

caption2image surveys

東京大学理学部情報科学科4年
古賀樹

目次

- はじめに
- 論文要約
 - Generative Adversarial Image to Text Synthesis
 - StackGAN
 - StackGAN++
 - AttnGAN
 - (おまけ) Show, Attend and Tell

はじめに

- ・ このサーベイは東大理情4年次の「情報科学演習III」で「宮尾研究室」に配属され、行ったものをまとめたものです。
- ・ キャプションによる画像生成を行なっている論文で、有名かつ自分の興味に沿うもののみをサーベイしています。（ある程度網羅性があるかとは思いますが）

論文要約

Generative Adversarial Image to Text Synthesis

論文要約 / 書誌情報

Generative Adversarial Text to Image Synthesis

- 著者
 - Reed et al.
 - ICLR 2016

論文要約 / 提案内容

- キャプションから画像を生成するGenerative Adversarial Network (GAN) を提案
 - ネットワーク構造
 - 学習手法
- Caltech-UCSD Birds (CUB), Oxford-102 Flowers(, MS COCO)で検証

論文要約 / 先行研究

- conditional GAN
labelによる条件付けをして画像を生成
- キャプションをエンコードして、類似度の高い画像を検索
- Variational (recurrent) autoencoder (VAE)によるキャプションを用いた画像生成

論文要約 / 手法 / (Background) GAN

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

- Goodfellow et al. により2014年に提案

- D (Discriminator): 画像の真偽を判定

G (Generator): 画像を生成

論文要約 / 手法 / (Background) char-CNN-RNN

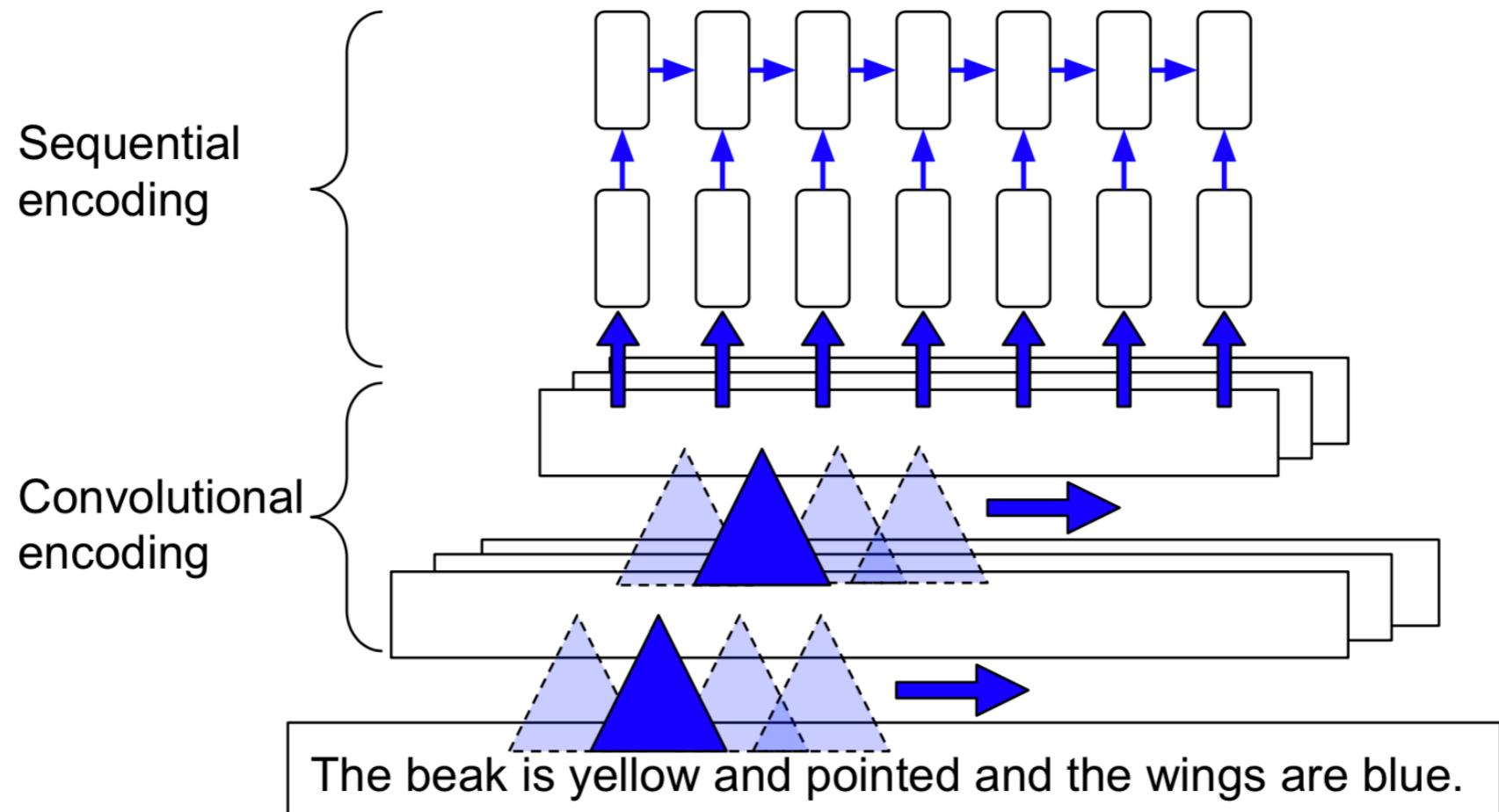
$$\frac{1}{N} \sum_{n=1}^N \Delta(y_n, f_v(v_n)) + \Delta(y_n, f_t(t_n))$$

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

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

- $\phi(v)$: 画像のエンコード
- $\varphi(t)$: テキストのエンコード
- 今回はテキストのエンコードを用いる

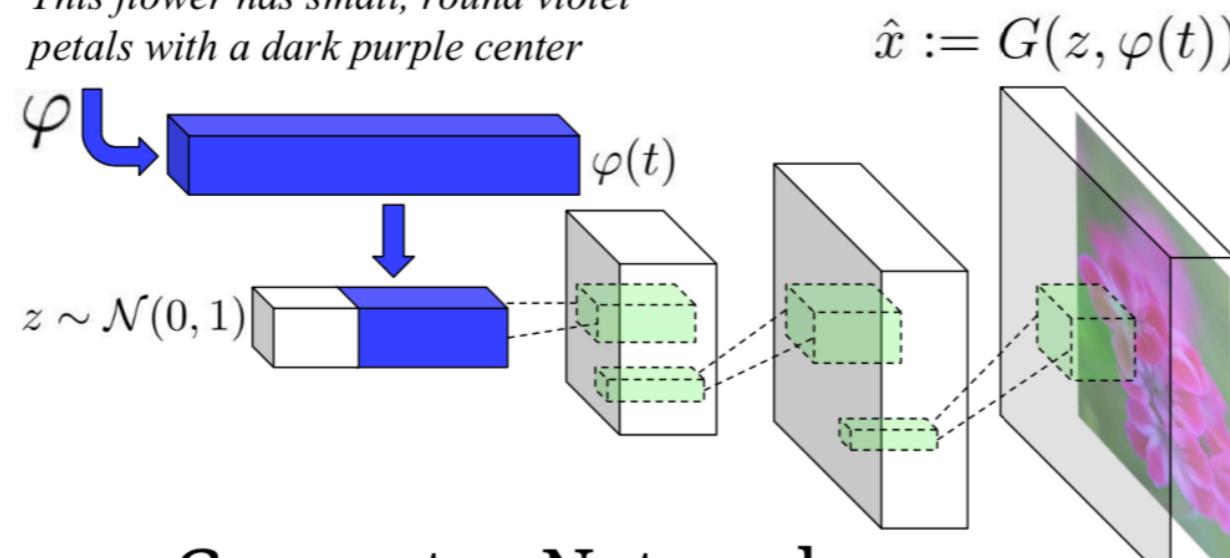
論文要約 / 手法 / (Background) char-CNN-RNN



- $(w \times h \times c) \rightarrow (1 \times (\text{sentence length}) \times (\text{alphabet size}))$
- CNNの隠れ層を時間軸方向に分割する \rightarrow RNN
- $\varphi(t)$ は RNN の全ての隠れ層の平均

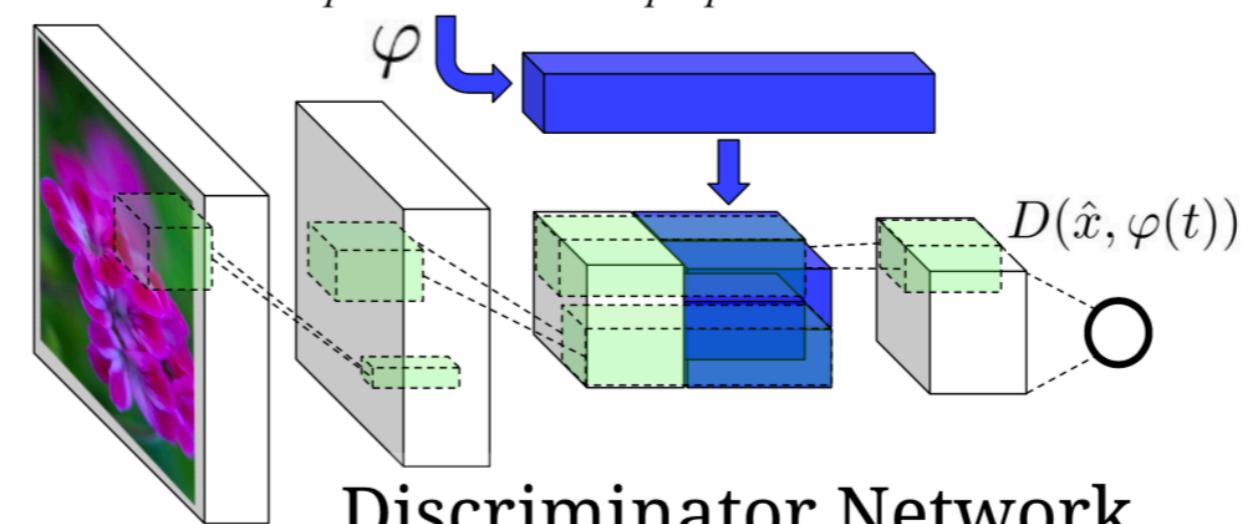
論文要約 / 手法 / Network

This flower has small, round violet petals with a dark purple center



Generator Network

This flower has small, round violet petals with a dark purple center



Discriminator Network

論文要約 / 手法 / Algorithm

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

- 1: **Input:** minibatch images x , matching text t , mis-matching \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**
-

論文要約 / 手法 / Algorithm / Matching Aware D

- キャプションと画像のマッチングが正しいかどうかを判定する必要あり
- 対応関係がない画像とキャプションの組みは偽物と判断するためのロス関数を用意

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

- 1: **Input:** minibatch images x , matching text t , mis-matching \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**

論文要約 / 手法 / Algorithm / Manifold interpolation

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

- Generatorのロス関数に上の項を追加
- $\beta = 0.5$ として異なるキャプションを作り、多様性のあるGeneratorを学習させる

論文要約 / 実験

- データ: CUB (birds), Oxford-102 (flowers), COCO
- encoder (char-CNN-RNN) は事前学習済み
- Optimizer: Adam
- 評価: 生成画像の観察

論文要約 / 結果



Figure 3. Zero-shot (i.e. conditioned on text from unseen test set categories) generated bird images using GAN, GAN-CLS, GAN-INT and GAN-INT-CLS. We found that interpolation regularizer was needed to reliably achieve visually-plausible results.



Figure 4. Zero-shot generated flower images using GAN, GAN-CLS, GAN-INT and GAN-INT-CLS. All variants generated plausible images. Although some shapes of test categories were not seen during training (e.g. columns 3 and 4), the color information is preserved.

