation

Text to Image Synthesis

- 1) Generative adversarial text to image synthesis (ICML, 2016)
- 2) StackGAN (ICCV, 2017)
- 3) StackGAN++ (TPAMI, 2017)
- 4) Semantic Image Synthesis via Adversarial Learning (ICCV, 2017)
- 5) AttnGAN (CVPR, 2018)
- 6) TaGAN (NIPS, 2018)

- **Generative adversarial text to image synthesis** (ICML, 2016)

  It is the first end-to-end differentiable architecture from the character level to pixel level (Conditions on Text Descriptions)

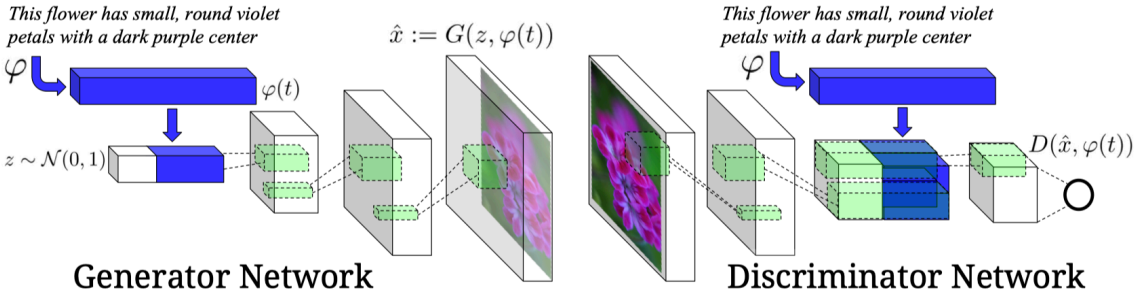


Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

Two sub problems:

1. Learn a text feature representation that captures the important visual details
2. Use these features to synthesis a compelling image that a human might mistake for real

- **Generative adversarial text to image synthesis** (ICML, 2016)

  It is the first end-to-end differentiable architecture from the character level to pixel level (Conditions on Text Descriptions)

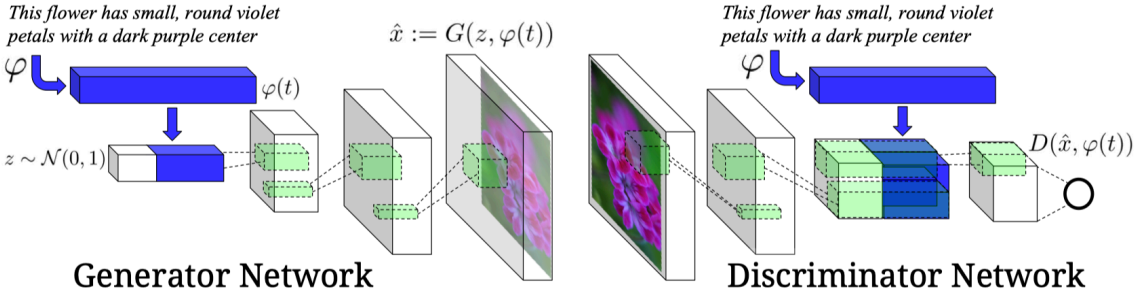


Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

A. Deep symmetric structured joint embedding (Text Encoder)
B. Matching-aware discriminator (GAN-CLS)
C. Learning with manifold interpolation (GAN-INT)
D. Inverting the generator for style transfer

- **Generative adversarial text to image synthesis** (ICML, 2016)

A. Deep symmetric structured joint embedding

$$\frac{1}{N}\sum_{n=1}^{N}\Delta(y_n, f_v(v_n)) + \Delta(y_n, f_t(t_n)) \qquad (2)$$

where $\{(v_n, t_n, y_n) : n = 1, ..., N\}$ is the training data set, Δ is the 0-1 loss, v_n are the images, t_n are the corresponding text descriptions, and y_n are the class labels. Classifiers f_v and f_t are parametrized as follows:

$$f_v(v) = \arg\max_{y \in \mathcal{Y}} \mathbb{E}_{t \sim \mathcal{T}(y)}[\phi(v)^T \varphi(t))] \qquad (3)$$

$$f_t(t) = \arg\max_{y \in \mathcal{Y}} \mathbb{E}_{v \sim \mathcal{V}(y)}[\phi(v)^T \varphi(t))] \qquad (4)$$

where ϕ is the image encoder (e.g. a deep convolutional neural network), φ is the text encoder (e.g. a character-level CNN or LSTM), $\mathcal{T}(y)$ is the set of text descriptions of class y and likewise $\mathcal{V}(y)$ for images. The intuition here is that a text encoding should have a higher compatibility score with images of the correspondong class compared to any other class and vice-versa.

GoogleNet (Szegedy, Christian, et al. 2015)
Imagenet pretrained

Char-CNN-RNN(Reed et al, 2016)
Oxford 102, CUB (pre-train)

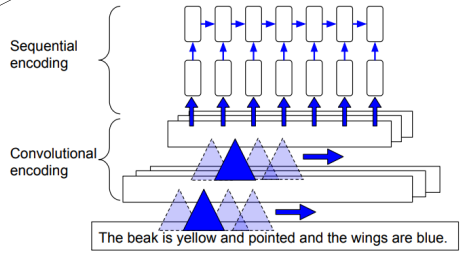


Figure 2: Our proposed convolutional-recurrent net.

• Generative adversarial text to image synthesis (ICML, 2016)

B. Matching-aware discriminator (GAN-CLS)

The most straightforward way to train a conditional GAN is to view (text, image) pairs as joint observations and train the discriminator to judge pairs as real or fake.
This type of conditioning is naive in the sense that the discriminator has no explicit notion of whether real training images match the text embedding context.

$$\min_{\theta_G} \max_{\theta_D} L(D,G) = \mathbb{E}_{\mathbf{t}\sim p_d(\mathbf{t}),x\sim p_d(x)} \log D_{\theta_D}(x,\varphi(\mathbf{t})) + \mathbb{E}_{\mathbf{t}\sim p_d(\mathbf{t}),z\sim p_z(z)} \log(1 - D_{\theta_D}(G_{\theta_G}(\varphi(\mathbf{t}),\mathbf{z}),\varphi(\mathbf{t})))$$

In naïve GAN, Discriminator inputs:
1) Real images with matching text
2) Fake images with arbitrary text

Therefore, it must implicitly separate two sources of error:
→ Unrealistic Images(for any text)
→ Realistic Images with mismatch text

Discriminator can provide an additional signal to Generator

Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

1: **Input:** minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
2: **for** $n = 1$ **to** S **do**
3: $h \leftarrow \varphi(t)$ {Encode matching text description}
4: $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
5: $z \sim \mathcal{N}(0,1)^Z$ {Draw sample of random noise}
6: $\hat{x} \leftarrow G(z,h)$ {Forward through generator}
7: $s_r \leftarrow D(x,h)$ {real image, right text}
8: $s_w \leftarrow D(x,\hat{h})$ {real image, wrong text}
9: $s_f \leftarrow D(\hat{x},h)$ {fake image, right text}
10: $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1-s_w) + \log(1-s_f))/2$
11: $D \leftarrow D - \alpha\partial\mathcal{L}_D/\partial D$ {Update discriminator}
12: $\mathcal{L}_G \leftarrow \log(s_f)$
13: $G \leftarrow G - \alpha\partial\mathcal{L}_G/\partial G$ {Update generator}
14: **end for**

• Generative adversarial text to image synthesis (ICML, 2016)

C. Learning with manifold interpolation (GAN-INT)

Deep networks have been shown to learn representations in which interpolations between embedding pairs tend to be near the data manifold

This can be viewed as adding an additional term to the generator objective to minimize

$$\mathbb{E}_{t_1, t_2 \sim p_d(t)} [\log(1 - D(G(z, \beta\varphi(t_1) + (1-\beta)\varphi(t_2))))]$$

Because the interpolated embeddings are synthetic, the discriminator D does not have "real" corresponding image and text pairs to train on.

However, D learns to predict whether image and text pairs match or not.

Thus, if D does a good job at this, then by satisfying D on interpolated text embeddings G can learn to fill in gaps on the data manifold in between training points.

Note that t_1 *and* t_2 may come from different images and even different categories.

• Generative adversarial text to image synthesis (ICML, 2016)

D. Inverting the generator for style transfer

If the text encoding $\varphi(t)$ captures the image content (e.g. flower shape and colors), then in order to generate a realistic image the noise sample z should capture style factors such as background color and pose. With a trained GAN, one may wish to transfer the style of a query image onto the content of a particular text description. To achieve this, one can train a convolutional network to invert G to regress from samples $\hat{x} \leftarrow G(z, \varphi(t))$ back onto z. We used a simple squared loss to train the style encoder:

$$\mathcal{L}_{style} = \mathbb{E}_{t,z\sim\mathcal{N}(0,1)}||z - S(G(z, \varphi(t)))||_2^2 \qquad (6)$$

where S is the style encoder network. With a trained generator and style encoder, style transfer from a query image x onto text t proceeds as follows:

$$s \leftarrow S(x), \hat{x} \leftarrow G(s, \varphi(t))$$

where $\hat{x}$ is the result image and s is the predicted style.

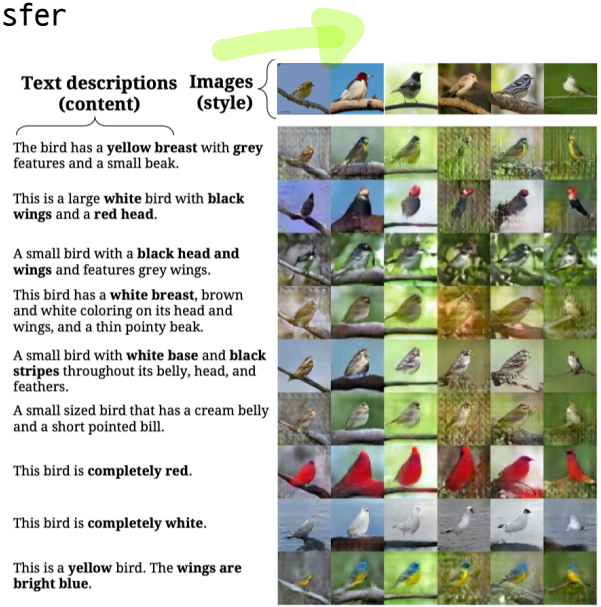


Figure 6. Transfering style from the top row (real) images to the content from the query text, with G acting as a deterministic decoder. The bottom three rows are captions made up by us.

- Text Guide Image Manipulation (TGIM)
- Text-to-Image Synthesis (T2I)

	TGIM	T2I	Negative text	Text + Image
1- Style Encoder	O	X	X	Concat
1- GAN-CLS, INT	X	O	Random	Concat

• StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Network (ICCV, 2017)

Abstract

Synthesizing high-quality images from text descriptions is a challenging problem in computer vision and has many practical applications. Samples generated by existing text-to-image approaches can roughly reflect the meaning of the given descriptions, but they fail to contain necessary details and vivid object parts. In this paper, we propose Stacked Generative Adversarial Networks (StackGAN) to generate 256×256 photo-realistic images conditioned on text descriptions. We decompose the hard problem into more manageable sub-problems through a sketch-refinement process. 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. It is able to rectify defects in Stage-I results and add compelling details with the refinement process. To improve the diversity of the synthesized images and stabilize the training of the conditional-GAN, we introduce a novel Conditioning Augmentation technique that encourages smoothness in the latent conditioning manifold. Extensive experiments and comparisons with state-of-the-arts on benchmark datasets demonstrate that the proposed method achieves significant improvements on generating photo-realistic images conditioned on text descriptions.

