

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

# Постановка задачи

Есть текстовое описание картинки на входе и мы хотим получить фото-реалистичное изображение высокого разрешения с сохранением деталей на выходе .

This bird is white with some black on its head and wings, and has a long orange beak

This bird has a yellow belly and tarsus, grey back, wings, and brown throat, nape with a black face

This flower has overlapping pink pointed petals surrounding a ring of short yellow filaments

(a) StackGAN  
Stage-I  
64x64  
images



(b) StackGAN  
Stage-II  
256x256  
images



(c) Vanilla GAN  
256x256  
images



# Проблема с существующими методами

- Обычная версия GAN - не умеет работать с высокими разрешениями (из-за нестабильности).
- Conditional image generation - добавляет не все детали к изображению и не исправляет ряд дефектов.

# Идея

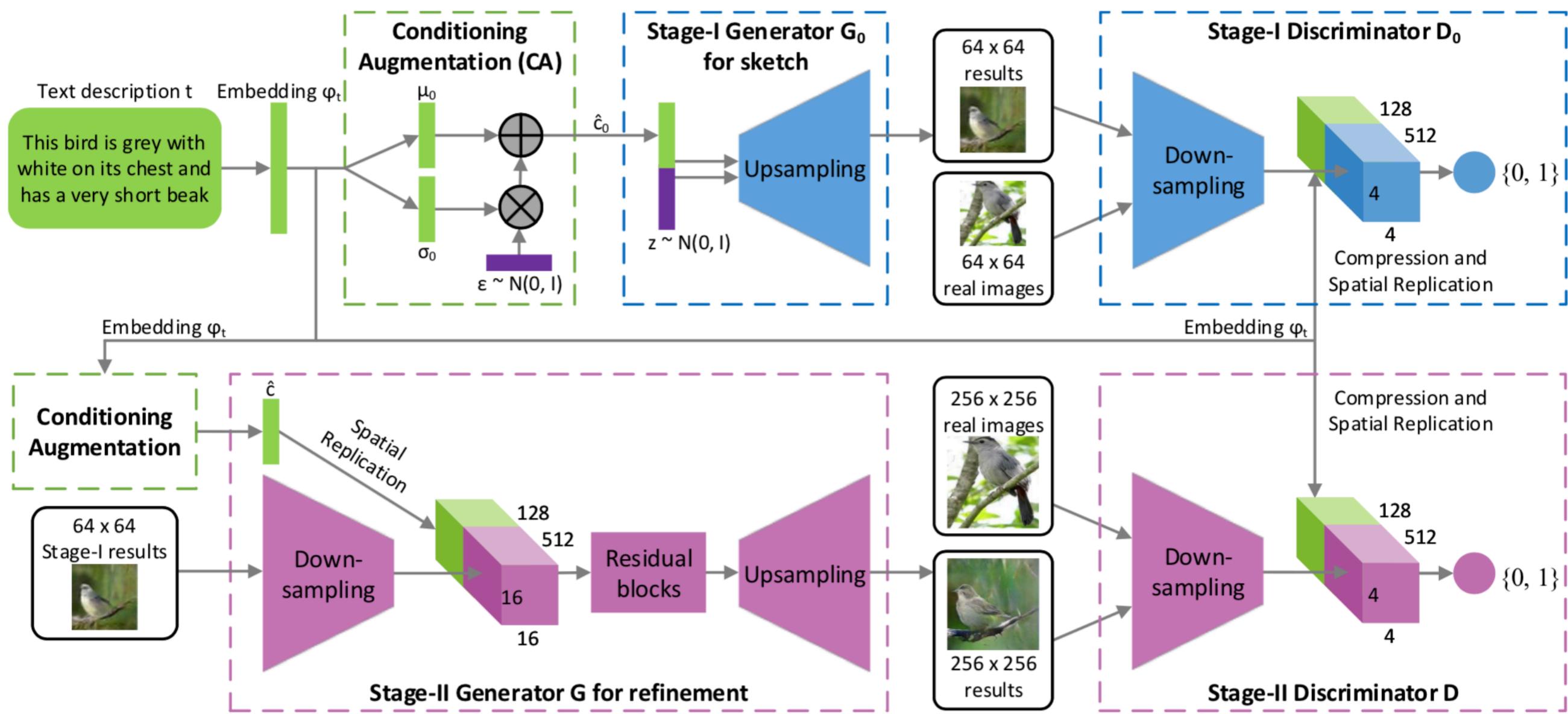
- Используется новая версия Stacked GAN
- Метод Conditioning Augmentation для стабилизации условного GAN и увеличения разнообразия результатов в полученной выборке

# Общий принцип

Поделим процесс на 2 стадии: Stage-I GAN & Stage-II GAN:

1. Stage-I GAN: Определяет примитивные формы и рисует фон из случайного вектора шума, получает изображение низкого разрешения
2. Stage-II GAN: Исправляет дефекты, полученные на предыдущей стадии и дополняет деталями, полученными при помощи текстового описания, выдавая на выход картинку более высокого разрешения.

# Схема алгоритма



# Conditional GAN

- Напомним, как выглядит обычный GAN

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

Где x - реальное изображение, полученное из распределения данных, а z - вектор шума (из гауссового распределения)

В условном GAN генератор и дискриминатор зависят от условной переменной с:  $G(z, c)$  &  $D(x, c)$

# Conditioning Augmentation

- Будем случайно сэмплировать из независимого нормального распределения

$$\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))$$

- Для избежания переобучения используется регуляризация

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

# Stage-I GAN

$$\hat{c_0} = \mu_0 + \sigma_0 \odot \epsilon$$

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

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

# Stage-II GAN

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

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

# Эксперименты

- Сравнение в экспериментах будет проводиться с GAN-INT-CLS и GAWWN.