論文要約 / まとめ

- キャプションから画像を生成するGenerative Adversarial Network (GAN) を提案
 - ネットワーク構造
 - 学習手法
- Caltech-UCSD Birds (CUB), Oxford-102 Flowers(, MS COCO)で検証

StackGAN

論文要約 / 書誌情報

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

- 著者
 - Zhang et al.
- ICCV 2017

論文要約 / 提案内容

- キャプションから画像 (256x256) を生成する

Generative Adversarial Network (GAN) を提案

- 学習の安定化と多様性のために Conditioning

Augmentation (CA) を提案

- 定性・定量評価

論文要約 / 先行研究

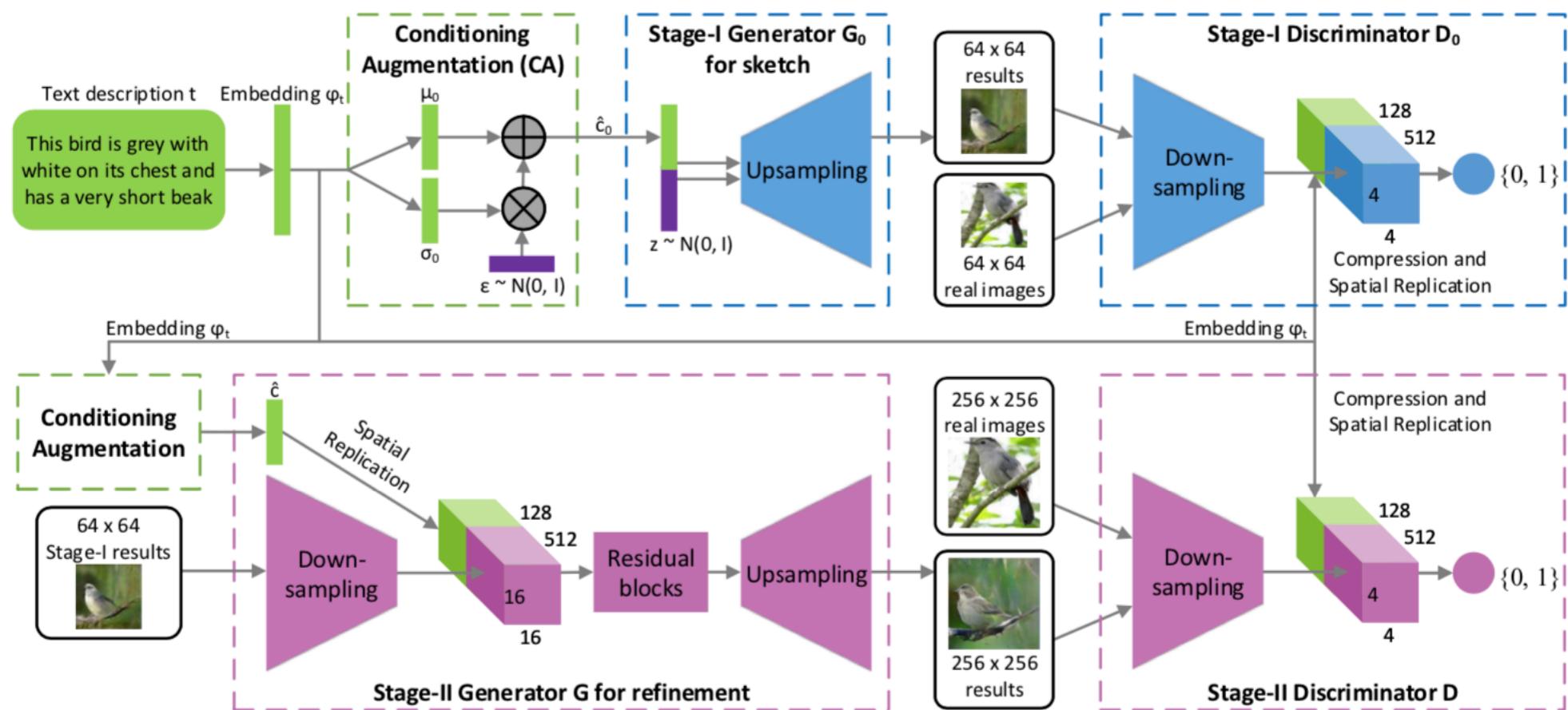
- Generative Adversarial Text to Image Synthesis
 - 64x64の生成しかできないことが問題点

論文要約 / 手法

- 2-stage GAN

1st: noise + text -> 64x64

2nd: 1st image + text -> 256x256



論文要約 / 手法 / Conditioning Augmentation

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

- text embeddingの際に潜在空間が非連続になることを防ぐためにAugmentationを行う
- \hat{c} を $\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))$ からサンプリング
- reparameterization trick により逆伝播を可能に

$$\hat{c}_0 = \mu_0 + \sigma_0 \odot \epsilon \quad \epsilon \sim \mathcal{N}(0, I)$$

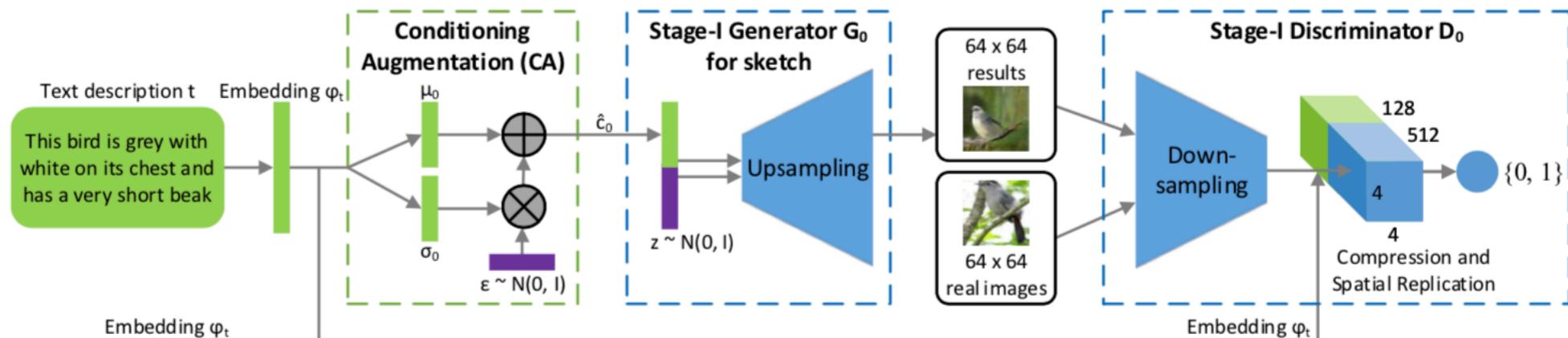
- 上の項で正則化

論文要約 / 手法 / Stage-I GAN

- 全体の形や色などを捉えるように生成
- embeddingは学習済みのものを用いる (c.f. GAN-INT-CLS)

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

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

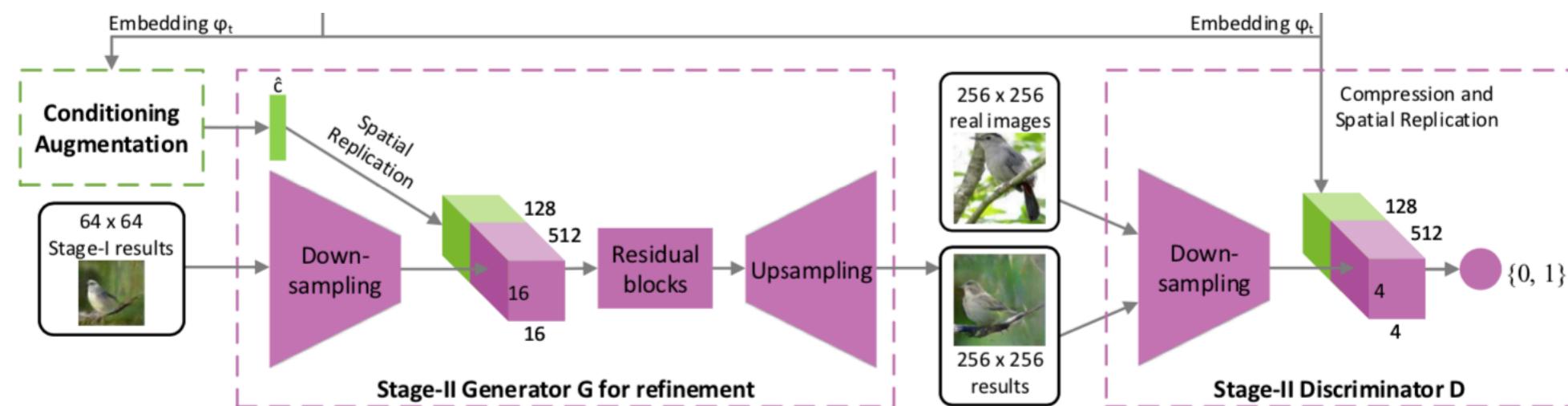


論文要約 / 手法 / Stage-II GAN

- Stage-I GANの画像の歪みや詳細の欠乏を補い、高解像度な画像を生成

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

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



論文要約 / 実験

- データ: CUB (birds), Oxford-102 (flowers), COCO
- Optimizer: Adam
- 評価: Inception Score & 人による評価
- GAN-INT-CLS と GAWWN と比較

論文要約 / 実験 / Inception Score

$$I = \exp(\mathbb{E}_{\mathbf{x}} D_{KL}(p(y|\mathbf{x}) \parallel p(y))),$$

- y は Inception model の予測クラス (入力は生成画像)
- 大きい方が良い
 - $p(y|x)$ の分布は特定のクラスで大きく、他は小さくあってほしい (meaningful)
 - $p(y)$ の分布は平坦であってほしい (diverse)
 - 二つの分布の KL divergence は大きい方が良い
- キャプションと画像の対応が取れているかはわからない

論文要約 / 結果 / 生成画像



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

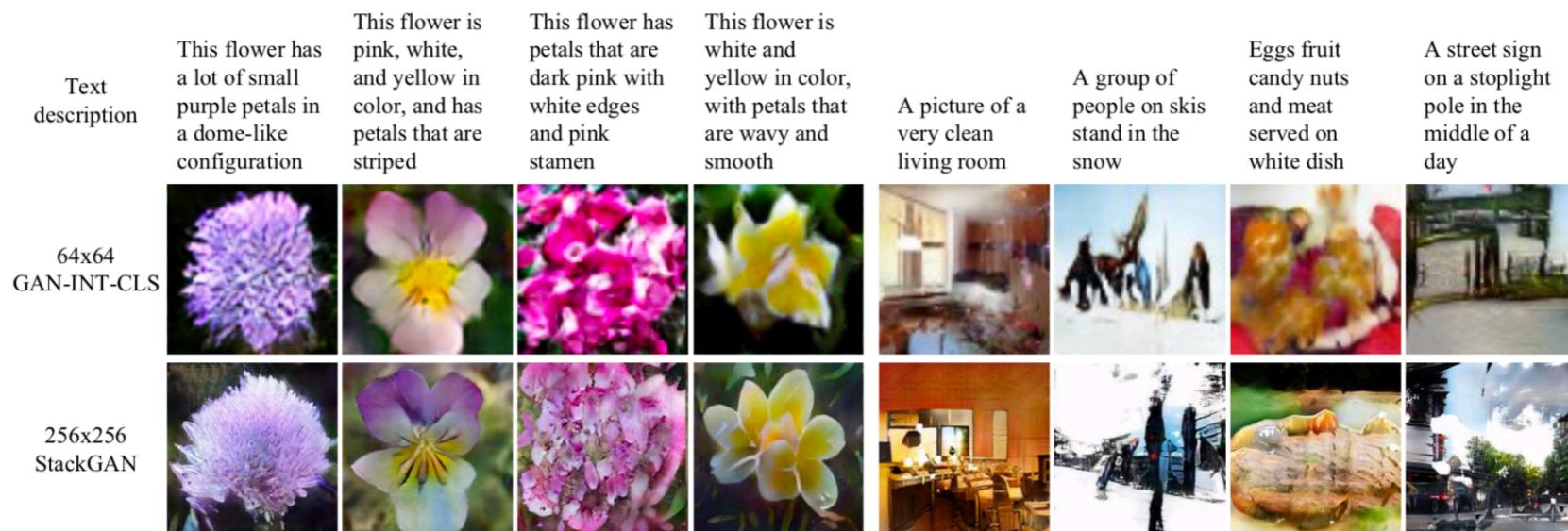


Figure 4. Example results by our StackGAN and GAN-INT-CLS [26] conditioned on text descriptions from Oxford-102 test set (leftmost four columns) and COCO validation set (rightmost four columns).