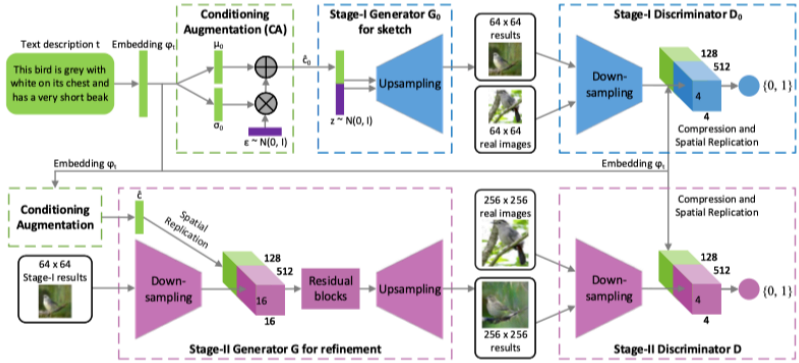


Figure 2. The architecture of the proposed StackGAN. The Stage-I generator draws a low-resolution image by sketching rough shape and basic colors of the object from the given text and painting the background from a random noise vector. Conditioned on Stage-I results, the Stage-II generator corrects defects and adds compelling details into Stage-I results, yielding a more realistic high-resolution image.

Decompose the hard problem into sub-problems!!

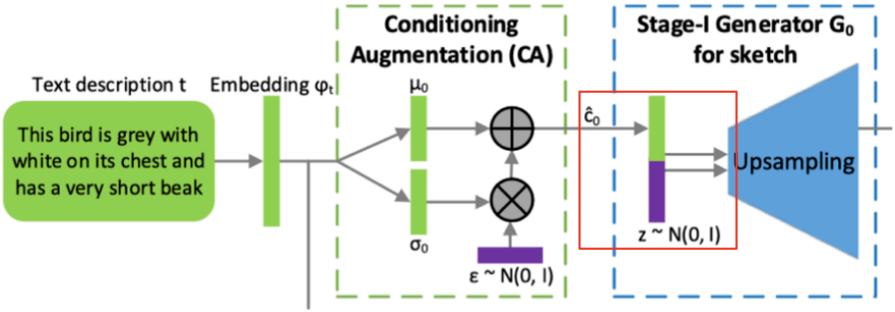
A.Stage-1: sketch the primitive shape and colors
B.Stage-2: rectify defects

- **StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Network** (ICCV, 2017)

A. Conditioning Augmentation

With limited amount of data, it usually causes discontinuity in the latent data manifold, which is not desirable for learning the generator.

To mitigate this problem, we introduce a Conditioning Augmentation technique to produce additional conditioning variables $\hat{c} \sim \mathcal{N}(\mu(\varphi_t), \sum(\varphi_t))$, $\hat{c}_0 = \mu_0 + \sigma_0 \odot \epsilon$, $\epsilon \sim \mathcal{N}(0,1)$



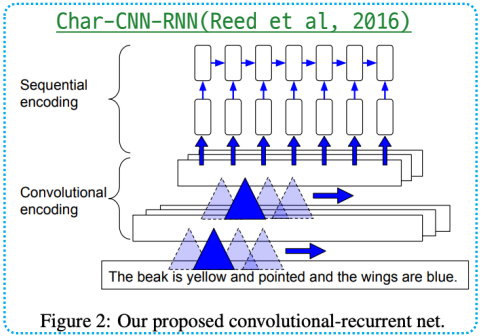


Figure 2: Our proposed convolutional-recurrent net.

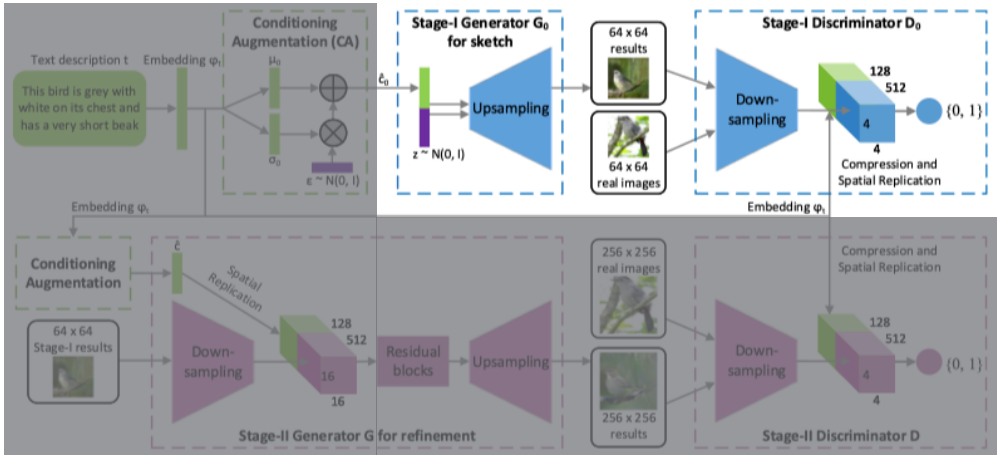
To further enforce the smoothness over the conditioning manifold and avoiding overfitting, add the regularization term to loss of G

$$D_{KL}(\mathcal{N}(\mu(\varphi_t), \sum(\varphi_t)) \parallel \mathcal{N}(0, I))$$

• StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Network (ICCV, 2017)

C. Stage-1

Instead of directly generating a high-resolution image conditioned on the text description, We simplify the task to first generate a low-resolution image with our Stage-I GAN, which focuses on drawing only rough shape and correct colors for the object.



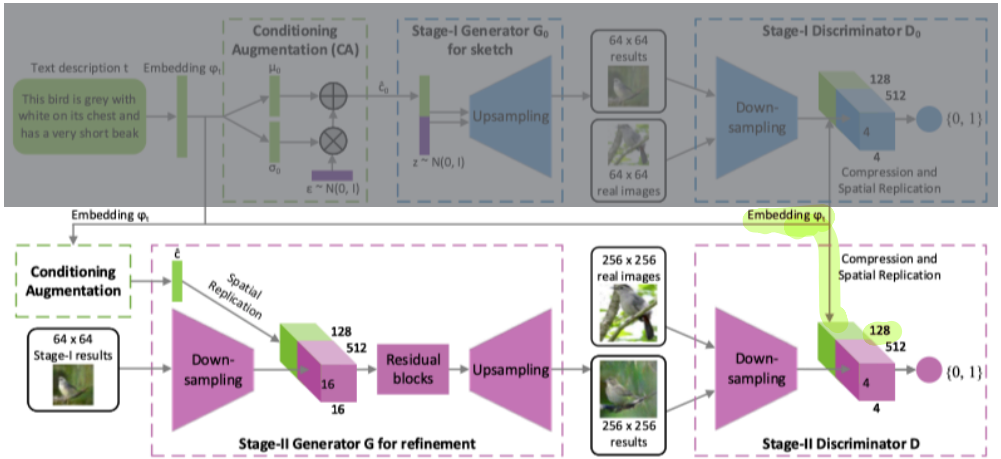
$$\mathcal{L}_{D_0} = \mathbb{E}_{(I_0,t)\sim p_{data}}[\log D_0(I_0, \varphi_t)] + \mathbb{E}_{z\sim p_z, t\sim p_{data}}[\log(1 - D_0(G_0(z, \hat{c}_0), \varphi_t))], \quad (3)$$

$$\mathcal{L}_{G_0} = \mathbb{E}_{z\sim p_z, t\sim p_{data}}[\log(1 - D_0(G_0(z, \hat{c}_0), \varphi_t))] + \lambda D_{KL}(\mathcal{N}(\mu_0(\varphi_t), \Sigma_0(\varphi_t)) \,||\, \mathcal{N}(0, I)), \quad (4)$$

- **StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Network** (ICCV, 2017)

D. Stage-2

Stage-2 GAN is built upon Stage-1 GAN results to generate high-resolution images.
It is conditioned on low-resolution images and also the text embedding again to correct defeats In Stage-1 results. The Stage-2 Gan completes previously ignored text information to generate more Photo-realistic details.



$$\mathcal{L}_D = \mathbb{E}_{(I,t)\sim p_{data}}[\log D(I, \varphi_t)] + \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{data}}[\log(1 - D(G(s_0, \hat{c}), \varphi_t))], \quad (5)$$

$$\mathcal{L}_G = \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{data}}[\log(1 - D(G(s_0, \hat{c}), \varphi_t))] + \lambda D_{KL}(\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t)) \,||\, \mathcal{N}(0, I)), \quad (6)$$

• StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Network (ICCV, 2017)

Figure 3. Example results by our StackGAN, GAWWN [24], and GAN-INT-CLS [26] conditioned on text descriptions from CUB test set.

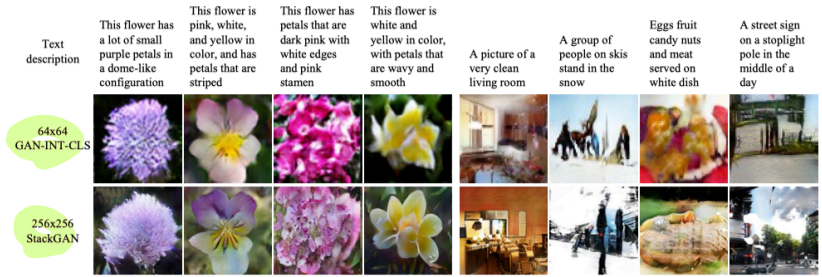


Figure 4. Example results by our StackGAN and GAN-INT-CLS [26] conditioned on text descriptions from Oxford-102 test set (leftmost four columns) and COCO validation set (rightmost four columns).

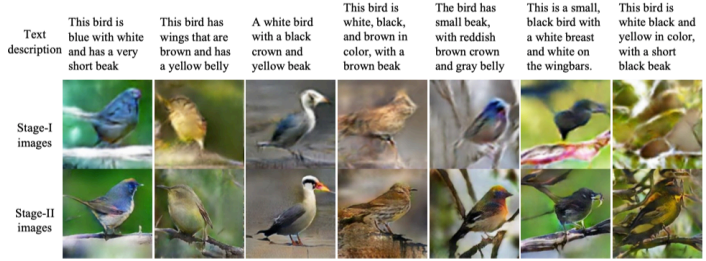


Figure 5. Samples generated by our StackGAN from unseen texts in CUB test set. Each column lists the text description, images generated from the text by Stage-I and Stage-II of StackGAN.

Figure 7. Conditioning Augmentation (CA) helps stabilize the training of conditional GAN and improves the diversity of the generated samples. (Row 1) without CA, Stage-I GAN fails to generate plausible 256×256 samples. Although different noise vector z is used for each column, the generated samples collapse to be the same for each input text description. (Row 2-3) with CA but fixing the noise vectors z, methods are still able to generate birds with different poses and viewpoints.

