

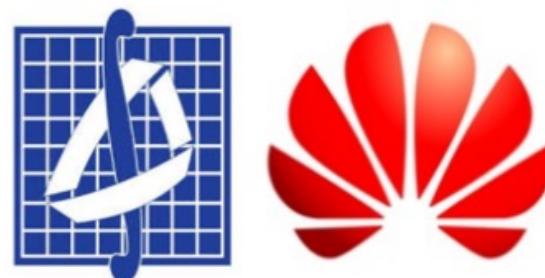
# Генерация изображений при помощи GAN

Иванюта Андрей

Лаборатория Интеллектуальных Систем,

Российский Исследовательский Институт Huawei

19 ноября 2019



## План лекции

- Что есть GAN?
- Типичная архитектура
- Примеры задач
- Примеры удачных и не очень GANов
- GANnotation
- StarGAN
- PG-GAN and StyleGAN
- DAVS
- GANы, ещё GANы, ещё больше GANов...
- Новейшие GANы: SMIT, SinGAN

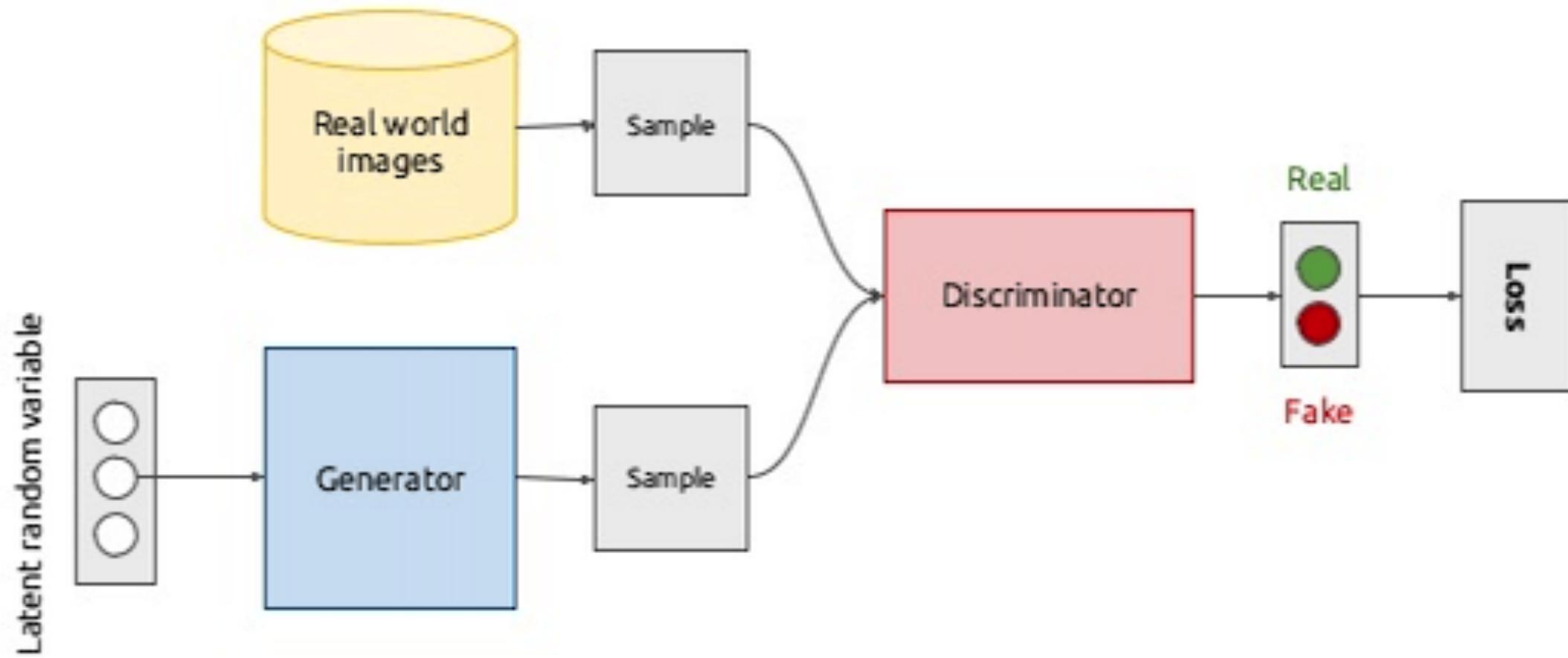
# Что есть GAN?