論文要約 / 結果

Metric	Dataset	GAN-INT-CLS	GAWWN	Our StackGAN
Inception score	CUB	2.88 ± .04	3.62 ± .07	3.70 ± .04
	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

論文要約 / まとめ

- キャプションから画像 (256x256) を生成する

Generative Adversarial Network (GAN) を提案

- 学習の安定化と多様性のために Conditioning

Augmentation (CA) を提案

- 定性・定量評価

StackGAN++

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

- 著者
 - Zhang et al.
- ICCV 2017

論文要約 / 提案内容

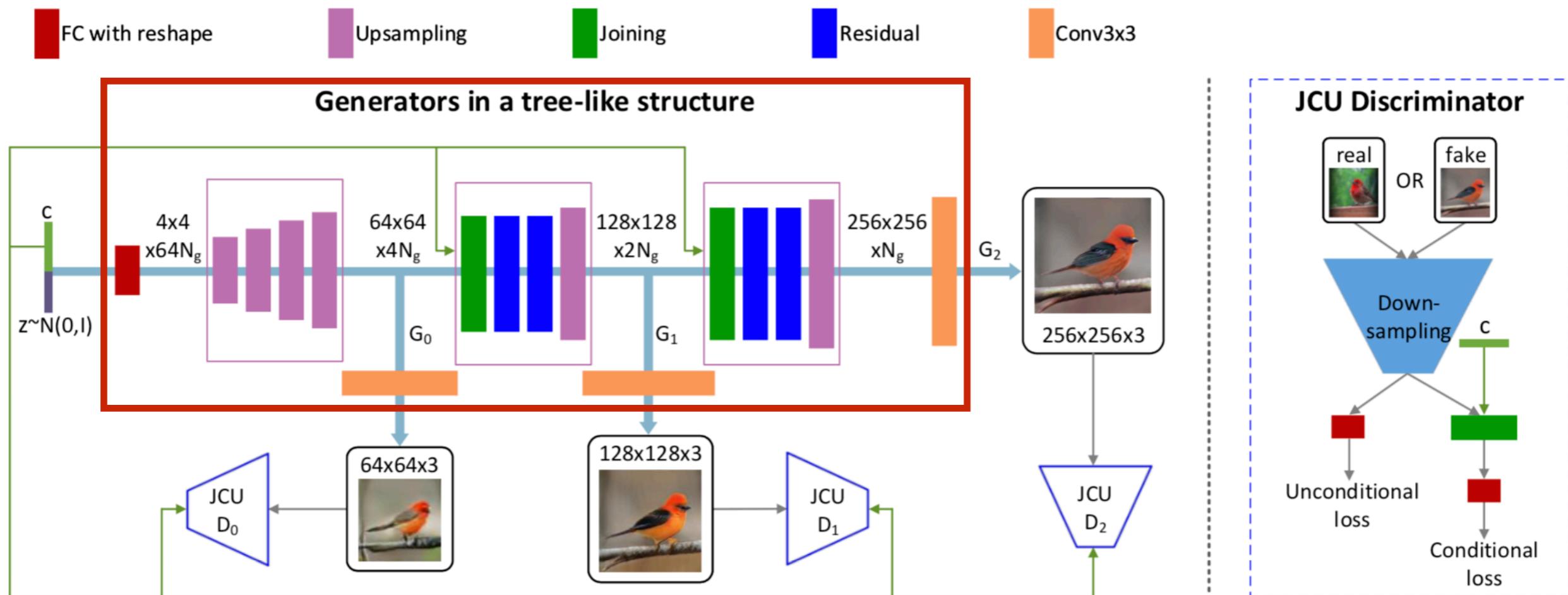
- StackGAN に加えて...
- 複数の解像度の画像の分布を同時に学習
 - 生成画像のクオリティの向上
 - 学習の安定化

論文要約 / 先行研究

- StackGANs, LAPGANs, Progressive GANs
 - 解像度を徐々に上げて 大枠 -> 詳細 と学習

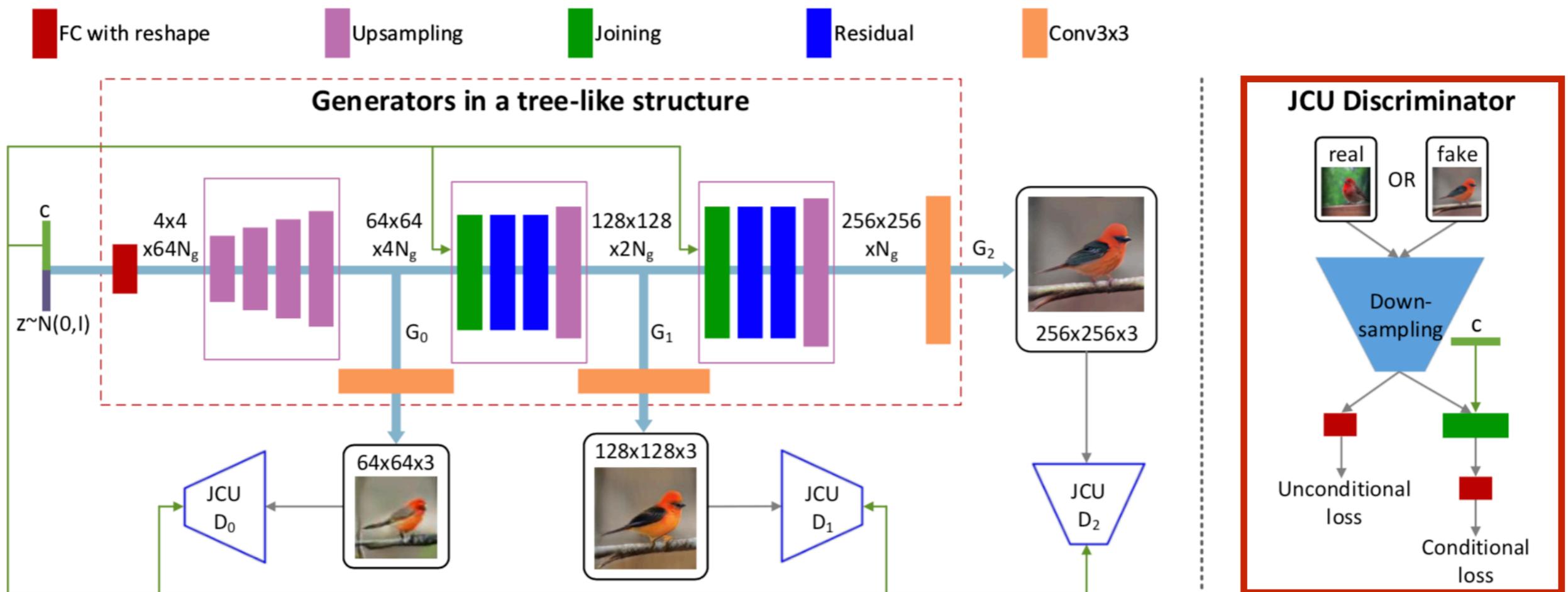
論文要約 / 手法 / Generator

- Multi-scale image distributions approximation
 - 1st: noise (+ text)
 - nth: hidden from (n-1)th (+ text)
- 手法 자체는必ずしもtextを必要としない



論文要約 / 手法 / Discriminator

- Joint conditional and unconditional distribution approximation
 - conditional / unconditional を分離して分布を学習



論文要約 / 手法 / 損失関数

- Discriminators

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

- Generator

$$\mathcal{L}_G = \sum_{i=1}^m \mathcal{L}_{G_i}, \quad \mathcal{L}_{G_i} = \underbrace{-\mathbb{E}_{s_i \sim p_{G_i}} [\log D_i(s_i)]}_{\text{unconditional loss}} + \underbrace{-\mathbb{E}_{s_i \sim p_{G_i}} [\log D_i(s_i, c)]}_{\text{conditional loss}}.\tag{12}$$

$$s_i = G_i(h_i), \quad i = 0, 1, \dots, m-1, \quad h_0 = F_0(z); \quad h_i = F_i(h_{i-1}, z), \quad i = 1, 2, \dots, m-1,$$

論文要約 / 手法 / Color-consistency regularization

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

$$x_k = (R, G, B)^T \quad \mu = \sum_k x_k / N \quad \Sigma = \sum_k (x_k - \mu)(x_k - \mu)^T / N$$

- 複数のGeneratorが基本構造、色などを保存するための正則化
- unconditional taskでは効果があるが、conditional taskでは効果なし
 - image-text の制約が強い

論文要約 / 実験

- データ
 - Conditional
 - CUB (birds), Oxford-102 (flowers), COCO
 - Unconditional
 - LSUN-bedroom, LSUN-church, ImageNet-dog, ImageNet-cat
- Optimizer: Adam
- 評価: Frechet inception distance, Inception Score, 人による評価
- StackGANと比較

論文要約 / 実験 / Frechet inception distance

$$FID = |\mu_1 - \mu_2|^2 + \text{tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1 \Sigma_2)^{1/2})$$

$$\mu = \frac{1}{|A|} \sum_{h \in H} h \quad \Sigma = \frac{1}{|A|-1} \sum_{h \in H} (h - \mu)(h - \mu)^T$$

- Inception modelの最後のプーリング層をデータセット、生成画像群それぞれから出力(h)
 h が多変量正規分布に従うと仮定し、二つのデータ間の Frechet distanceを計算する
- 小さい方が良い