- **StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks** (TPAMI, 2017)

1. Multi-scale image distributions approximation
2. Joint conditional and unconditional distribution approximation

x

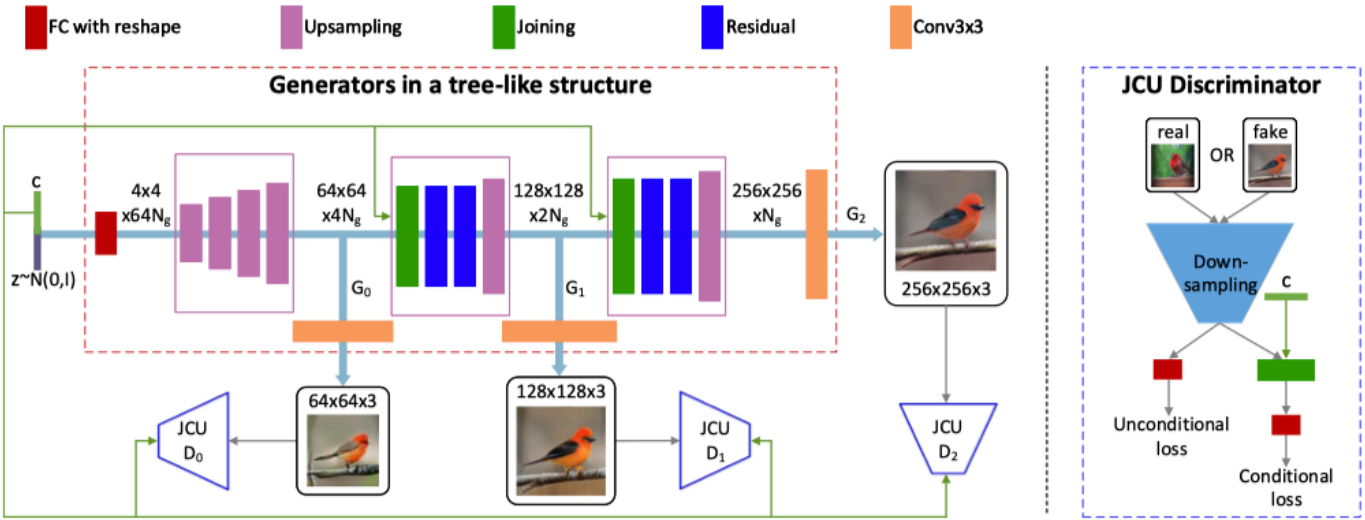


Fig. 2: The overall framework of our proposed StackGAN-v2 for the conditional image synthesis task. c is the vector of conditioning variables which can be computed from the class label, the text description, *etc.*. N_g and N_d are the numbers of channels of a tensor.

• StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks (TPAMI, 2017)

1. Multi-scale image distributions approximation

Our StackGAN-v2 framework has a tree-like structure, which takes a noise vector $z \sim p_{noise}$ as the input and has multiple generators to produce images of different scales. The p_{noise} is a prior distribution, which is usually chosen as the standard normal distribution. The latent variables z are transformed to hidden features layer by layer. We compute the hidden features h_i for each generator G_i by a non-linear transformation,

$$h_0 = F_0(z); \quad h_i = F_i(h_{i-1}, z), \ i = 1, 2, ..., m-1, \quad (7)$$

where h_i represents hidden features for the i^{th} branch, m is the total number of branches, and F_i are modeled as neural networks (see Fig. 2 for illustration). In order to capture information omitted in preceding branches, the noise vector z is concatenated to the hidden features h_{i-1} as the inputs of F_i for calculating h_i. Based on hidden features at different layers (h_0, h_1, ..., h_{m-1}), generators produce samples of small-to-large scales (s_0, s_1, ..., s_{m-1}),

$$s_i = G_i(h_i), \ i = 0, 1, ..., m-1, \quad (8)$$

where G_i is the generator for the i^{th} branch.

Following each generator G_i, a discriminator D_i, which takes a real image x_i or a fake sample s_i as input, is trained to classify inputs into two classes (real or fake) by minimizing the following cross-entropy loss,

$$\mathcal{L}_{D_i} = -\mathbb{E}_{x_i \sim p_{data_i}}[\log D_i(x_i)] - \mathbb{E}_{s_i \sim p_{G_i}}[\log(1 - D_i(s_i))], \quad (9)$$

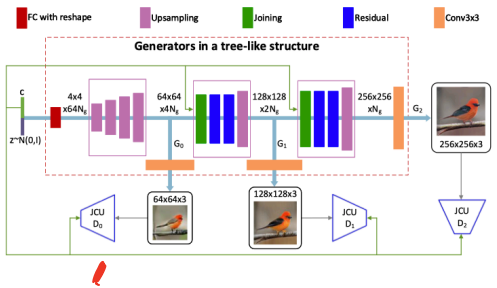
where x_i is from the true image distribution p_{data_i} at the i^{th} scale, and s_i is from the model distribution p_{G_i} at the same scale. The multiple discriminators are trained in parallel, and each of them focuses on a single image scale.

Guided by the trained discriminators, the generators are optimized to jointly approximate multi-scale image distributions ($p_{data_0}, p_{data_1}, ..., p_{data_{m-1}}$) by minimizing the following loss function,

$$\mathcal{L}_G = \sum_{i=1}^{m} \mathcal{L}_{G_i}, \quad \mathcal{L}_{G_i} = -\mathbb{E}_{s_i \sim p_{G_i}}[\log D_i(s_i)], \quad (10)$$

where $\mathcal{L}_{G_i}$ is the loss function for approximating the image distribution at the i^{th} scale (*i.e.*, p_{data_i}). During the training process, the discriminators D_i and the generators G_i are alternately optimized till convergence.

The motivation of the proposed StackGAN-v2 is that, by modeling data distributions at multiple scales, if any one of those model distributions shares support with the real data distribution at that scale, the overlap could provide good gradient signal to expedite or stabilize training of the whole network at multiple scales. For instance, approximating the low-resolution image distribution at the first branch results in images with basic color and structures. Then the generators at the subsequent branches can focus on completing details for generating higher resolution images.



• StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks (TPAMI, 2017)

2. Joint conditional and unconditional distribution approximation

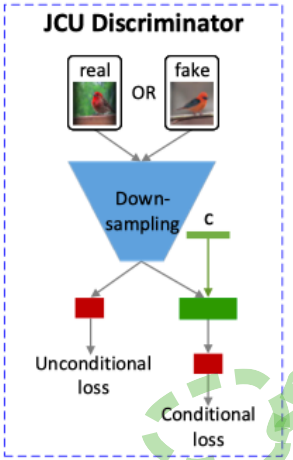
For the generator of our conditional StackGAN-v2, F_0 and F_i are converted to take the conditioning vector c as input, such that $h_0 = F_0(c, z)$ and $h_i = F_i(h_{i-1}, c)$. For F_i, the conditioning vector c replaces the noise vector z to encourage the generators to draw images with more details according to the conditioning variables. Consequently, multi-scale samples are now generated by $s_i = G_i(h_i)$. The objective function of training the discriminator D_i for conditional StackGAN-v2 now consists of two terms, the unconditional loss and the conditional loss,

$$\mathcal{L}_{D_i} = \underbrace{-\mathbb{E}_{x_i \sim p_{data_i}} [\log D_i(x_i)] - \mathbb{E}_{s_i \sim p_{G_i}} [\log(1 - D_i(s_i)]}_{\text{unconditional loss}} + \underbrace{-\mathbb{E}_{x_i \sim p_{data_i}} [\log D_i(x_i, c)] - \mathbb{E}_{s_i \sim p_{G_i}} [\log(1 - D_i(s_i, c)]}_{\text{conditional loss}}. \tag{11}$$

The unconditional loss determines whether the image is real or fake while the conditional one determines whether the image and the condition match or not. Accordingly, the loss function for each generator G_i is converted to

$$\mathcal{L}_{G_i} = \underbrace{-\mathbb{E}_{s_i \sim p_{G_i}} [\log D_i(s_i)]}_{\text{unconditional loss}} + \underbrace{-\mathbb{E}_{s_i \sim p_{G_i}} [\log D_i(s_i, c)]}_{\text{conditional loss}}. \tag{12}$$

The generator G_i at each scale therefore jointly approximates unconditional and conditional image distributions. The final loss for jointly training generators of conditional StackGAN-v2 is computed by substituting Eq. (12) into Eq. (10).



Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

1: **Input:** minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
2: **for** $n = 1$ **to** S **do**
3: $h \leftarrow \varphi(t)$ {Encode matching text description}
4: $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
5: $z \sim \mathcal{N}(0, 1)^Z$ {Draw sample of random noise}
6: $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
7: $s_r \leftarrow D(x, h)$ {real image, right text}
8: $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
9: $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
10: $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
11: $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
12: $\mathcal{L}_G \leftarrow \log(s_f)$
13: $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
14: **end for**

- StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks (TPAMI, 2017)

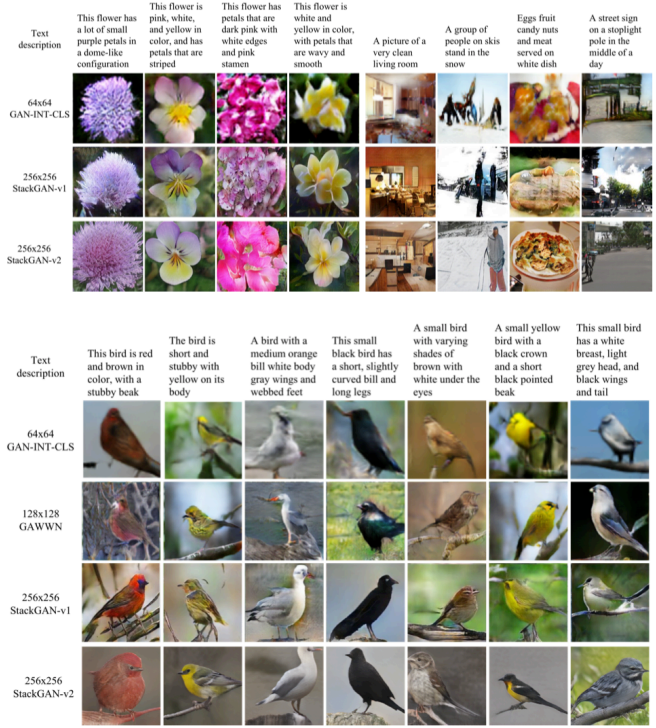


Fig. 3: Example results by our StackGANs, GAWWN [33], and GAN-INT-CLS [35] conditioned on text descriptions from CUB test set.

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 5: 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.

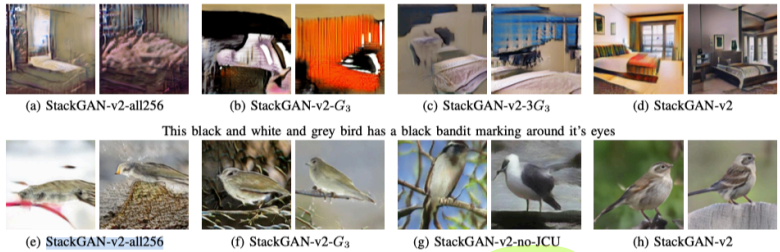


Fig. 14: Example images generated by the StackGAN-v2 and its baseline models on LSUN bedroom (top) and CUB (bottom) datasets.