Generative Adversary Networks - Генеративно-состязательная сеть.

В простом случае это комбинация двух сетей: Генератора и Дискриминатора, которые учатся параллельно, используя выходы друг друга для обучения.

Конфигурация способствует плавно-нарастающему стимулированию сетей к обучению, и своего рода противостоянию: генератор пытается создать изображение чтобы дискриминатор принял его за настоящее, а дискриминатор в свою очередь пытается их различать.

# Типичная архитектура



## Типичные задачи для GAN:

Всевозможные генерации изображений, 3D-объектов, последовательностей и  
чего угодно ещё, из шума либо других объектов.

## Менее типичные задачи:

Смешивание объектов <https://arxiv.org/abs/1703.07195>

Детекция мелких объектов <https://arxiv.org/pdf/1706.05274v2>

Детекция аномалий <https://arxiv.org/pdf/1703.05921>

Дистилляция знаний <https://arxiv.org/abs/1906.08467>

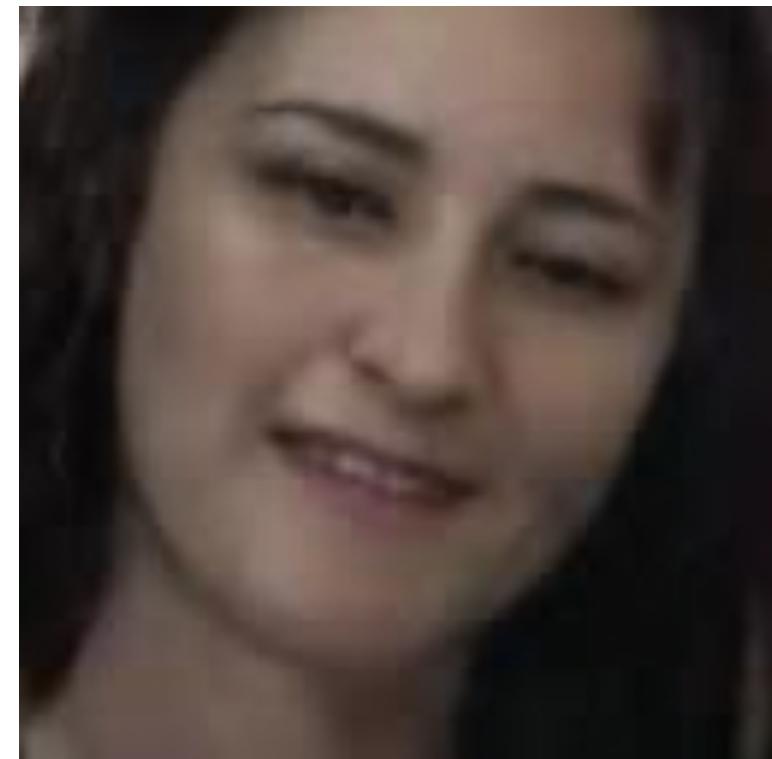
...