論文要約 / 結果 / 生成画像

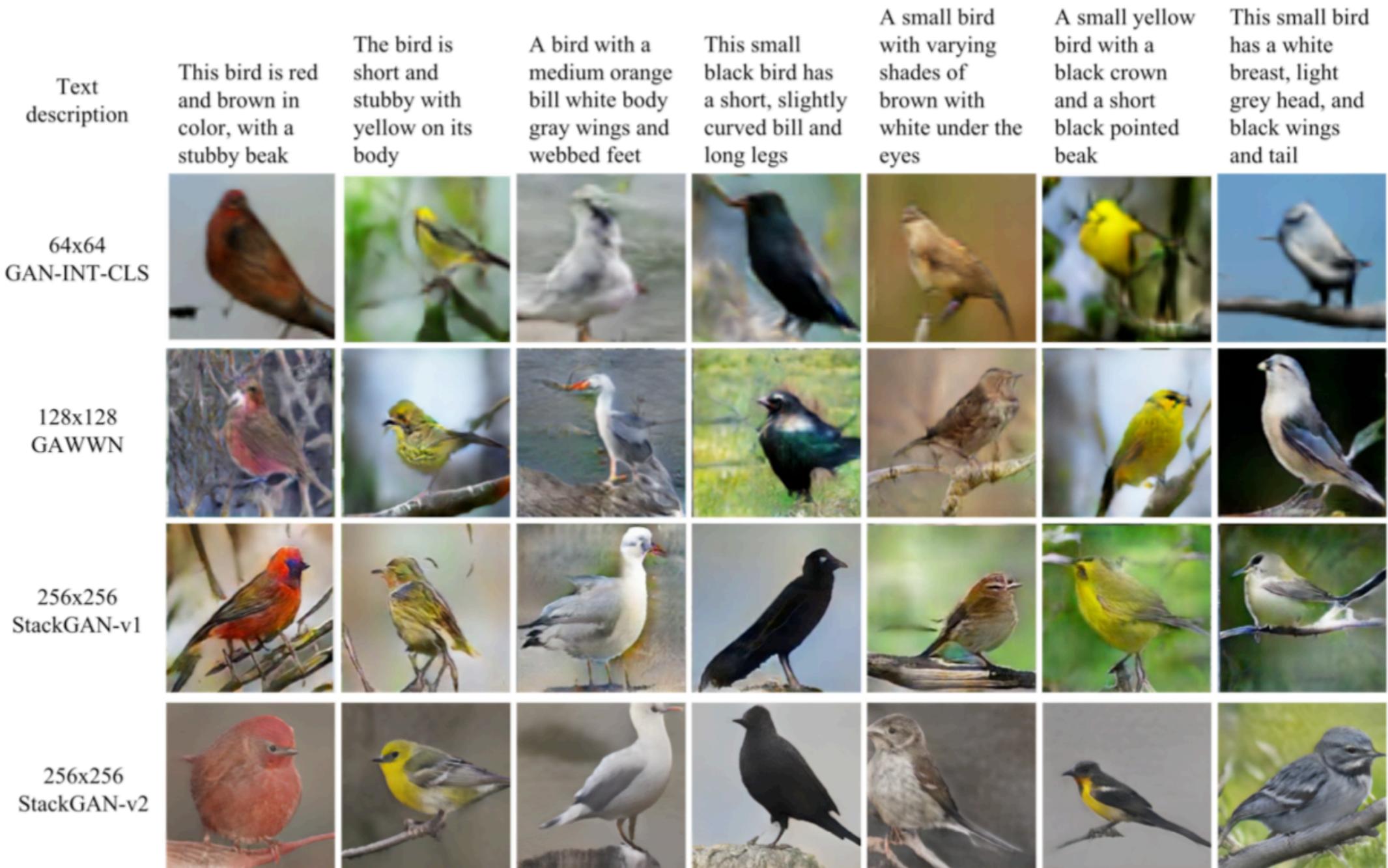


Fig. 3: Example results by our StackGANs, GAWWN [33], and GAN-INT-CLS [35] conditioned on text descriptions from CUB test set.

論文要約 / 結果 / 生成画像

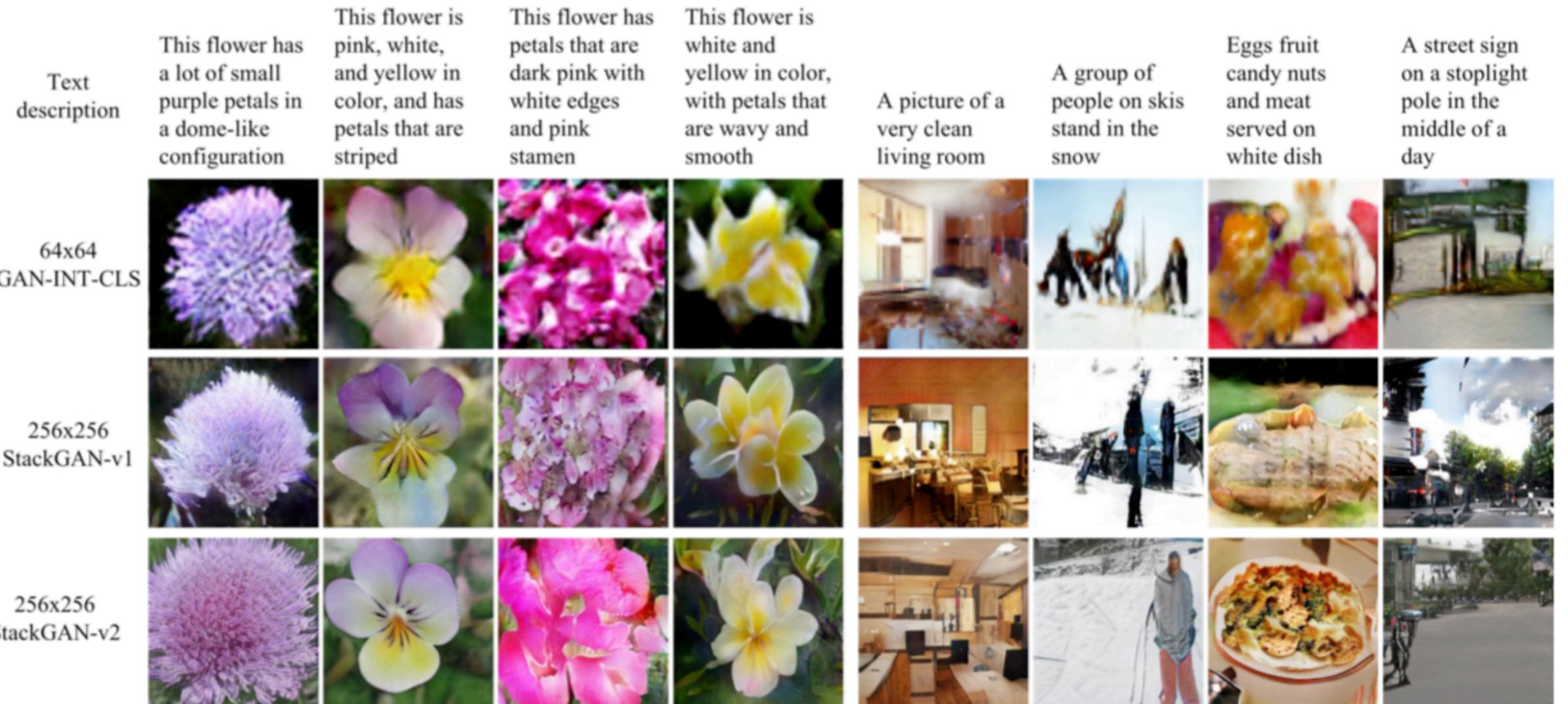
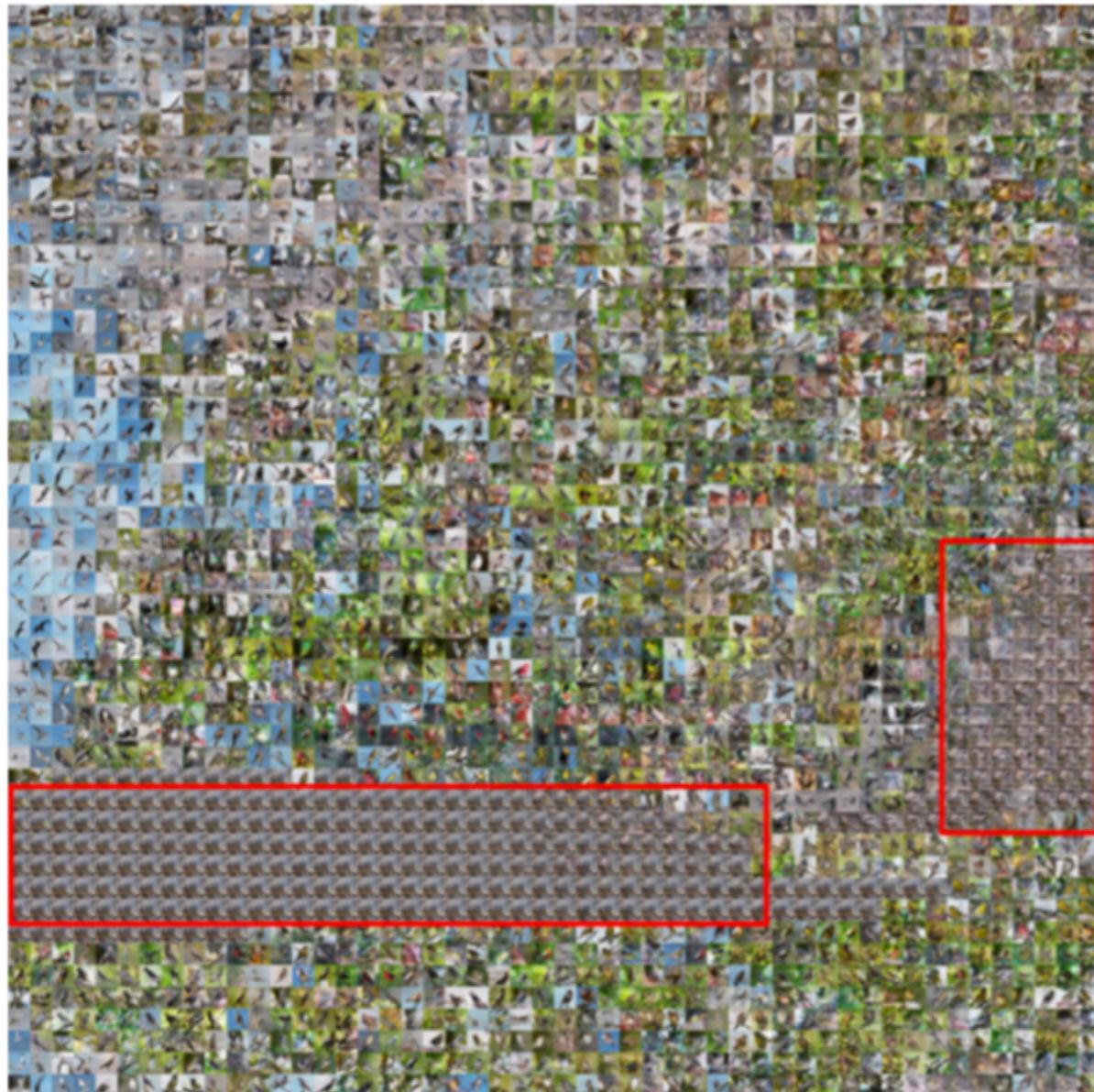
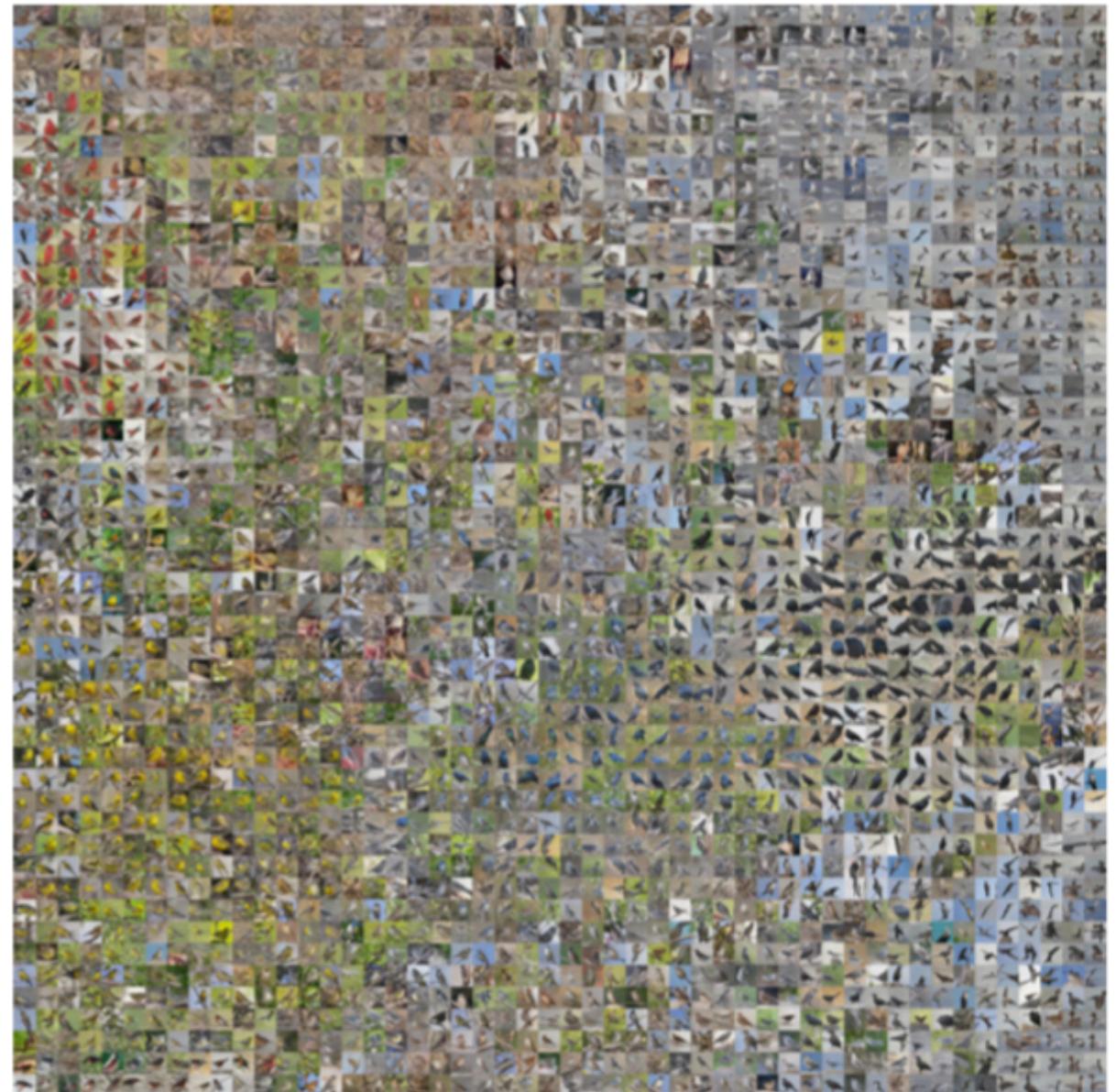