- Text Guide Image Manipulation (TGIM)
- Text-to-Image Synthesis (T2I)

	TGIM	T2I	Negative text	Text + Image
1- Style Encoder	O	X	X	Concat
1- GAN-CLS, INT	X	O	Random	Concat
2- StackGAN,++	X	O	Random	Concat

• Semantic Image Synthesis via Adversarial Learning (ICCV, 2017)

Abstract

In this paper, we propose a way of synthesizing realistic images directly with natural language description, which has many useful applications, e.g. *intelligent image manipulation. We attempt to accomplish such synthesis: given a source image and a target text description, our model synthesizes images to meet two requirements: 1) being realistic while matching the target text description; 2) maintaining other image features that are irrelevant to the text description. The model should be able to disentangle the semantic information from the two modalities (image and text), and generate new images from the combined semantics. To achieve this, we proposed an end-to-end neural architecture that leverages adversarial learning to automatically learn implicit loss functions, which are optimized to fulfill the aforementioned two requirements. We have evaluated our model by conducting experiments on Caltech-200 bird dataset and Oxford-102 flower dataset, and have demonstrated that our model is capable of synthesizing realistic images that match the given descriptions, while still maintain other features of original images.*

Two sub problems:

1. Realistic image
2. Preserve & change

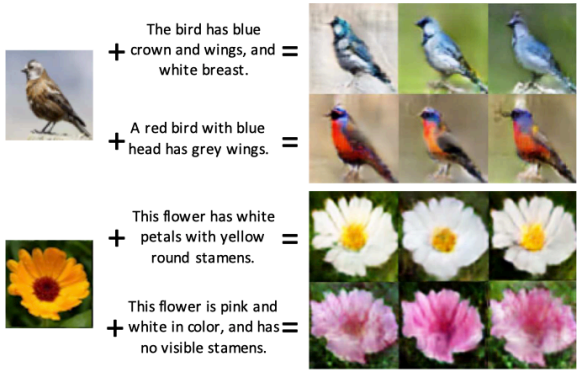


Figure 1. Examples of flower and bird images synthesized by our model from given source images and target text descriptions. Both source images and target text descriptions are unseen during training, demonstrating zero-shot learning ability of our model.

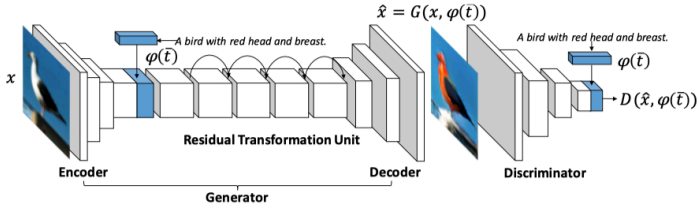


Figure 2. Network architecture of our proposed model. It consists of a generator network and a discriminator network. The generator has an encoder-decoder architecture and synthesizes images conditioned on both images and text embeddings. The discriminator performs the discriminative task conditioned on text embeddings.

• Semantic Image Synthesis via Adversarial Learning (ICCV, 2017)

A. Network architecture

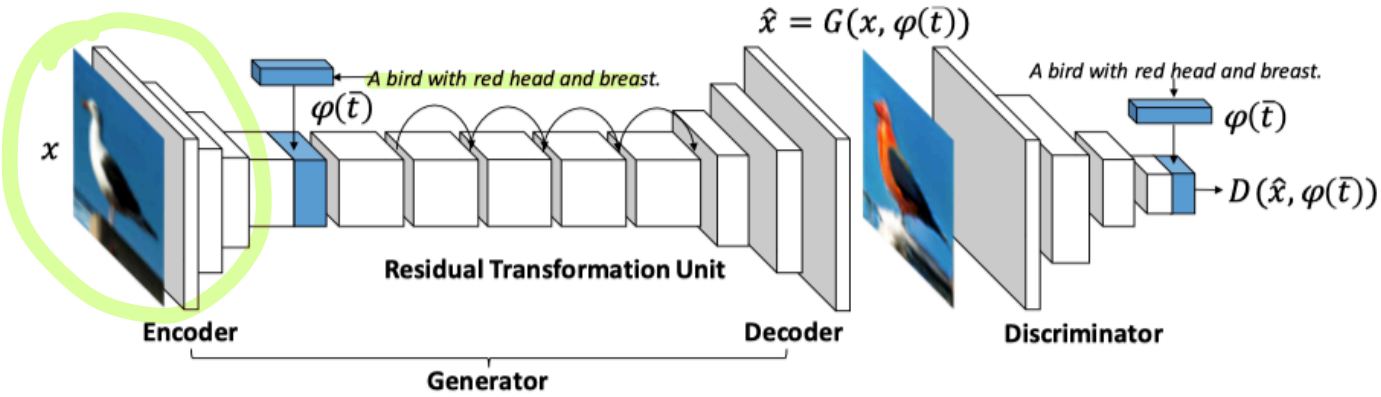


Figure 2. Network architecture of our proposed model. It consists of a generator network and a discriminator network. The generator has an encoder-decoder architecture and synthesizes images conditioned on both images and text embeddings. The discriminator performs the discriminative task conditioned on text embeddings.

- **Semantic Image Synthesis via Adversarial Learning** (ICCV, 2017)

Text encoder; VSE

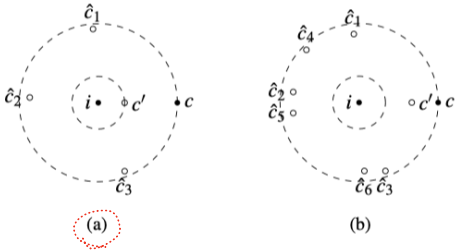


Figure 1: An illustration of typical positive pairs and the nearest negative samples. Here assume similarity score is the negative distance. Filled circles show a positive pair (i, c), while empty circles are negative samples for the query i. The dashed circles on the two sides are drawn at the same radii. Notice that the hardest negative sample c' is closer to i in (a). Assuming a zero margin, (b) has a higher loss with the *SH* loss compared to (a). The *MH* loss assigns a higher loss to (a).

$$\min_{\theta} \sum_{x} \sum_{k} max\{0, \alpha - s(\phi(x), \varphi(t)) + s(\phi(x), \varphi(t_k))\}$$
$$+ \sum_{t} \sum_{k} max\{0, \alpha - s(\phi(x), \varphi(t)) + s(\phi(x_k), \varphi(t))\} \quad (4)$$

• Semantic Image Synthesis via Adversarial Learning (ICCV, 2017)

B. Adaptive loss for semantic image synthesis

In our approach, we feed the discriminator D with three types of input pairs, and the outputs of discriminator D are the independent probabilities of these types:

- $s_r^+ \leftarrow D(x, \varphi(t))$ for real image with matching text;
- $s_w^- \leftarrow D(x, \varphi(\hat{t}))$ for real image with mismatching text;
- $s_s^- \leftarrow D(\hat{x}, \varphi(\bar{t}))$ for synthesized image with semantically relevant text.

where $+$ and $-$ denote positive and negative examples respectively.

The term s_w^-, proposed by Reed *et al.* [29], enables the discriminator to generate stronger image / text matching signal, which makes the generator G able to synthesize realistic images that better match the text descriptions. G synthesizes images via $\hat{x} \leftarrow G(x, \varphi(\bar{t}))$ and is optimized adversarially with D.

Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0, 1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

• Semantic Image Synthesis via Adversarial Learning (ICCV, 2017)

B. Adaptive loss for semantic image synthesis

In our approach, we feed the discriminator D with three types of input pairs, and the outputs of discriminator D are the independent probabilities of these types:

- $s_r^+ \leftarrow D(x, \varphi(t))$ for real image with matching text;
- $s_w^- \leftarrow D(x, \varphi(\hat{t}))$ for real image with mismatching text;
- $s_s^- \leftarrow D(\hat{x}, \varphi(\bar{t}))$ for synthesized image with semantically relevant text.

where $+$ and $-$ denote positive and negative examples respectively.

The term s_w^-, proposed by Reed *et al.* [29], enables the discriminator to generate stronger image / text matching signal, which makes the generator G able to synthesize realistic images that better match the text descriptions. G synthesizes images via $\hat{x} \leftarrow G(x, \varphi(\bar{t}))$ and is optimized adversarially with D.

$$
\begin{aligned}
\mathcal{L}_D &= \mathbb{E}_{(x,t)\sim p_{data}} log D(x, \varphi(t)) \\
&\quad + \mathbb{E}_{(x,\hat{t})\sim p_{data}} log(1 - D(x, \varphi(\hat{t}))) \\
&\quad + \mathbb{E}_{(x,\bar{t})\sim p_{data}} log(1 - D(G(x, \varphi(\bar{t})), \varphi(\bar{t}))) \\
\mathcal{L}_G &= \mathbb{E}_{(x,\bar{t})\sim p_{data}} log(D(G(x, \varphi(\bar{t})), \varphi(\bar{t})))
\end{aligned}
\tag{3}
$$

where t denotes matching text, $\hat{t}$ denotes mismatching text, $\bar{t}$ denotes semantically relevant text. The generator $G(x, \varphi(\bar{t}))$ captures conditional generative distribution $p_G(\hat{x}|x, \bar{t})$, and the loss functions encourage the generator to fit the distribution of real data $p_{data}(x, \bar{t})$.

- **Semantic Image Synthesis via Adversarial Learning** (ICCV, 2017)

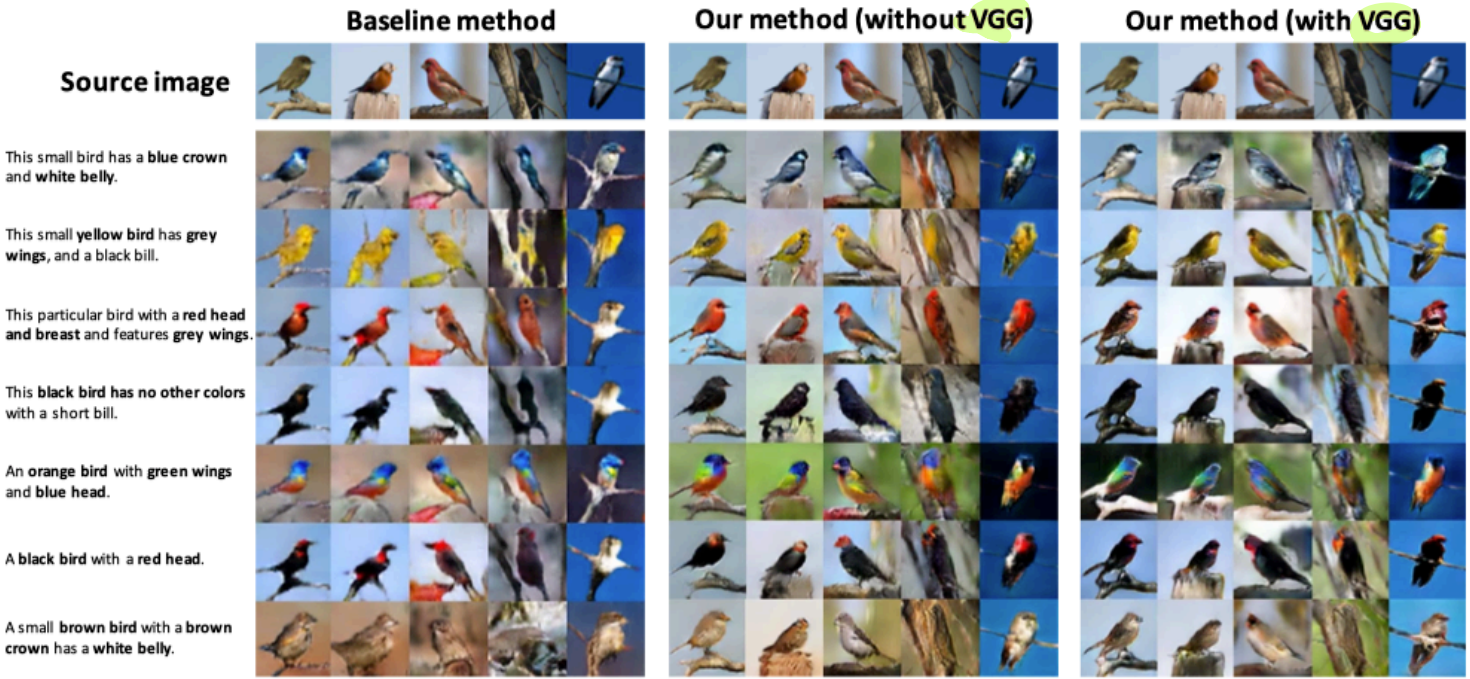


Figure 4. Zero-shot results of the baseline method and our method with and without pretrained VGG encoder on Caltech-200 bird dataset.