# GANnotation: архитектура

**Генератор:** resnet

**Дискриминатор:** взят из PatchGAN  
(свёртки + Leaky ReLU)

**Ключевая идея:** взять побольше  
лоссов, минимизирующих всё  
подряд.

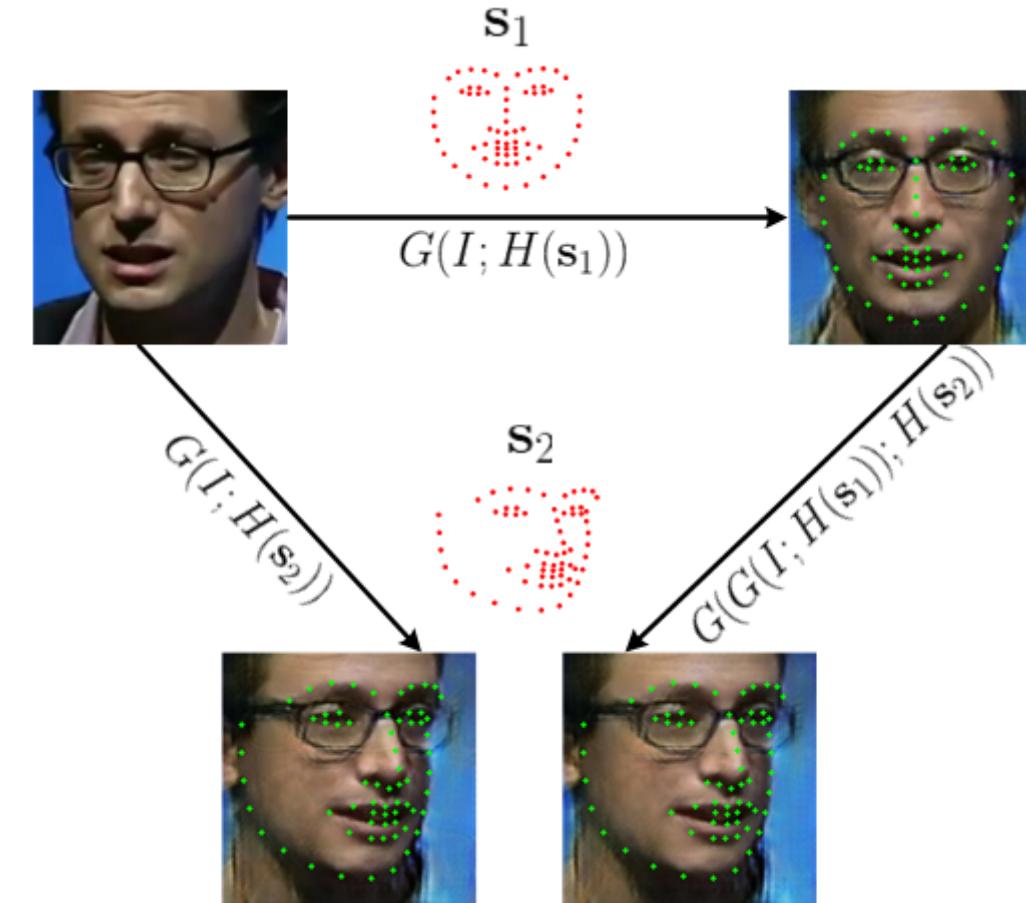


<https://arxiv.org/pdf/1811.03492>

# GANnotation

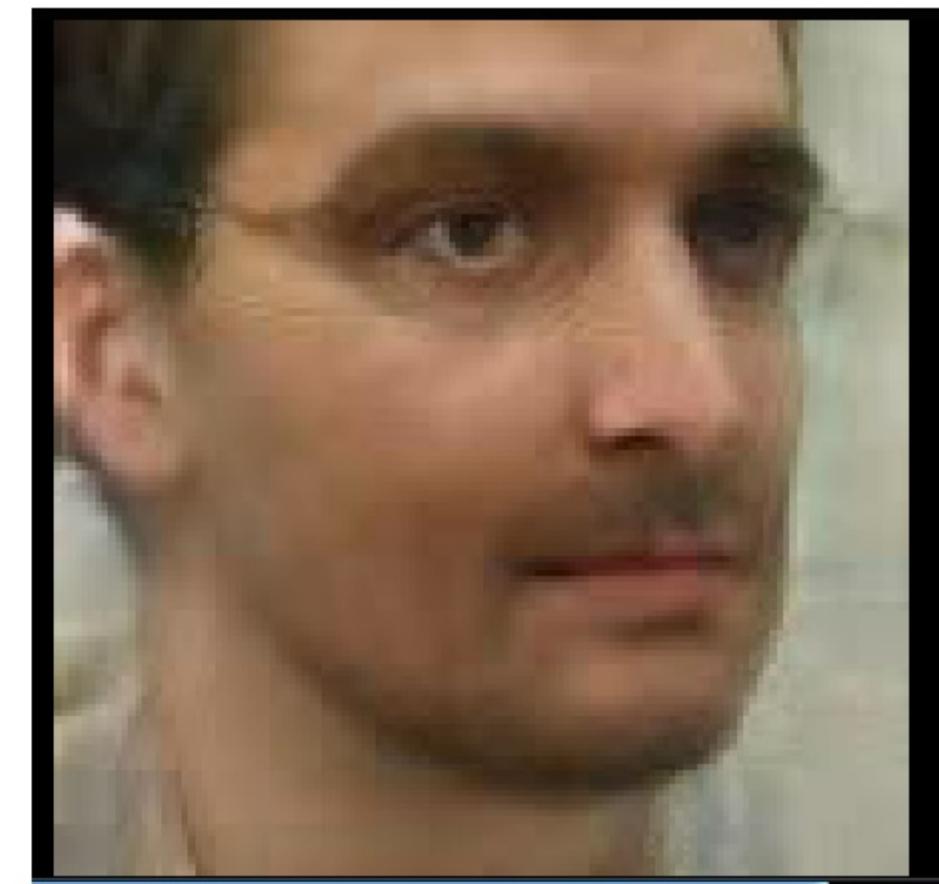
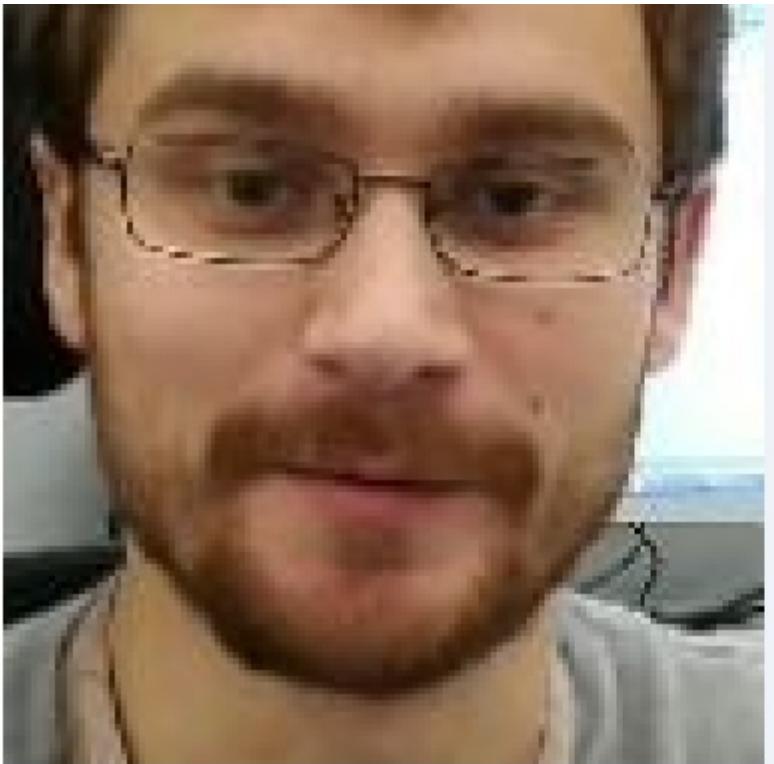
- Состязательный лосс
- Пиксельный лосс
- Последовательный лосс
- Тройной последовательный лосс
- Лосс сохранения личности
- Лосс восприятия

$$\begin{aligned}\mathcal{L}(G) = & \lambda_{adv} \mathcal{L}_{adv} + \lambda_{pix} \mathcal{L}_{pix} + \lambda_{self} \mathcal{L}_{self} \\ & + \lambda_{triple} \mathcal{L}_{triple} + \lambda_{id} \mathcal{L}_{id} + \lambda_{pp} \mathcal{L}_{pp} + \lambda_{tv} \mathcal{L}_{tv},\end{aligned}$$



$$\mathcal{L}_{triple} = \|G(I; H(s_2)) - G(G(I; H(s_1)); H(s_2))\|$$

# GANnotation:



ESanchezLozano commented 17 hours ago

Owner + ...