Fig. 4: Example results by our StackGANs and GAN-INT-CLS [35] conditioned on text descriptions from Oxford-102 test set (leftmost four columns) and COCO validation set (rightmost four columns).

論文要約 / 結果 / 生成画像 / mode collapse



(a) StackGAN-v1 has two collapsed modes (in red rectangles). (b) StackGAN-v2 contains no collapsed nonsensical mode.



論文要約 / 結果

Dataset		CUB	Oxford-102	COCO	LSUN-bedroom	LSUN-church	ImageNet-dog	ImageNet-cat
FID ↓	StackGAN-v1	51.89	55.28	74.05	91.94	57.20	89.21	58.73
	StackGAN-v2	15.30	48.68	81.59	35.61	25.36	44.54	28.59
IS ↑	StackGAN-v1	3.70 ± .04	3.20 ± .01	8.45 ± .03	3.59 ± .05	2.87 ± .05	8.84 ± .08	4.77 ± .06
	StackGAN-v2	4.04 ± .05	3.26 ± .01	8.30 ± .10	3.02 ± .04	2.38 ± .03	9.55 ± .11	4.23 ± .05
HR ↓	StackGAN-v1	1.81 ± .02	1.70 ± .03	1.45 ± .04	1.95 ± .01	1.86 ± .02	1.90 ± .01	1.88 ± .02
	StackGAN-v2	1.19 ± .02	1.30 ± .03	1.55 ± .05	1.05 ± .01	1.14 ± .02	1.10 ± .01	1.12 ± .02

TABLE 3: Comparison of StackGAN-v1 and StackGAN-v2 on different datasets by inception scores (IS), fréchet inception distance (FID) and average human ranks (HR).

論文要約 / まとめ

- StackGAN に加えて...
- 複数の解像度の画像の分布を同時に学習
 - 生成画像のクオリティの向上
 - 学習の安定化

AttnGAN

AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks

- 著者
 - Xu et al.
 - CVPR 2018

論文要約 / 提案内容

- Attentional Generative Adversarial Network による
キャプションからの画像生成
 - Attentional generative network
 - Deep Attentional Multimodal Similarity Model
(DAMSM)
- 定量的な性能評価
- Attentionの可視化

論文要約 / 先行研究

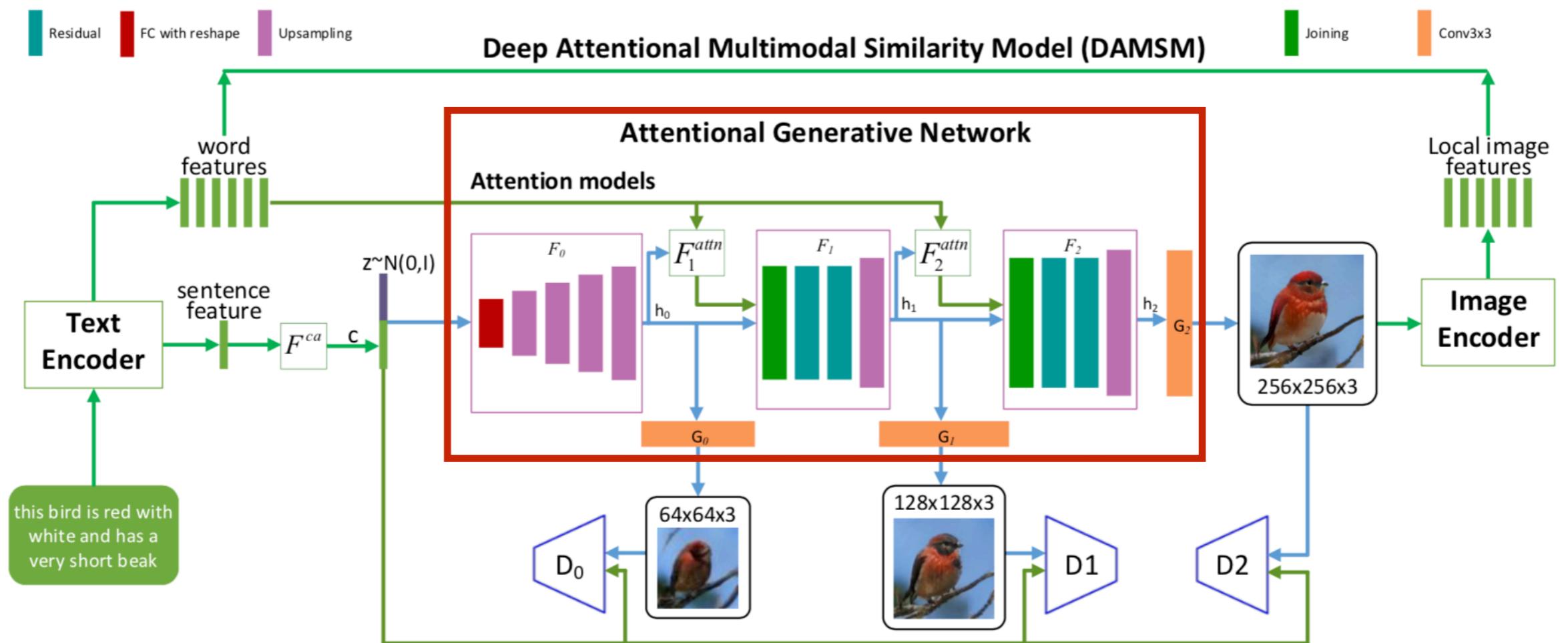
- StackGAN, StackGAN++
 - 単語レベルでの情報が画像に反映されない
 - COCOなどの複雑なデータセットで顕著

論文要約 / 手法 / Attentional Generative Network

- Attentional Generative Network

1st: noise + sentence feature w/ Conditioning Augmentation

nth: hidden from (n-1)th + word features w/ Attention Model



論文要約 / 手法 / Attentional Generative Network

- Attentional Generative Network

1st: noise + sentence feature w/ Conditioning Augmentation

nth: hidden from (n-1)th + word features w/ Attention Model

$$h_0 = F_0(z, F^{ca}(\bar{e}));$$

$$h_i = F_i(h_{i-1}, F_i^{attn}(e, h_{i-1})) \text{ for } i = 1, 2, \dots, m - 1;$$

$$\hat{x}_i = G_i(h_i).$$

(1)

論文要約 / 手法 / Attentional Generative Network

- Attention Model

word feature & image feature から 画像の各部分に対するそれぞれの語による重み付けを計算

$$F^{attn}(e, h) = (c_0, c_1, \dots, c_{N-1}) \in \mathbb{R}^{\hat{D} \times \bar{N}}$$

$$e \in \mathbb{R}^{D \times T} \quad e' = Ue \quad U \in \mathbb{R}^{\hat{D} \times D}$$

$$h \in \mathbb{R}^{\hat{D} \times N}$$

$$c_j = \sum_{i=0}^{T-1} \beta_{j,i} e'_i, \quad \text{where } \beta_{j,i} = \frac{\exp(s'_{j,i})}{\sum_{k=0}^{T-1} \exp(s'_{j,k})}, \quad s'_{j,i} = h_j^T e'_i,$$

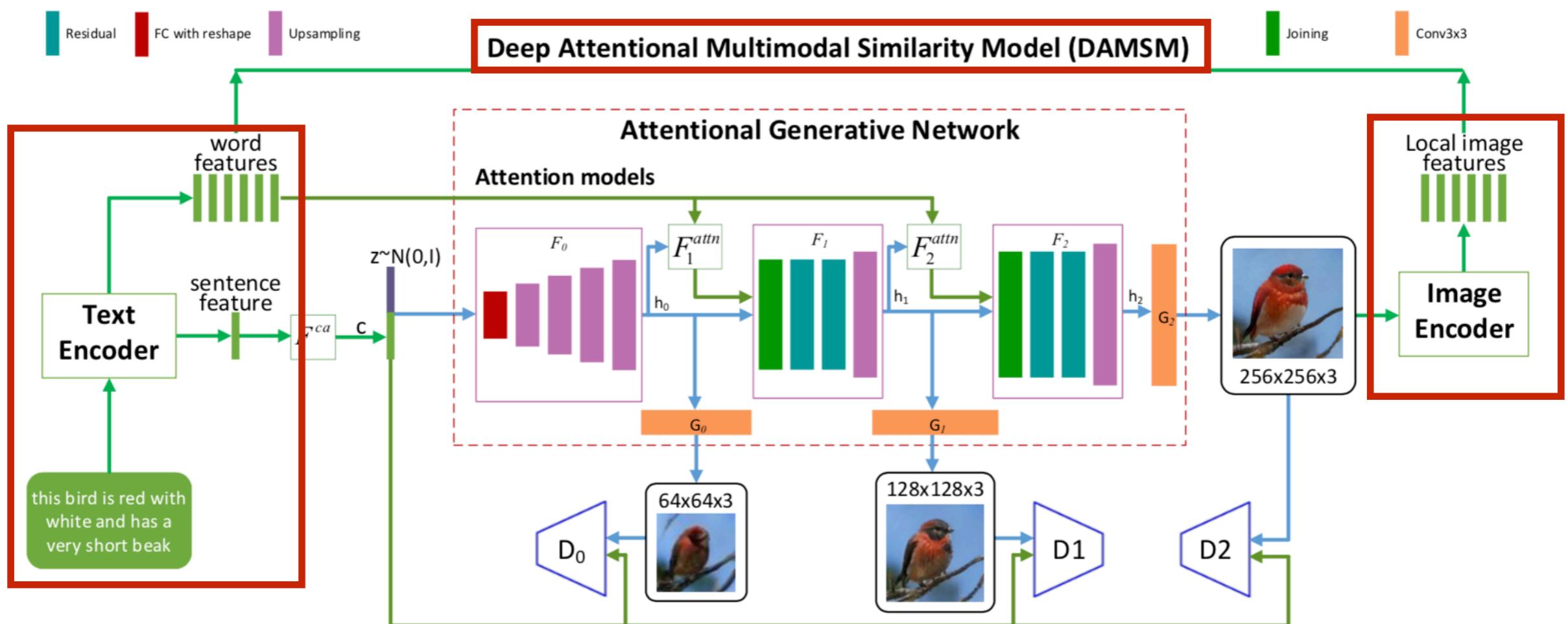
論文要約 / 手法 / DAMSM

- Deep Attentional Multimodal Similarity Model

画像の部分と単語を共通の意味空間に写像

画像と文章の類似度の（単語レベルでの）計算を可能に

損失関数に組み込む



論文要約 / 手法 / DAMSM / Encoders

- Text encoder
 - Bi-directional LSTM
 - Words embedding: 各単語に対して双方向の隠れ状態
 - Sentence embedding: 双方向の最後の隠れ状態
- Image Encoder
 - Inception-v3 (pretrained on ImageNet)
 - Local features: “mixed_6e” layer
 - 289 ($=17^2$)個のsub-region
 - Global features: 最後のaverage pooling layer
 - それぞれ一層のパーセプトロンを通してTextと同一の空間に写像