• Semantic Image Synthesis via Adversarial Learning (ICCV, 2017)

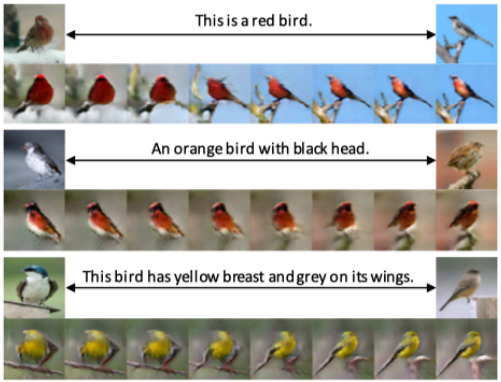


Figure 6. Zero-shot results of interpolation between two source images with the same target text description. The images pointed by arrows are the source images.

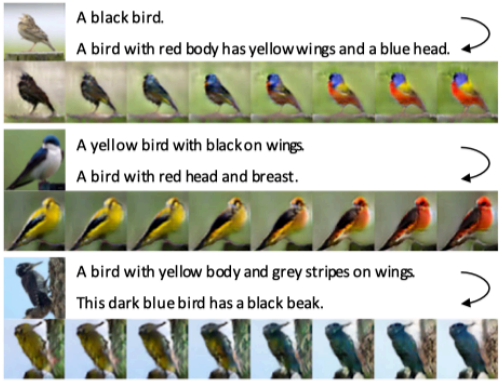


Figure 7. Zero-shot results of interpolation between two target text descriptions for the same source image. The images on the left-hand side of sentences are the source images.

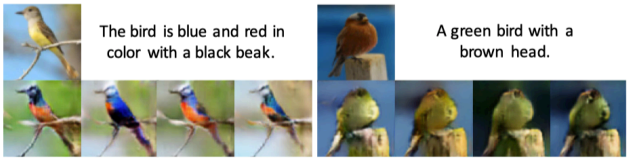


Figure 8. Zero-shot results from same source image and target text description for showing variety.

- Text Guide Image Manipulation (TGIM)
- Text-to-Image Synthesis (T2I)

	TGIM	T2I	Negative text	Text + Image
1 - Style Encoder	O	X	X	Concat
1 - GAN-CLS, INT	X	O	Random	Concat
2 - StackGAN,++	X	O	Random	Concat
3 - SISGAN	O	X	Random	Concat

Text Guided Image Manipulation

Text to Image Synthesis

Natural Language Processing & Artificial Intelligence

- Text encoder: Char-CNN-RNN, VSE
- Unconditional Loss & Conditioning Augmentation
- (image, wrong text), (image, relative text)

-> Image & Language는 단순 concat
-> Text는 단순 Sentence level
-> Spatial이나 word 혹은 channel에 대한 깊은 탐구 부족
-> Multimodal에 대한 alignment 부족

- 1) Generative adversarial text to image synthesis (ICML, 2016)
- 2) StackGAN (ICCV, 2017)
- 3) StackGAN++ (TPAMI, 2017)
- 4) Semantic Image Synthesis via Adversarial Learning (ICCV, 2017)
- 5) AttnGAN (CVPR, 2018)
- 6) TaGAN (NIPS, 2018)

고려대학교 컴퓨터학과 석사과정
2019011001 박성진

• AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Network (CVPR, 2018)

Abstract

In this paper, we propose an Attentional Generative Adversarial Network (AttnGAN) that allows attention-driven, multi-stage refinement for fine-grained text-to-image generation. With a novel attentional generative network, the AttnGAN can synthesize fine-grained details at different subregions of the image by paying attentions to the relevant words in the natural language description. In addition, a deep attentional multimodal similarity model is proposed to compute a fine-grained image-text matching loss for training the generator. The proposed AttnGAN significantly outperforms the previous state of the art, boosting the best reported inception score by 14.14% on the CUB dataset and 170.25% on the more challenging COCO dataset. A detailed analysis is also performed by visualizing the attention layers of the AttnGAN. It for the first time shows that the layered attentional GAN is able to automatically select the condition at the word level for generating different parts of the image.

Figure 1. Example results of the proposed AttnGAN. The first row gives the low-to-high resolution images generated by G_0, G_1 and G_2 of the AttnGAN; the second and third row shows the top-5 most attended words by F_1^{attn} and F_2^{attn} of the AttnGAN, respectively. Here, images of G_0 and G_1 are bilinearly upsampled to have the same size as that of G_2 for better visualization.

- AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Network (CVPR, 2018)

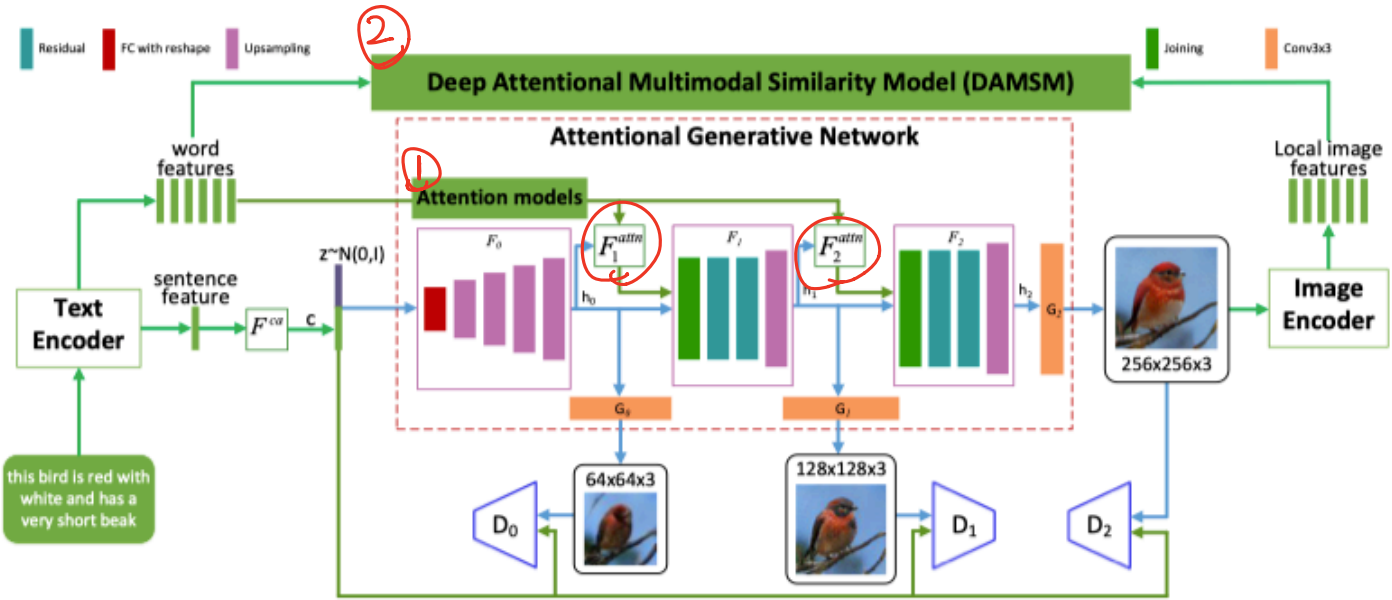
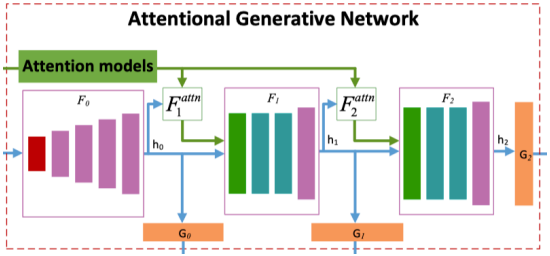


Figure 2. The architecture of the proposed AttnGAN. Each attention model automatically retrieves the conditions (*i.e.*, the most relevant word vectors) for generating different sub-regions of the image; the DAMSM provides the fine-grained image-text matching loss for the generative network.

• AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Network (CVPR, 2018)

A. Attention Generative Network



To generate realistic images with multiple levels (*i.e.*, sentence level and word level) of conditions, the final objective function of the attentional generative network is defined as

$$\mathcal{L} = \mathcal{L}_G + \lambda \mathcal{L}_{DAMSM}, \quad \text{where } \mathcal{L}_G = \sum_{i=0}^{m-1} \mathcal{L}_{G_i}. \tag{3}$$

Here, λ is a hyperparameter to balance the two terms of Eq. (3). The first term is the GAN loss that jointly approximates conditional and unconditional distributions [37]. At the i^{th} stage of the AttnGAN, the generator G_i has a corresponding discriminator D_i. The adversarial loss for G_i is defined as

$$\mathcal{L}_{G_i} = \underbrace{-\frac{1}{2}\mathbb{E}_{\hat{x}_i \sim p_{G_i}}[\log(D_i(\hat{x}_i))]}_{\text{unconditional loss}} \underbrace{-\frac{1}{2}\mathbb{E}_{\hat{x}_i \sim p_{G_i}}[\log(D_i(\hat{x}_i, \bar{e})]}_{\text{conditional loss}}, \tag{4}$$

where the unconditional loss determines whether the image is real or fake while the conditional loss determines whether the image and the sentence match or not.

Alternately to the training of G_i, each discriminator D_i is trained to classify the input into the class of real or fake by minimizing the cross-entropy loss defined by

$$\mathcal{L}_{D_i} = \underbrace{-\frac{1}{2}\mathbb{E}_{x_i \sim p_{data_i}}[\log D_i(x_i)] - \frac{1}{2}\mathbb{E}_{\hat{x}_i \sim p_{G_i}}[\log(1 - D_i(\hat{x}_i)]}_{\text{unconditional loss}} + \underbrace{-\frac{1}{2}\mathbb{E}_{x_i \sim p_{data_i}}[\log D_i(x_i, \bar{e})] - \frac{1}{2}\mathbb{E}_{\hat{x}_i \sim p_{G_i}}[\log(1 - D_i(\hat{x}_i, \bar{e})]}_{\text{conditional loss}}, \tag{5}$$

where x_i is from the true image distribution p_{data_i} at the i^{th} scale, and $\hat{x}_i$ is from the model distribution p_{G_i} at the same scale. Discriminators of the AttnGAN are structurally disjoint, so they can be trained in parallel and each of them focuses on a single image scale.