В качестве датасетов рассмотрим:

- CUB - 200 видов птиц и 11,788 картинок
- Oxford-102 - 102 категории и 8,189 изображений цветов
- MS COCO - 80к изображений для обучения и 40к изображений для валидации, изображения имеют различный фон

# Метрика

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

Так же для оценивания используются результаты человеческого оценивания - для каждого предложения выбиралось 5 сгенерированных картинок и 10 человек оценивали их, оценка суммировалась

# Результаты метрик

Metric	Dataset	GAN-INT-CLS	GAWWN	Our StackGAN
Inception score	CUB	$2.88 \pm .04$	$3.62 \pm .07$	<b><math>3.70 \pm .04</math></b>
	Oxford	$2.66 \pm .03$	/	<b><math>3.20 \pm .01</math></b>
	COCO	$7.88 \pm .07$	/	<b><math>8.45 \pm .03</math></b>
Human rank	CUB	$2.81 \pm .03$	$1.99 \pm .04$	<b><math>1.37 \pm .02</math></b>
	Oxford	$1.87 \pm .03$	/	<b><math>1.13 \pm .03</math></b>
	COCO	$1.89 \pm .04$	/	<b><math>1.11 \pm .03</math></b>

# Полученные изображения

Text description

This bird is red and brown in color, with a stubby beak



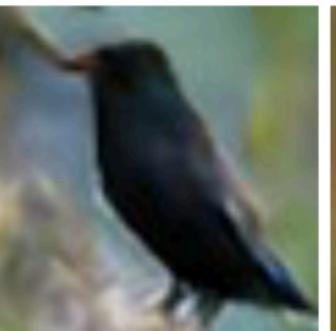
The bird is short and stubby with yellow on its body



A bird with a medium orange bill white body gray wings and webbed feet



This small black bird has a short, slightly curved bill and long legs



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



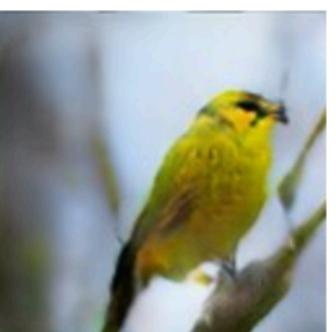
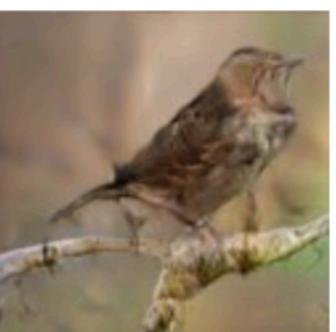
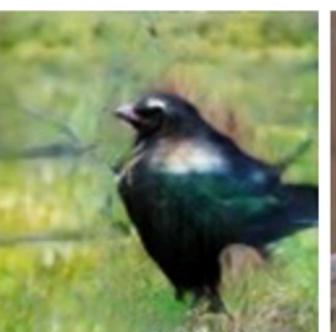
A small yellow bird with a black crown and a short black pointed beak



This small bird has a white breast, light grey head, and black wings and tail



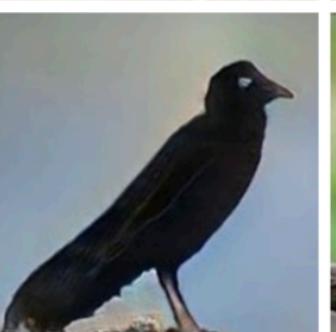
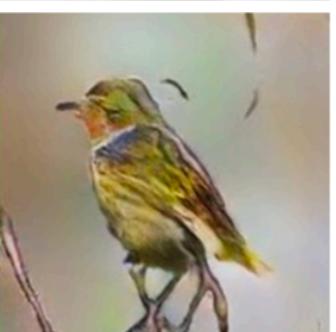
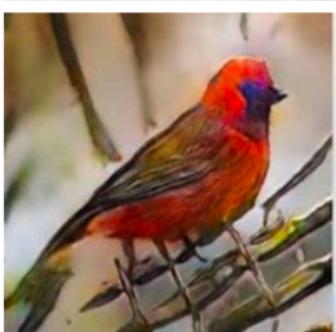
64x64  
GAN-INT-CLS



128x128  
GAWWN



256x256  
StackGAN



# Поэтапные результаты

Method	CA	Text twice	Inception score
64×64 Stage-I GAN	no	/	2.66 ± .03
	yes	/	2.95 ± .02
256×256 Stage-I GAN	no	/	2.48 ± .00
	yes	/	3.02 ± .01
128×128 StackGAN	yes	no	3.13 ± .03
	no	yes	3.20 ± .03
	yes	yes	3.35 ± .02
256×256 StackGAN	yes	no	3.45 ± .02
	no	yes	3.31 ± .03
	yes	yes	3.70 ± .04

# Выводы

- Stack GAN работает лучше чем существующие аналоги, показывая хорошие результаты на приведенных датасетах по 2 метрикам.
- Все составляющие алгоритма имеют большое значение, и интуитивное понимание подкрепляется реальными результатами.

# Список литературы

- <https://arxiv.org/pdf/1612.03242.pdf>