論文要約 / 手法 / DAMSM / image-text matching score

- 画像と文章の類似度をattentionベースでスコア化

$$s = e^T v,$$

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

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

$$R(c_i, e_i) = (\bar{c}_i^T e_i) / (||c_i|| ||e_i||).$$

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

論文要約 / 手法 / DAMSM / image-text matching score

- 画像と文章の類似度をattentionベースでスコア化

$$s = e^T v,$$

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

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

$$R(c_i, e_i) = (\bar{c}_i^T e_i) / (||c_i|| ||e_i||).$$

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

- 単語と画像の部分の組み合
わせの類似度行列を計算
 - e: word features
 - v: image features

論文要約 / 手法 / DAMSM / image-text matching score

- 画像と文章の類似度をattentionベースでスコア化

$$s = e^T v,$$

- 類似度行列を正規化

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

(単語方向)

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

$$R(c_i, e_i) = (\bar{c}_i^T e_i) / (||c_i|| ||e_i||).$$

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

論文要約 / 手法 / DAMSM / image-text matching score

- 画像と文章の類似度をattentionベースでスコア化

$$s = e^T v,$$

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

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

- それぞれの語に対応する画像の（各部分を合算した）表現を計算
- γ_1 はattentionの程度を決めるパラメータ

$$R(c_i, e_i) = (\bar{c}_i^T e_i) / (||c_i|| ||e_i||).$$

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

論文要約 / 手法 / DAMSM / image-text matching score

- 画像と文章の類似度をattentionベースでスコア化

$$s = e^T v,$$

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

- 単語の埋め込み表現とその単語に対応する画像の表現の類似度（コサイン類似度）を計算

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

$$R(c_i, e_i) = (c_i^T e_i) / (\|c_i\| \|e_i\|).$$

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

論文要約 / 手法 / DAMSM / image-text matching score

- 画像と文章の類似度をattentionベースでスコア化

$$s = e^T v,$$

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

- 最終的なスコアを計算
- γ_2 はどの程度類似度が高いペアを重視するかのパラメータ

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

$$R(c_i, e_i) = (\bar{c}_i^T e_i) / (||c_i|| ||e_i||).$$

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

論文要約 / 手法 / DAMSM / 損失関数

- スコアを用いてロス関数を設計

$$P(D_i|Q_i) = \frac{\exp(\gamma_3 R(Q_i, D_i))}{\sum_{j=1}^M \exp(\gamma_3 R(Q_i, D_j))}, \quad \mathcal{L}_1^w = - \sum_{i=1}^M \log P(D_i|Q_i),$$

$$P(Q_i|D_i) = \frac{\exp(\gamma_3 R(Q_i, D_i))}{\sum_{j=1}^M \exp(\gamma_3 R(Q_j, D_i))} \quad \mathcal{L}_2^w = - \sum_{i=1}^M \log P(Q_i|D_i),$$

$R(Q, D) = (\bar{v}^T \bar{e}) / (\|\bar{v}\| \|\bar{e}\|)$ として L_1^s, L_2^s も計算

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

論文要約 / 手法 / 損失関数

- Discriminators (c.f. StackGAN++, JCU Discriminator)

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

- Generator

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

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

論文要約 / 実験

- データ
 - CUB (birds), COCO
- Optimizer: Adam
- 評価: Inception Score, R-precision, Attention可視化
- GAN-INT-CLS, GAWWN, StackGAN, StackGAN++, PPGNと比較

論文要約 / 実験 / R-precision

- 検索の精度検証などに用いられる
- クエリに対してR個の関連結果があるとする

検索によりR個関連すると思われる結果を取り出した時にr個実際に関連結果を取り出した

この時 $R\text{-precision} = r / R$ となる

- 今回はクエリが生成画像

$R=1$ (生成の際に用いた文章が唯一の関連結果)

DAMSMを用いて合計100個の文章と画像の類似度を計算し、R-precisionを計算

30000枚の画像で上を行う

論文要約 / 実験 / Attention可視化

$$\hat{\beta}_{j,i} = \begin{cases} \beta_{j,i}, & \text{if } \beta_{j,i} > 1/T, \\ 0, & \text{otherwise.} \end{cases}$$

- 上のように計算された $\hat{\beta}$ を upsample して画像と同じサイズとし、画像に重ね合わせる

論文要約 / 結果 / 生成画像

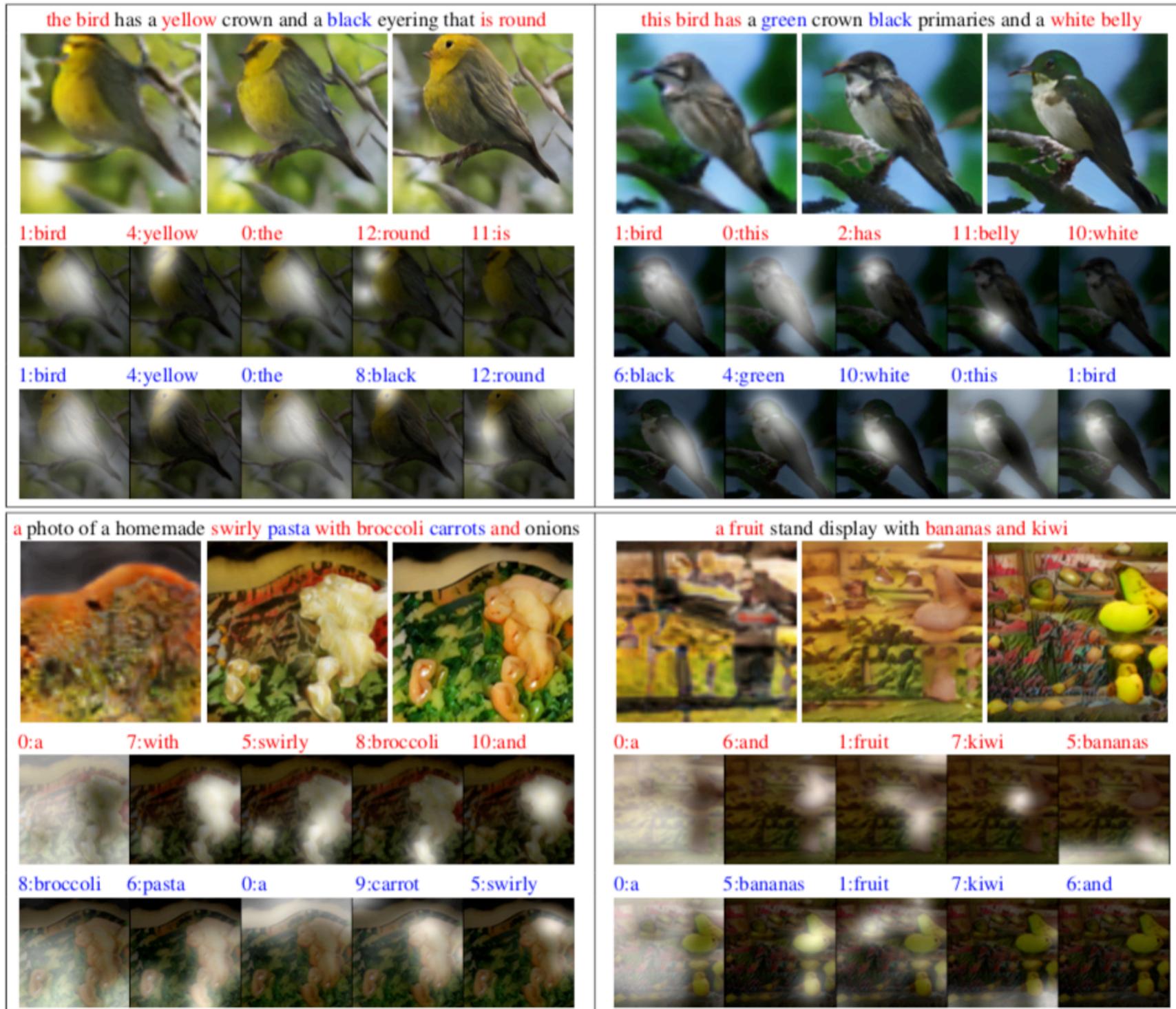


Figure 4. Intermediate results of our AttnGAN on CUB (top) and COCO (bottom) test sets. In each block, the first row gives 64×64 images by G_0 , 128×128 images by G_1 and 256×256 images by G_2 of the AttnGAN; the second and third row shows the top-5 most attended words by F_1^{attn} and F_2^{attn} of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

論文要約 / 結果 / 汎化性能



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.)

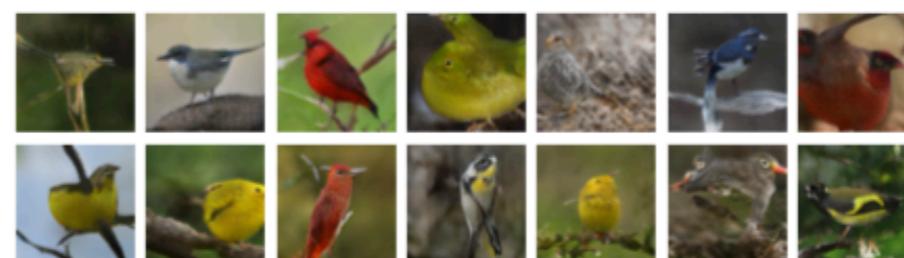


Figure 7. Novel images by our AttnGAN on the CUB test set.

論文要約 / 結果

Dataset	GAN-INT-CLS [20]	GAWWN [18]	StackGAN [31]	StackGAN-v2 [32]	PPGN [16]	Our AttnGAN
CUB	$2.88 \pm .04$	$3.62 \pm .07$	$3.70 \pm .04$	$3.82 \pm .06$	/	$4.36 \pm .03$
COCO	$7.88 \pm .07$	/	$8.45 \pm .03$	/	$9.58 \pm .21$	$25.89 \pm .47$

Table 3. Inception scores by state-of-the-art GAN models [20, 18, 31, 32, 16] and our AttnGAN on CUB and COCO test sets.