- **AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Network** (CVPR, 2018)

B. Deep Attentional Multimodal Similarity Model (DAMSM)

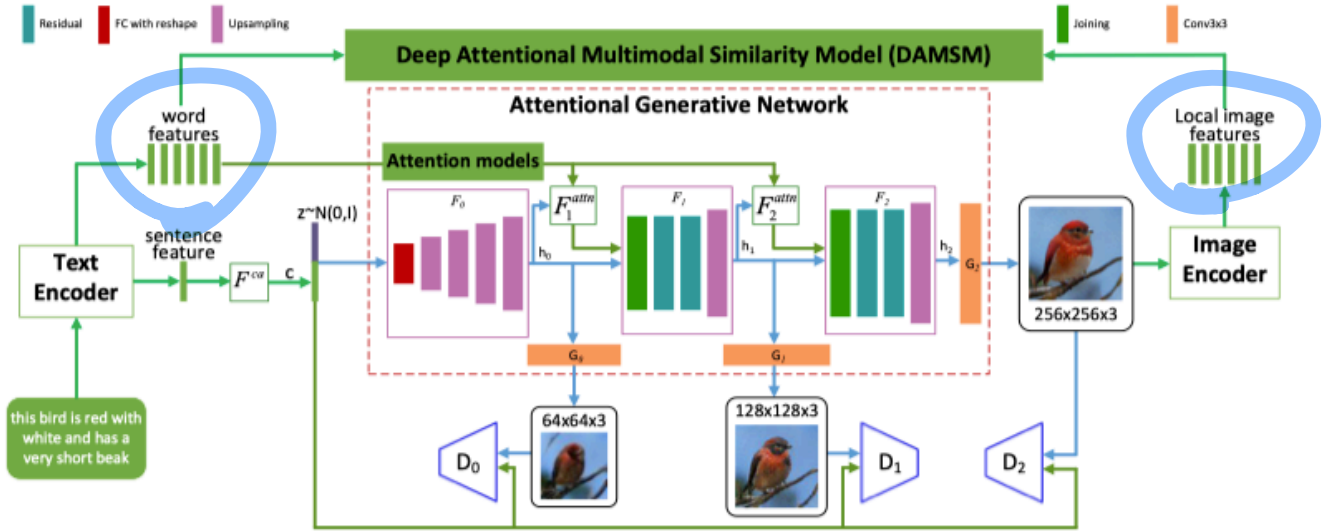


Figure 2. The architecture of the proposed AttnGAN. Each attention model automatically retrieves the conditions (*i.e.*, the most relevant word vectors) for generating different sub-regions of the image; the DAMSM provides the fine-grained image-text matching loss for the generative network.

• AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Network (CVPR, 2018)

B. Deep Attentional Multimodal Similarity Model (DAMSM)

The text encoder is a bi-directional Long Short-Term Memory (LSTM) [25] that extracts semantic vectors from the text description. In the bi-directional LSTM, each word corresponds to two hidden states, one for each direction. Thus, we concatenate its two hidden states to represent the semantic meaning of a word. The feature matrix of all words is indicated by $e \in \mathbb{R}^{D \times T}$. Its i^{th} column e_i is the feature vector for the i^{th} word. D is the dimension of the word vector and T is the number of words. Meanwhile, the last hidden states of the bi-directional LSTM are concatenated to be the global sentence vector, denoted by $\overline{e} \in \mathbb{R}^{D}$.

The image encoder is a Convolutional Neural Network (CNN) that maps images to semantic vectors. The intermediate layers of the CNN learn local features of different sub-regions of the image, while the later layers learn global features of the image. More specifically, our image encoder is built upon the Inception-v3 model [26] pretrained on ImageNet [22]. We first rescale the input image to be 299×299 pixels. And then, we extract the local feature matrix $f \in \mathbb{R}^{768 \times 289}$ (reshaped from 768×17×17) from the "*mixed_6e*" layer of Inception-v3. Each column of f is the feature vector of a sub-region of the image. 768 is the dimension of the local feature vector, and 289 is the number of sub-regions in the image. Meanwhile, the global feature vector $\overline{f} \in \mathbb{R}^{2048}$ is extracted from the last average pooling layer of Inception-v3. Finally, we convert the image features to a common semantic space of text features by adding a perceptron layer:

$$v = Wf, \qquad \overline{v} = \overline{W}\,\overline{f}, \tag{6}$$

where $v \in \mathbb{R}^{D \times 289}$ and its i^{th} column v_i is the visual feature vector for the i^{th} sub-region of the image; and $\overline{v} \in \mathbb{R}^{D}$ is the global vector for the whole image. D is the dimension of the multimodal (*i.e.*, image and text modalities) feature space. For efficiency, all parameters in layers built from the Inception-v3 model are fixed, and the parameters in newly added layers are jointly learned with the rest of the network.

• AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Network (CVPR, 2018)

B. Deep Attentional Multimodal Similarity Model (DAMSM)

The attention-driven image-text matching score is designed to measure the matching of an image-sentence pair based on an attention model between the image and the text.

We first calculate the similarity matrix for all possible pairs of words in the sentence and sub-regions in the image by

$$s = e^T v, \tag{7}$$

where $s \in \mathbb{R}^{T \times 289}$ and $s_{i,j}$ is the dot-product similarity between the i^{th} word of the sentence and the j^{th} sub-region of the image. We find that it is beneficial to normalize the similarity matrix as follows

$$\bar{s}_{i,j} = \frac{\exp(s_{i,j})}{\sum_{k=0}^{T-1} \exp(s_{k,j})}. \tag{8}$$

Then, we build an attention model to compute a region-context vector for each word (query). The region-context vector c_i is a dynamic representation of the image's sub-regions related to the i^{th} word of the sentence. It is computed as the weighted sum over all regional visual vectors, *i.e.*,

$$c_i = \sum_{j=0}^{288} \alpha_j v_j, \quad \text{where } \alpha_j = \frac{\exp(\gamma_1 \bar{s}_{i,j})}{\sum_{k=0}^{288} \exp(\gamma_1 \bar{s}_{i,k})}. \tag{9}$$

Here, γ_1 is a factor that determines how much attention is paid to features of its relevant sub-regions when computing the region-context vector for a word.

Finally, we define the relevance between the i^{th} word and the image using the cosine similarity between c_i and e_i, *i.e.*, $R(c_i, e_i) = (c_i^T e_i)/(||c_i||||e_i||)$. Inspired by the minimum classification error formulation in speech recognition (see, *e.g.*, [11, 8]), the *attention-driven image-text matching score* between the entire image (Q) and the whole text description (D) is defined as

$$R(Q, D) = \log \Big(\sum_{i=1}^{T-1} \exp(\gamma_2 R(c_i, e_i)) \Big)^{\frac{1}{\gamma_2}}, \tag{10}$$

where γ_2 is a factor that determines how much to magnify the importance of the most relevant word-to-region-context pair. When $\gamma_2 \to \infty$, $R(Q, D)$ approximates to $\max_{i=1}^{T-1} R(c_i, e_i)$.

The DAMSM loss is designed to learn the attention model in a semi-supervised manner, in which the only supervision is the matching between entire images and whole sentences (a sequence of words). Similar to [4, 9], for a batch of image-sentence pairs $\{(Q_i, D_i)\}_{i=1}^{M}$, the posterior probability of sentence D_i being matching with image Q_i is computed as

$$P(D_i|Q_i) = \frac{\exp(\gamma_3 R(Q_i, D_i))}{\sum_{j=1}^{M} \exp(\gamma_3 R(Q_i, D_j))}, \tag{11}$$

where γ_3 is a smoothing factor determined by experiments. In this batch of sentences, only D_i matches the image Q_i, and treat all other $M-1$ sentences as mismatching descriptions. Following [4, 9], we define the loss function as the negative log posterior probability that the images are matched with their corresponding text descriptions (ground truth), *i.e.*,

$$\mathcal{L}_1^w = -\sum_{i=1}^{M} \log P(D_i|Q_i), \tag{12}$$

where 'w' stands for "word".

Symmetrically, we also minimize

$$\mathcal{L}_2^w = -\sum_{i=1}^{M} \log P(Q_i|D_i), \tag{13}$$

Finally, the DAMSM loss is defined as

$$\mathcal{L}_{DAMSM} = \mathcal{L}_1^w + \mathcal{L}_2^w + \mathcal{L}_1^s + \mathcal{L}_2^s. \tag{14}$$

- **AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Network** (CVPR, 2018)

Figure 5. Example results of our AttnGAN model trained on CUB while changing some most attended words in the text descriptions.

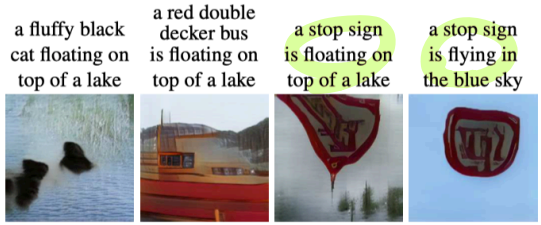


Figure 6. 256×256 images generated from descriptions of novel scenarios using the AttnGAN model trained on COCO. (Intermediate results are given in the supplementary material.)

Dataset	GAN-INT-CLS [20]	GAWWN [18]	StackGAN [36]	StackGAN-v2 [37]	PPGN [16]	Our AttnGAN
CUB	2.88 ± .04	3.62 ± .07	3.70 ± .04	3.84 ± .06	/	**4.36 ± .03**
COCO	7.88 ± .07	/	8.45 ± .03	/	9.58 ± .21	**25.89 ± .47**

Table 3. Inception scores by state-of-the-art GAN models [20, 18, 36, 37, 16] and our AttnGAN on CUB and COCO test sets.

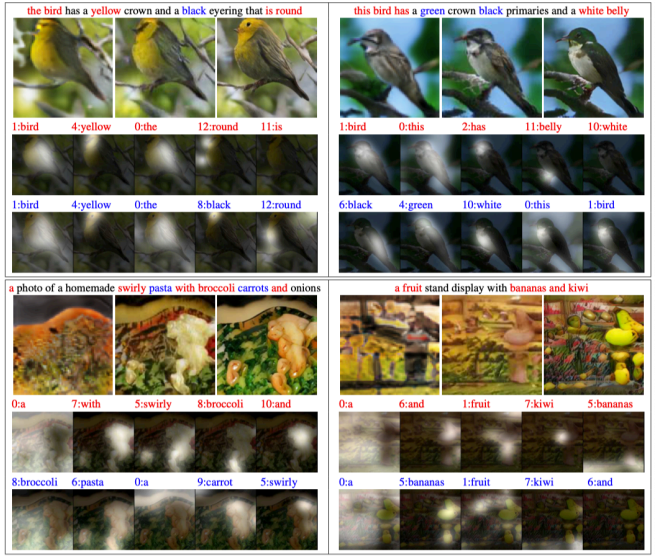


Figure 4. Intermediate results of our AttnGAN on CUB (top) and COCO (bottom) test sets. In each block, the first row gives 64×64 images by G_0, 128×128 images by G_1 and 256×256 images by G_2 of the AttnGAN; the second and third row shows the top-5 most attended words by F_1^{attn} and F_2^{attn} of the AttnGAN, respectively. Refer to the supplementary material for more examples.