Hi,

Thanks for your message. After inspecting the image and after having shown it to people around in the lab, everyone agreed that both the generated image and the input image look alike (leaving aside the glasses and the beard). I would like you to try a similar approach with someone you don't know. Not even a famous

# StarGAN: архитектура

**Генератор:** взят из CycleGAN  
(модифицированный resnet)

**Дискриминатор:**

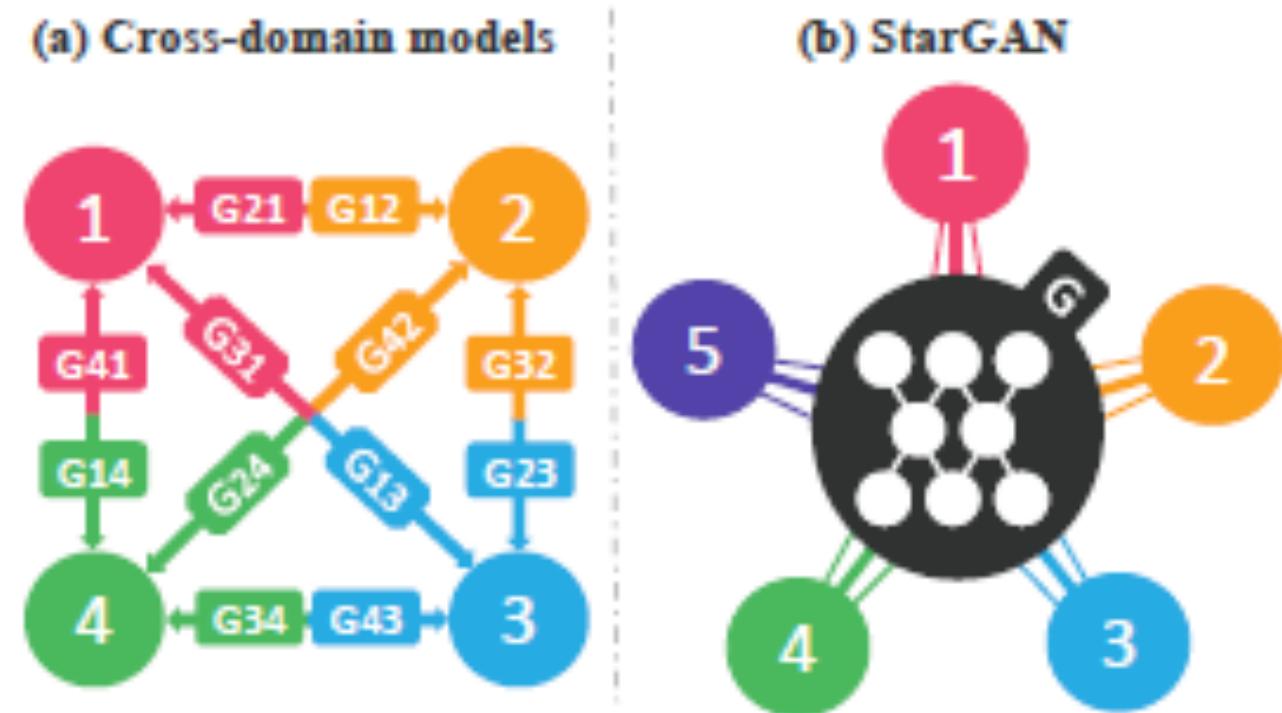
взят из PatchGAN  
(свёртки + Leaky ReLU)

**Лоссы:**

- Adversarial
- Domain Classification
- Reconstruction

**Ключевая идея:** На вход подаётся изображение + метка из какого домена в какой переводить. Это позволяет обучить 1 генератор с возможностью работы на различных доменах одновременно.