Method	inception score	R-precision(%)
AttnGAN1, no DAMSM	$3.98 \pm .04$	10.37 ± 5.88
AttnGAN1, $\lambda = 0.1$	$4.19 \pm .06$	16.55 ± 4.83
AttnGAN1, $\lambda = 1$	$4.35 \pm .05$	34.96 ± 4.02
AttnGAN1, $\lambda = 5$	$4.35 \pm .04$	58.65 ± 5.41
AttnGAN1, $\lambda = 10$	$4.29 \pm .05$	63.87 ± 4.85
AttnGAN2, $\lambda = 5$	$4.36 \pm .03$	67.82 ± 4.43
AttnGAN2, $\lambda = 50$ (COCO)	$25.89 \pm .47$	85.47 ± 3.69

Table 2. The best inception score and the corresponding R-precision rate of each AttnGAN model on CUB (top six rows) and COCO (the last row) test sets. More results in Figure 3.

論文要約 / まとめ

- Attentional Generative Adversarial Network による
キャプションからの画像生成
 - Attentional generative network
 - Deep Attentional Multimodal Similarity Model
(DAMSM)
- 定量的な性能評価
- Attentionの可視化

(おまけ) Show, Attend and Tell

論文要約 / 書誌情報

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

- 著者
 - Kelvin Xu et al.
 - 責任著者はYoshua Bengio
- ICML 2015

論文要約 / 提案内容

- attention ベースの画像のキャプション生成を行なった
 - “hard” & “soft” attentionを提案
 - attentionが「どこ」の「何」に注目したかを可視化
 - 提案手法をベンチマークデータセット(Flicker30k, Flicker8k, MS COCO)で検証 (SOTA)

論文要約 / 先行研究

- 画像 -> Encoder -> Decoder -> キャプションの形式（機械翻訳と同様の流れ）
- Encoder: CNN (全結合層を特徴ベクトルとして抽出)
Decoder: RNN
- 物体認識と組み合わせる手法もあった
(<- 提案手法ではexplicitに物体認識をしてはいない)

論文要約 / 手法

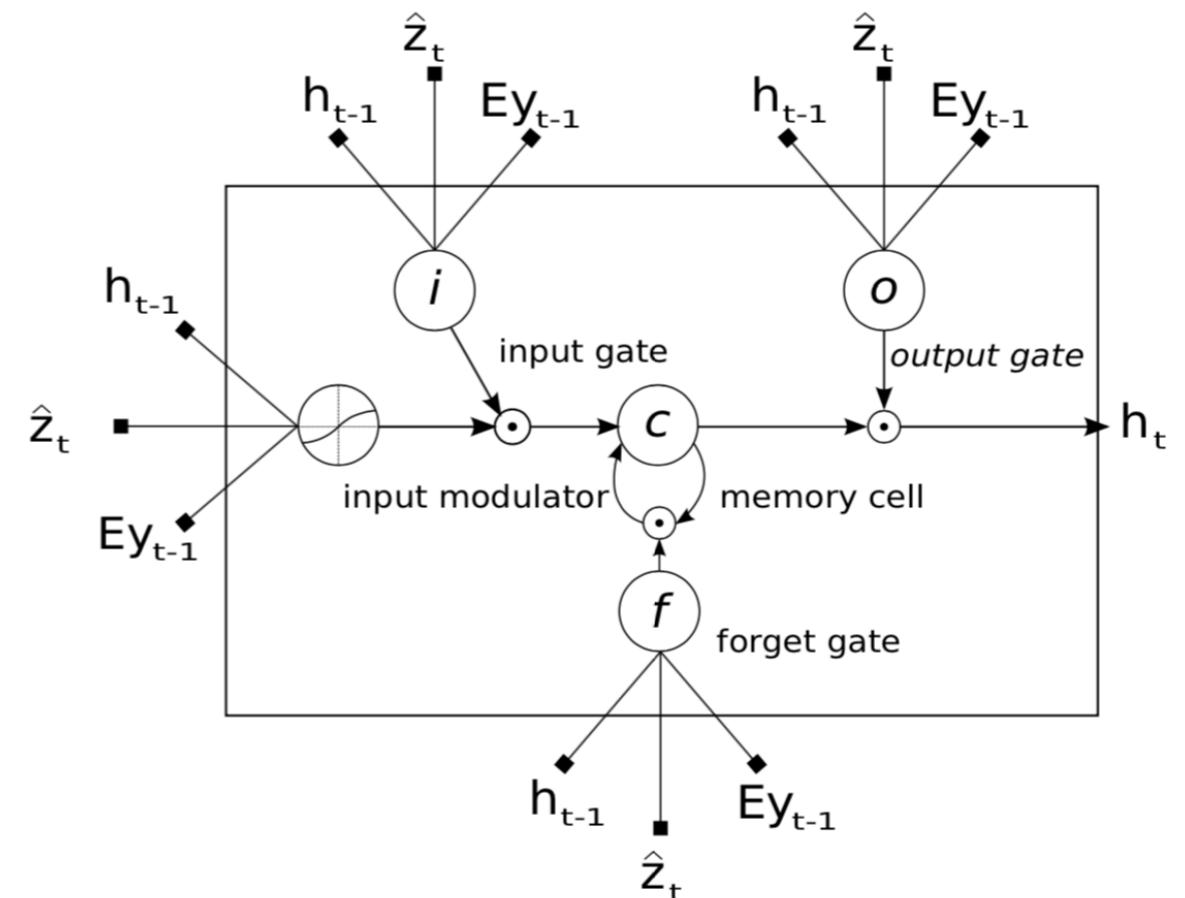
- 画像 -> Encoder -> Decoder -> キャプション
の形式を採用

論文要約 / 手法 / Encoder

- 入力: 画像
出力: "annotation vectors" $a = \{\mathbf{a}_1, \dots, \mathbf{a}_L\}$, $\mathbf{a}_i \in \mathbb{R}^D$
一つ一つが画像の一部分に対応
のCNN
- ネットワークの最後の方の畳み込み層 (x 全結合層)
を "annotation vectors" として抽出
- 画像の部分と "annotation vectors" の対応が取れる

論文要約 / 手法 / Decoder

- 1語ずつの生成をするLSTM (long short-term memory)
 - context vector \hat{z}_t
 - 隠れ状態 h_{t-1}
 - 生成された語 y_{t-1}
 - により条件付け
 - E : 埋め込み行列



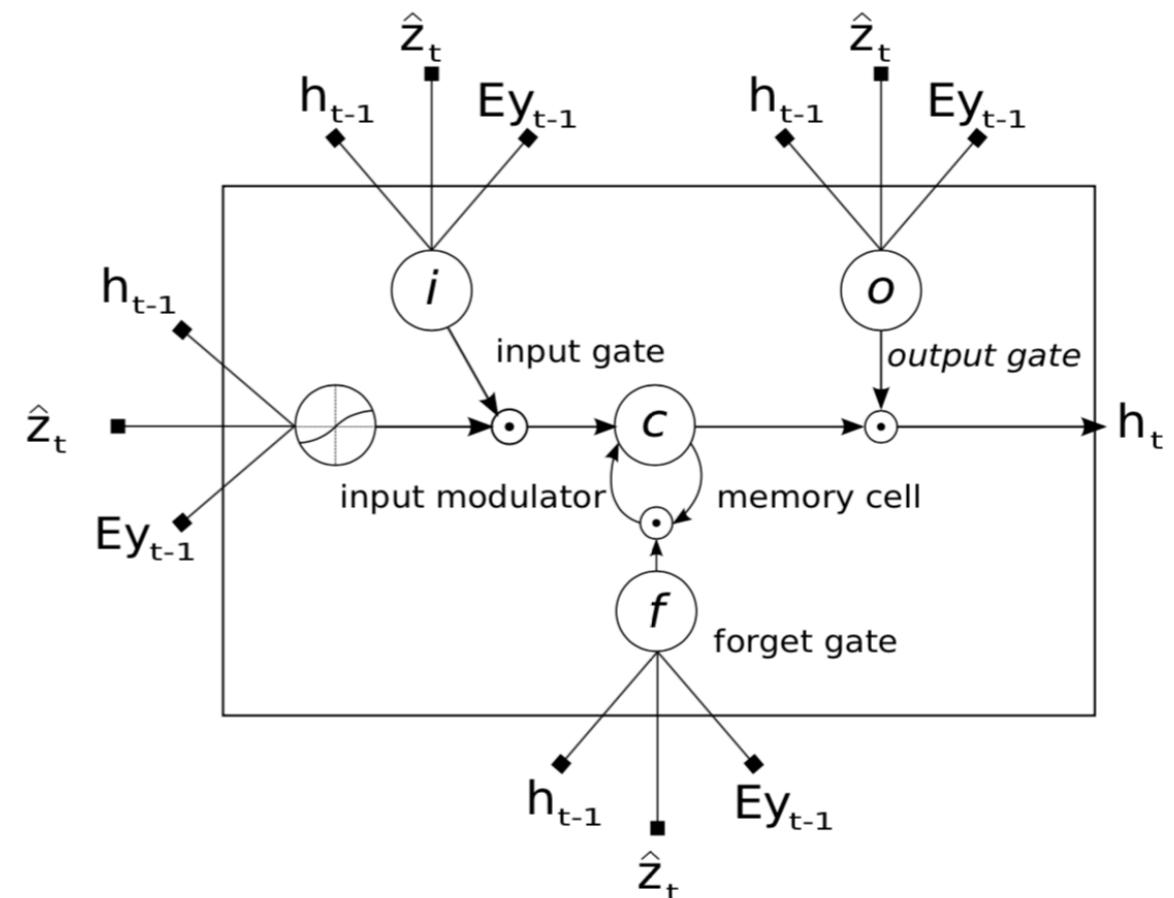
論文要約 / 手法 / Decoder

- input gate
入力への重み付け
- input modulator
LSTMのメモリへの入力の影響を調整
- forget gate
元々のメモリをどの程度忘却するか
- output gate
メモリの値を出力のために重み付け

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n, n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix} \quad (1)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \quad (2)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \quad (3)$$