- Text Guide Image Manipulation (TGIM)
- Text-to-Image Synthesis (T2I)

	TGIM	T2I	Negative text	Text + Image
1 - Style Encoder	O	X	X	Concat
1 - GAN-CLS, INT	X	O	Random	Concat
2 - StackGAN,++	X	O	Random	Concat
3 - SISGAN	O	X	Random	Concat
4 - AttnGAN	X	O	Random (constrastive)	Attention, Concat

- **Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language** (NIPS, 2018)

Abstract

This paper addresses the problem of manipulating images using natural language description. Our task aims to semantically modify visual attributes of an object in an image according to the text describing the new visual appearance. Although existing methods synthesize images having new attributes, they do not fully preserve text-irrelevant contents of the original image. In this paper, we propose the text-adaptive generative adversarial network (TAGAN) to generate semantically manipulated images while preserving text-irrelevant contents. The key to our method is the text-adaptive discriminator that creates word-level local discriminators according to input text to classify fine-grained attributes independently. With this discriminator, the generator learns to generate images where only regions that correspond to the given text are modified. Experimental results show that our method outperforms existing methods on CUB and Oxford-102 datasets, and our results were mostly preferred on a user study. Extensive analysis shows that our method is able to effectively disentangle visual attributes and produce pleasing outputs.

This particular bird with a **red head and breast** and features **grey wings**.

This small bird has a **blue crown** and **white belly**.

Figure 1: Examples of image manipulation using natural language description. Existing methods produce reasonable results, but fail to preserve text-irrelevant contents such as the background of the original image. In comparison, our method accurately manipulates images according to the text while preserving text-irrelevant contents.

Same as SISGAN (From original image)
- Preserve & change
- Realistic image
- Text adaptive, word-level
- Modifying color or textual

3번 (SISGAN)

1번 (Style Encoder)

- **Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language** (NIPS, 2018)

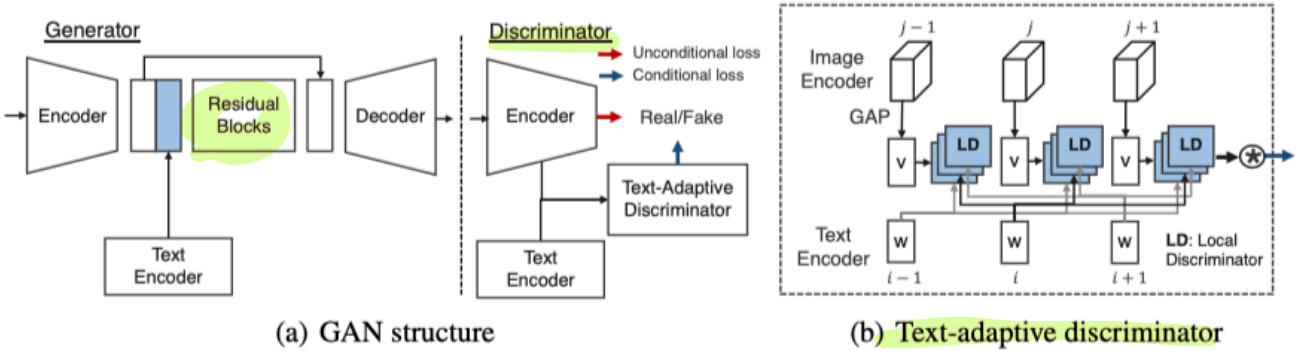


Figure 2: The proposed GAN structure. (a) shows the overall GAN architecture and (b) depicts our text-adaptive discriminator. In (b), the attention and the layer-wise weight are omitted for simplicity.

- **Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language** (NIPS, 2018)

Let $\mathbf{x}$, $\mathbf{t}$, $\hat{\mathbf{t}}$ denote an image, a positive text where the description matches the image, and a negative text that does not correctly describe the image, respectively. Given an image $\mathbf{x}$ and a target negative text $\hat{\mathbf{t}}$, our task is to semantically manipulate $\mathbf{x}$ according to $\hat{\mathbf{t}}$ so that the visual attributes of the manipulated image $\hat{\mathbf{y}}$ match the description of $\hat{\mathbf{t}}$ while preserving other information. We use GAN as our framework, in which the generator is trained to produce $\hat{\mathbf{y}} = G(\mathbf{x}, \hat{\mathbf{t}})$. Similar to text-to-image GANs [11, 15], we train our GAN to generate a realistic image that matches the conditional text semantically. In the following, we describe the TAGAN in detail.

- **Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language** (NIPS, 2018)

A. Generator

Generator The generator is an encoder-decoder network as shown in Fig. 2 (a)[1]. It first encodes an input image to a feature representation, then transforms it to a semantically manipulated representation according to the features of the given conditional text. For the text representation, we use a bidirectional RNN to encode the whole text. Unlike existing works [11, 15], we train the RNN from scratch, without pretraining. Additionally, we adopt the conditioning augmentation method [12] for smooth text representation and the diversity of generated outputs. As shown in Fig. 2 (a), manipulated contents are generated through several residual blocks with a skip connection. However, this process may generate a new background and other contents that are not described in the text. Therefore, we use the reconstruction loss [27] when a positive text is given, which enforces the generator to reconstruct the text-irrelevant contents from the input image instead of generating new contents:

$$L_{rec} = \|\mathbf{x} - G(\mathbf{x}, \mathbf{t})\|. \quad (1)$$

However, learning invariant representation is still difficult unless the discriminator provides useful feedback for disentangling visual attributes. To cope with it, we propose a text-adaptive discriminator.

• Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language (NIPS, 2018)

B. Text-adaptive discriminator

Fig. 2 (b) shows the structure of the text-adaptive discriminator. Similar to the generator, the discriminator is trained with its own text encoder. For each word vector $\mathbf{w}_i$, i-th output from the text encoder, we create 1D sigmoid local discriminator $f_{\mathbf{w}_i}$, which determines whether a visual attribute related to $\mathbf{w}_i$ exists in the image. Formally, $f_{\mathbf{w}_i}$ is described as:

$$f_{\mathbf{w}_i}(\mathbf{v}) = \sigma(\mathbf{W}(\mathbf{w}_i) \cdot \mathbf{v} + \mathbf{b}(\mathbf{w}_i)), \quad (2)$$

where $\mathbf{W}(\mathbf{w}_i)$ and $\mathbf{b}(\mathbf{w}_i)$ are the weight and the bias dependent on $\mathbf{w}_i$. $\mathbf{v}$ is an 1D image vector computed by applying global average pooling to the feature map of the image encoder.

With the local discriminators, the final classification decision is made by adding word-level attentions to reduce the impact of less important words to the final score. Our attention is a softmax values across T words, which is computed by:

$$\alpha_i = \frac{\exp(\mathbf{u}^T \mathbf{w}_i)}{\sum_i \exp(\mathbf{u}^T \mathbf{w}_i)}, \quad (3)$$

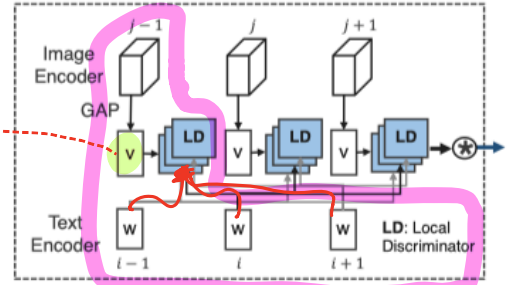
where $\mathbf{u}$ is a temporal average of $\mathbf{w}_i$. The final score is computed according to the following formulation:

$$D(\mathbf{x}, \mathbf{t}) = \prod_{i=1}^{T} [f_{\mathbf{w}_i}(\mathbf{v})]^{\alpha_i}. \quad (4)$$

We additionally consider multi-scale image features to make some attribute detectors to focus on small-scale features and others to focus on large-scale features. Therefore, our conditional discriminator is rewritten as:

$$D(\mathbf{x}, \mathbf{t}) = \prod_{i=1}^{T} [\sum_j \beta_{ij} f_{\mathbf{w}_i, j}(\mathbf{v}_j)]^{\alpha_i}, \quad (5)$$

where $\mathbf{v}_j$ is the image vector of j-th layer, and β_{ij} is a softmax weight that determines the importance of the layer j for each word $\mathbf{w}_i$.



(b) Text-adaptive discriminator

- **Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language** (NIPS, 2018)

C. Objectives

GAN objective The final GAN objective consists of unconditional adversarial losses for $D(\mathbf{x})$, text-conditional losses for $D(\mathbf{x}, \hat{\mathbf{t}})$, and a reconstruction loss as shown in Fig. 2. The discriminator has one image encoder and two branches of classifier on the top of the encoder to compute both the unconditional and the conditional losses. Our network is trained by alternatively minimizing both the discriminator and the generator objectives described as:

$$L_D = \mathbb{E}_{\mathbf{x},\mathbf{t},\hat{\mathbf{t}}\sim p_{data}}[\log D(\mathbf{x}) + \lambda_1(\log D(\mathbf{x}, \mathbf{t}) + \log(1 - D(\mathbf{x}, \hat{\mathbf{t}})))] + \mathbb{E}_{\mathbf{x},\hat{\mathbf{t}}\sim p_{data}}[\log(1 - D(G(\mathbf{x}, \hat{\mathbf{t}})))], \tag{6}$$

$$L_G = \mathbb{E}_{\mathbf{x},\hat{\mathbf{t}}\sim p_{data}}[\log D(\mathbf{x}) + \lambda_1 \log D(G(\mathbf{x}, \hat{\mathbf{t}}), \hat{\mathbf{t}})] + \lambda_2 L_{rec}, \tag{7}$$

where λ_1 and λ_2 control the importance of additional losses, and $\hat{\mathbf{t}}$ is randomly sampled from a dataset regardless of $\mathbf{x}$. Note that we do not penalize generated outputs using the conditional discriminator in Eq. (6) due to instability of training. In our experiment, our objective was enough to produce real images having manipulated attributes.

- **Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language** (NIPS, 2018)

Table 1: Quantitative comparison. Accuracy and Naturalness were evaluated by users, and the values indicate the average ranking. L_2 reconstruction error was additionally compared.

	CUB			Oxford-102		
Method	Accuracy	Naturalness	L_2 error	Accuracy	Naturalness	L_2 error
SISGAN [15]	2.33	2.34	0.30	2.67	2.28	0.29
AttnGAN [13]	2.19	2.11	0.25	2.21	2.10	0.32
Ours	**1.49**	**1.56**	**0.11**	**1.52**	**1.62**	**0.11**

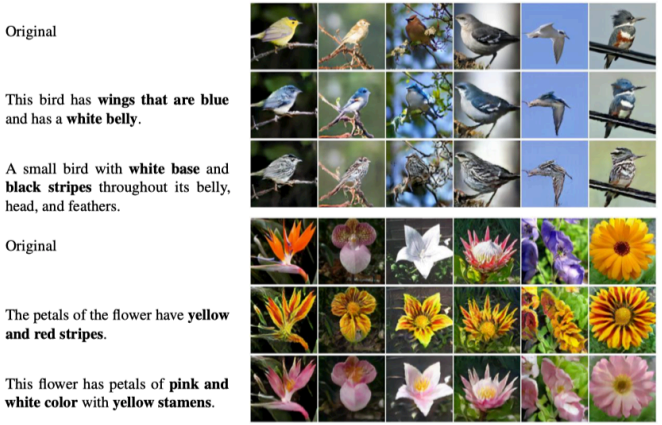


Figure 3: Qualitative results of our method on CUB and Oxford-102 datasets.