論文要約 / 手法 / Decoder

- メモリと隠れ状態の

初期化は以下の通り

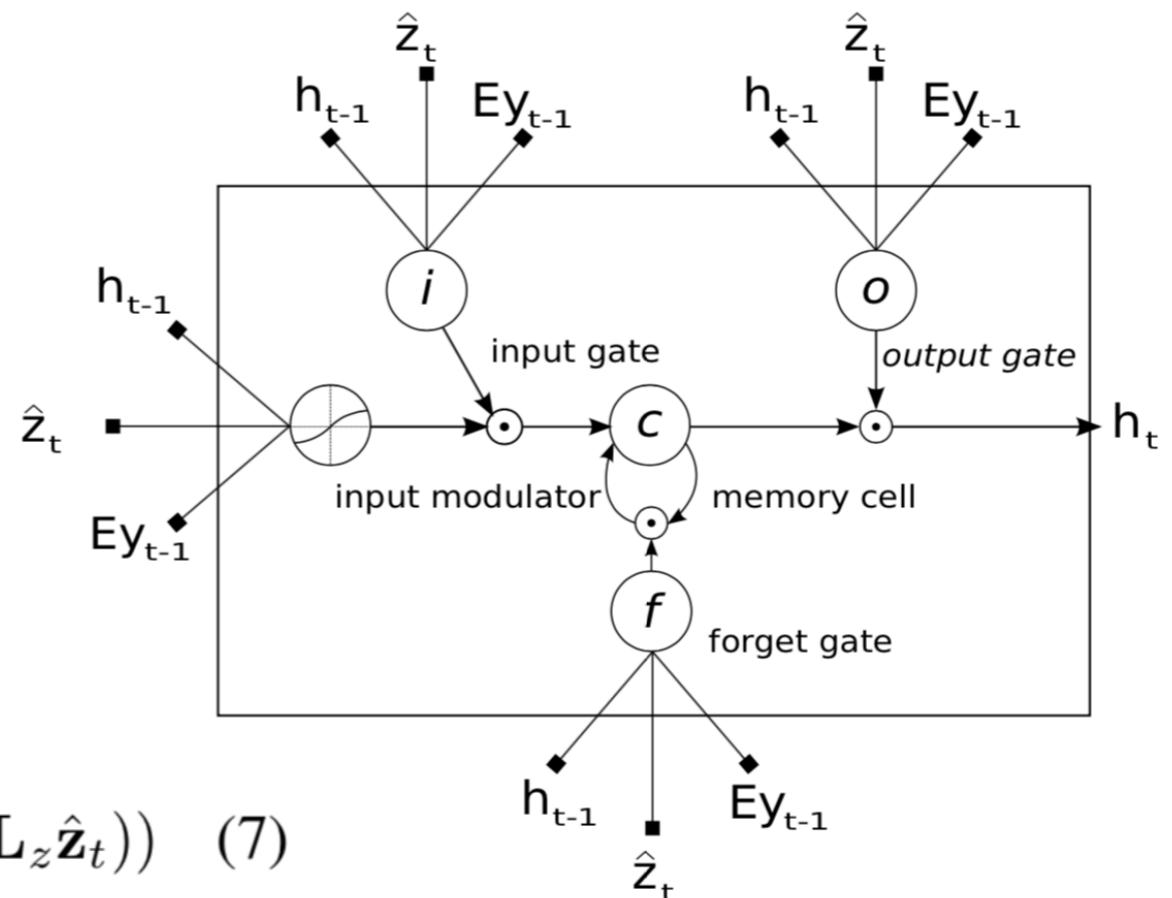
$$\mathbf{c}_0 = f_{\text{init},c}\left(\frac{1}{L} \sum_i^L \mathbf{a}_i\right)$$

$$\mathbf{h}_0 = f_{\text{init},h}\left(\frac{1}{L} \sum_i^L \mathbf{a}_i\right)$$

- 出力の語は L_o, L_h, L_z で

以下のように決まる

$$p(\mathbf{y}_t | \mathbf{a}, \mathbf{y}_1^{t-1}) \propto \exp(\mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h \mathbf{h}_t + \mathbf{L}_z \hat{\mathbf{z}}_t)) \quad (7)$$



論文要約 / 手法 / Decoder

- “context vector” \hat{z}_t は時刻tでの、関係のある画像の部分の表現と解釈できる

$$e_{ti} = f_{att}(\mathbf{a}_i, \mathbf{h}_{t-1})$$
$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^L \exp(e_{tk})}. \quad \hat{\mathbf{z}}_t = \phi(\{\mathbf{a}_i\}, \{\alpha_i\}),$$

- f_{att} により α_i (i番目のannotation vectorに対応する場所が注目すべき場所である確率)を計算 (attention model)
- annotation vectorと α_i から ϕ で \hat{z}_t を計算
- (f_{att} と) ϕ が“soft”と“hard”で異なる

論文要約 / 手法 / “Hard” attention

- s_t : 時刻tで画像のどの部分に注目するかのonehotベクトル

$$p(s_{t,i} = 1 \mid s_{j < t}, \mathbf{a}) = \alpha_{t,i}$$

$$\hat{\mathbf{z}}_t = \sum_i s_{t,i} \mathbf{a}_i.$$

- s_t 上の確率分布は $\{\alpha_i\}$ で条件付られるカテゴリ分布
- $\hat{\mathbf{z}}_t$ を確率変数と見ることができる
- ϕ は上のカテゴリ分布からサンプルされた a_i を返す関数

論文要約 / 手法 / “Hard” attention

- 学習時には $\log p(y | a)$ の変分下

限である L_s を直接最適化すれば
良い

- パラメータに対する勾配の計算には以下を用いる
 - モンテカルロ法による近似
 - 推定値の分散を小さくするための工夫(一番下)

$$\begin{aligned} L_s &= \sum_s p(s | \mathbf{a}) \log p(\mathbf{y} | s, \mathbf{a}) \\ &\leq \log \sum_s p(s | \mathbf{a}) p(\mathbf{y} | s, \mathbf{a}) \\ &= \log p(\mathbf{y} | \mathbf{a}) \end{aligned}$$

$$\frac{\partial L_s}{\partial W} = \sum_s p(s | \mathbf{a}) \left[\frac{\partial \log p(\mathbf{y} | s, \mathbf{a})}{\partial W} + \log p(\mathbf{y} | s, \mathbf{a}) \frac{\partial \log p(s | \mathbf{a})}{\partial W} \right].$$

$$\tilde{s}_t \sim \text{Multinoulli}_L(\{\alpha_i\})$$

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_{n=1}^N \left[\frac{\partial \log p(\mathbf{y} | \tilde{s}^n, \mathbf{a})}{\partial W} + \log p(\mathbf{y} | \tilde{s}^n, \mathbf{a}) \frac{\partial \log p(\tilde{s}^n | \mathbf{a})}{\partial W} \right]$$

$$\begin{aligned} \frac{\partial L_s}{\partial W} &\approx \frac{1}{N} \sum_{n=1}^N \left[\frac{\partial \log p(\mathbf{y} | \tilde{s}^n, \mathbf{a})}{\partial W} + \lambda_r (\log p(\mathbf{y} | \tilde{s}^n, \mathbf{a}) - b) \frac{\partial \log p(\tilde{s}^n | \mathbf{a})}{\partial W} + \lambda_e \frac{\partial H[\tilde{s}^n]}{\partial W} \right] \end{aligned}$$

論文要約 / 手法 / “Soft” attention

- (確率的な)サンプリングの代わりに \hat{z}_t の期待値を直接計算し、それを \hat{z}_t とする

$$\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^L \alpha_{t,i} \mathbf{a}_i \quad \phi(\{\mathbf{a}_i\}, \{\alpha_i\}) = \sum_i^L \alpha_i \mathbf{a}_i$$

- モデル全体が連続で微分可能なのでend-to-endに学習可能
- このモデルは注目する場所についての周辺尤度を近似していると解釈できる

論文要約 / 手法 / “Soft” attention

- Doubly Stochastic Attention
 - $\sum_t \alpha_{ti} \approx 1.$ となるように正則化を行う
 - 一連の生成で画像の全ての部分に平等に注目するようになる
 - 最終的には以下の目的関数を最小化する

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_i^L \left(1 - \sum_t^C \alpha_{ti}\right)^2$$

論文要約 / 手法 / 学習

- Optimizer: RMSProp, Adam
- Encoder: VGGNet pretrained on ImageNet
4番目の畳み込み層を抽出 ((14x14x512)->(196x512))
- 文の長さごとにマップを作り、同じ長さの文で minibatchを作る
- BLEU scoreによるearly stopping

論文要約 / 実験

- データ: Flickr8k, Flickr30k, MS COCO
- 評価指標: BLEU (1-4), METEOR
- 他の手法との違い (統一できていない部分)
 - 使用するCNN (VGGNet, GoogLeNet, AlexNet)
 - アンサンブルの有無
 - データ分割

論文要約 / 結果

- 定量評価 (SOTA)

Dataset	Model	BLEU				METEOR
		BLEU-1	BLEU-2	BLEU-3	BLEU-4	
Flickr8k	Google NIC(Vinyals et al., 2014) $^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a) $^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	67	44.8	29.9	19.5	18.93
	Hard-Attention	67	45.7	31.4	21.3	20.30
Flickr30k	Google NIC $^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	18.49
	Hard-Attention	66.9	43.9	29.6	19.9	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014) a	—	—	—	—	20.41
	MS Research (Fang et al., 2014) $^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014) $^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC $^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear $^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	23.90
	Hard-Attention	71.8	50.4	35.7	25.0	23.04

論文要約 / 結果

• 定性評価 (attentionの箇所を可視化)

Figure 2. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image. “soft” (top row) vs “hard” (bottom row) attention. (Note that both models generated the same captions in this example.)

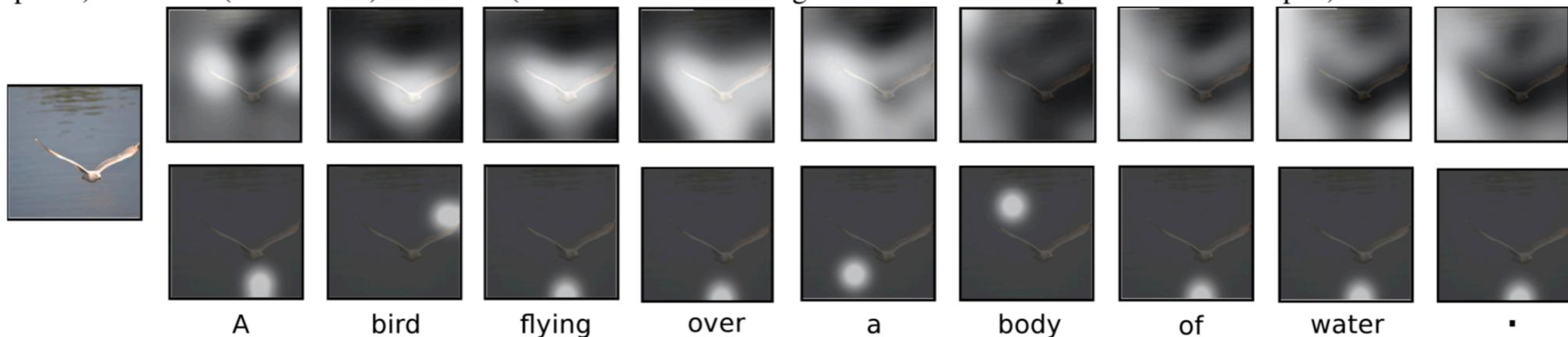


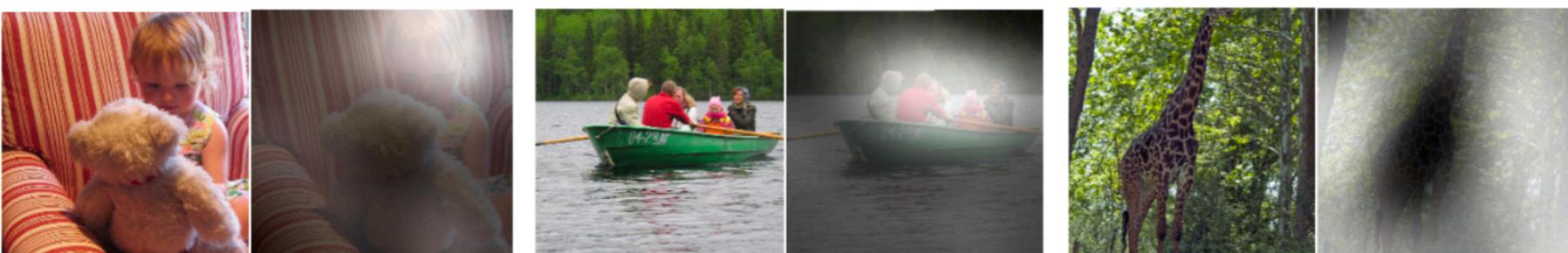
Figure 3. Examples of attending to the correct object (white indicates the attended regions, *underlines* indicated the corresponding word)



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

論文要約 / まとめ

- attention ベースの画像のキャプション生成を行なった
 - “hard” & “soft” attentionを提案
 - attentionが「どこ」の「何」に注目したかを可視化
 - 提案手法をベンチマークデータセット(Flicker30k, Flicker8k, MS COCO)で検証 (SOTA)