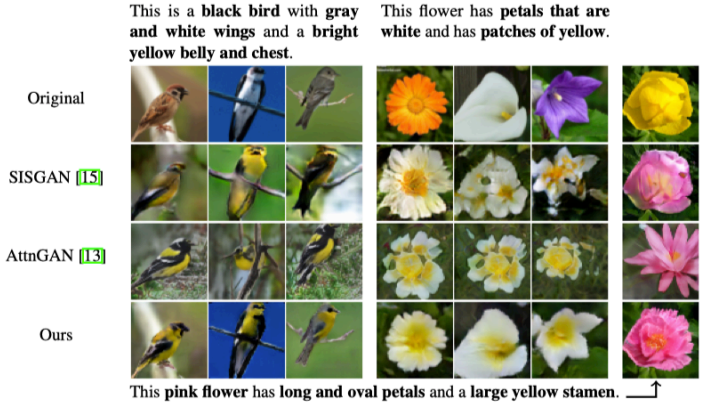


Figure 4: Qualitative comparison of three methods. In most cases, our method outperforms baseline methods qualitatively. The rightmost column shows a failure case using our method.

Figure 7: Sentence interpolation results. Our generator smoothly generates new visual attributes without loosing original image.

- Text Guide Image Manipulation (TGIM)
- Text-to-Image Synthesis (T2I)

	TGIM	T2I	Negative text	Text + Image
1 - Style Encoder	O	X	X	Concat
1 - GAN-CLS, INT	X	O	Random	Concat
2 - StackGAN,++	X	O	Random	Concat
3 - SISGAN	O	X	Random	Concat
4 - AttnGAN	X	O	Random (constrastive)	Attention, Concat
5 - TaGAN	O	X	Random	Attention, Concat

Text Guided Image Manipulation
Text to Image Synthesis

Natural Language Processing
& Artificial Intelligence

- Word & Sentence level 같이 활용
- Various Loss (DAMSM, Reconstruction)
- word-region, word-layer wise attention

-> Multimodal에 대한 alignment를 좀 더 고도화했음
-> 하지만 여전히 negative sample에 대해 Random
-> Text embedding을 concat으로 넘겨줌

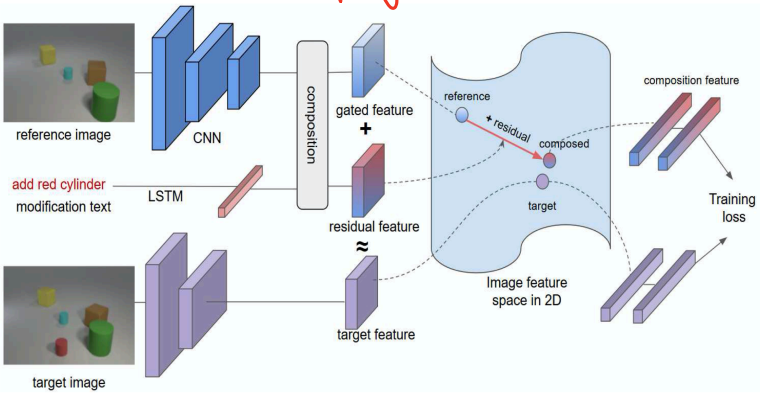
- 1) Generative adversarial text to image synthesis (ICML, 2016)
- 2) StackGAN (ICCV, 2017)
- 3) StackGAN++ (TPAMI, 2017)
- 4) Semantic Image Synthesis via Adversarial Learning (ICCV, 2017)
- 5) AttnGAN (CVPR, 2018)
- 6) TaGAN (NIPS, 2018)

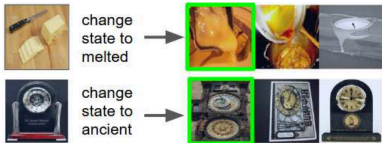
고려대학교 컴퓨터학과 석사과정
2019011001 박성진

Index

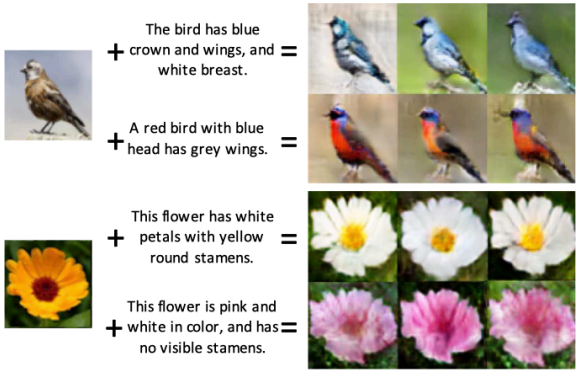
- What we are considering deeply…

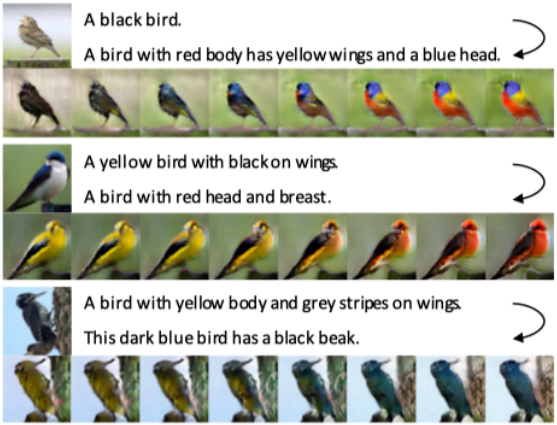
target text





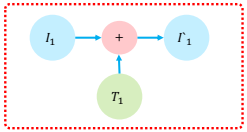
sentence, caption



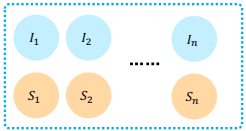


- What we are considering deeply…

  - Unpaired dataset



Task specific dataset for supervision
(I, T, I`)



Common image, caption pairs
(I,S)

- I: Source Image
- S: Source Text (caption)
- (I,S): Normal pairs
- T: Target Text
- I`: Target Image

•(I,T) -> I`; What we want to Learn

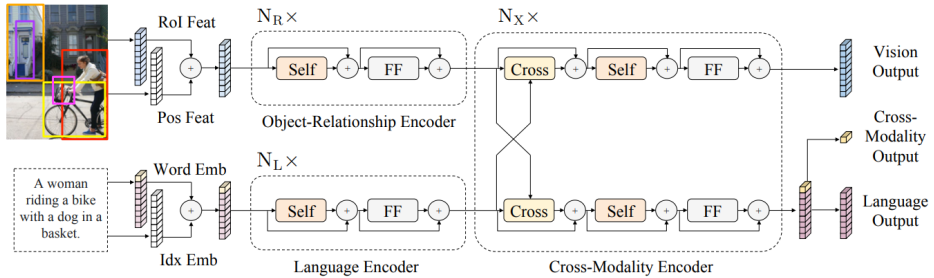
***T is not negative text**

  - Common space
    - Text space -> Image space
    - Image space -> Text space
    - Visual-linguistic space
    - Joint space

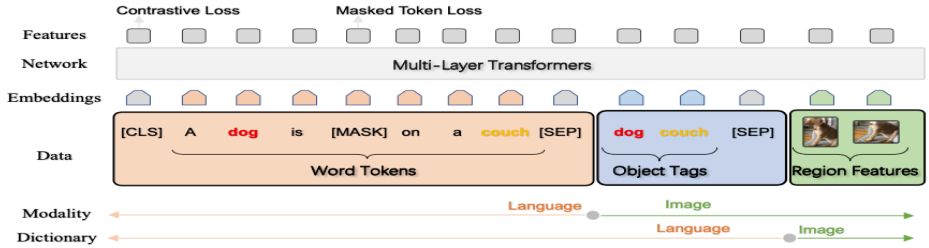
- Unpaired dataset -> Unsupervised Learning, Self-supervised Learning

Representation learning

- Vision
  - Pretext task, Contrastive learning
  - MoCo, SimCLR, ···, BYOL, SwAV, ···
- NLP
  - Pretraining Model, Objective
  - Bert, GPT, ··· , ELECTRA, T5,···
- Visual-linguistic representation
  - Two-Stream: Lxmert, ViLBERT,···
  - One-Stream: Uniter, Univer-VL, VL-BERT, Oscar, ···



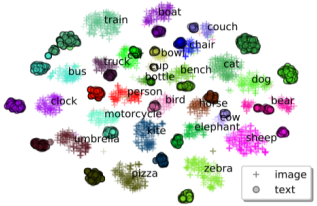
- Two-stream: Lxmert (2019 EMNLP)



- One-stream: Oscar (2020 ECCV)

- Unpaired dataset -> Unsupervised Learning, Self-supervised Learning

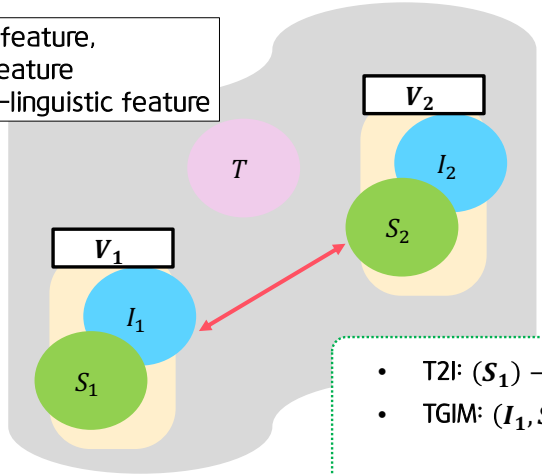
Representation learning

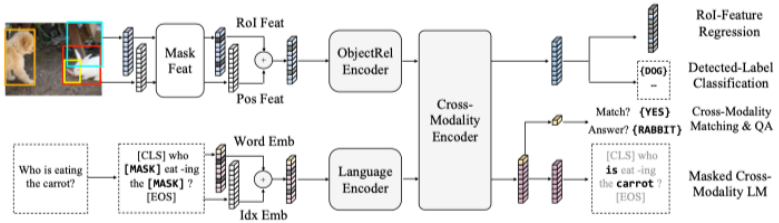


(a) OSCAR

- Visual-linguistic space
- Image-Text Semantic Alignment

I = image feature,
S = text feature
V = visual-linguistic feature





- T2I: $(S_1) \rightarrow \hat{I}$
- TGIM: $(I_1, S_2) \rightarrow \hat{I}$

- $V_1 - V_2$ *or* $V_2 - V_1 \cong T$
- $S_1 - S_2$ *or* $S_2 - S_1 \cong I_1 - I_2$ *or* $I_2 - I_1$
  ; text semantic gap ≅ Image semantic gap
- $(I_1, S_1) - (I_1, S_2) \cong T$
  Text에서의 Interpolation

Q & A