<https://arxiv.org/pdf/1811.03492>



# StarGAN

CelebA label

Black / Blond / Brown / Male / Young

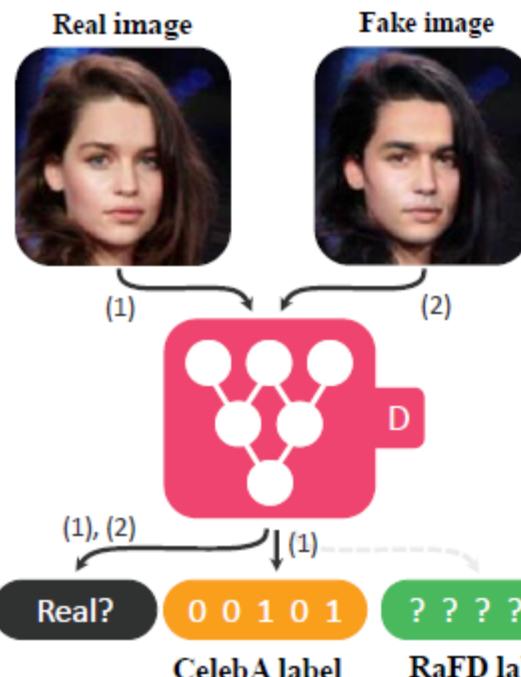
RaFD label

Angry / Fearful / Happy / Sad / Disgusted

Mask vector

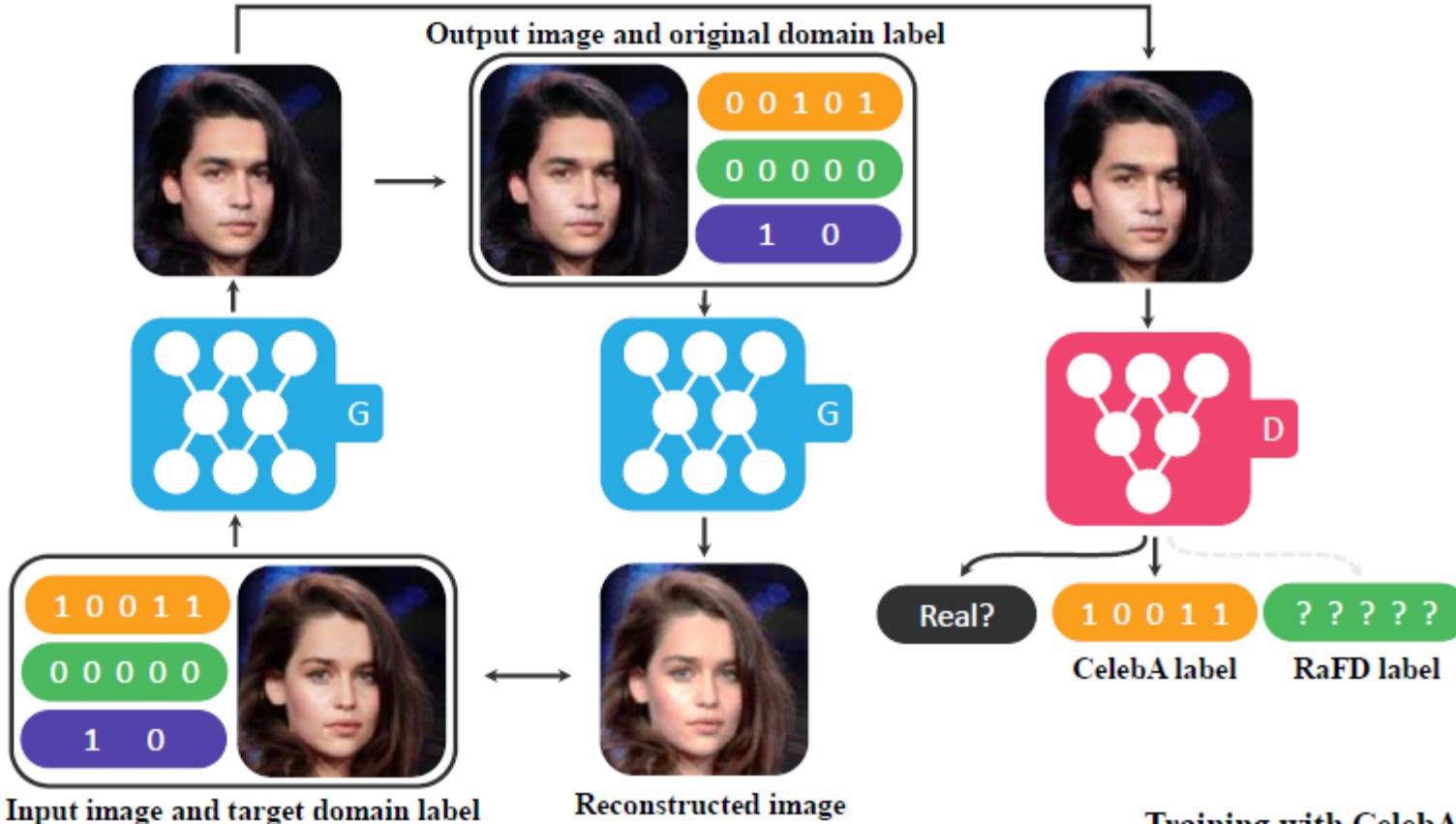
CelebA / RaFD

(a) Training the discriminator

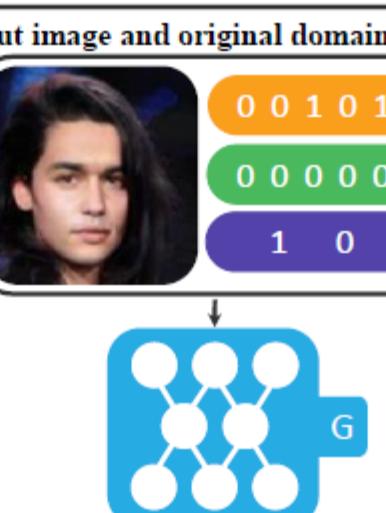


- (1) when training with real images  
(2) when training with fake images

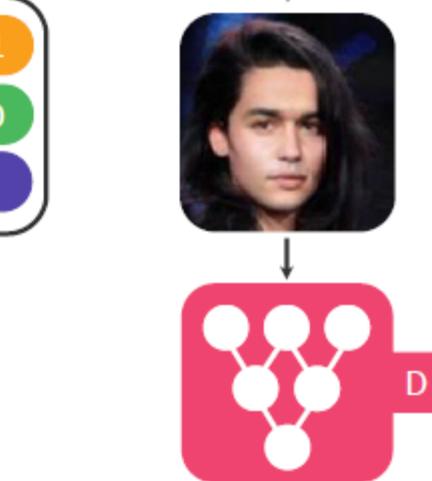
(b) Original-to-target domain



(c) Target-to-original domain



(d) Fooling the discriminator



# StarGAN: наши результаты

## С сохранением личности:

- Улыбка
- Очки
- Стиль/цвет волос
- ...
- Возраст
- Угол
- Освещенность

Очки



Улыбка



Возраст



Изменение нескольких атрибутов

## Без сохранения личности:

- Размер бровей/глаз/ушей...
- Пол
- Цвет кожи



# StarGAN: наши результаты, попытка вращения лица

Если сеть принимает любые атрибуты, то почему бы не сделать в качестве атрибута что-нибудь более сложное? Например угол поворота?

В меру возможностей и учитывая специфику датасета, можно считать что сеть хорошо справилась.



# Progressive GAN

FAKE images

**Генератор:**

Свёртки + Leaky ReLU

**Дискриминатор:**

Свёртки + Leaky ReLU

**Лоссы:**

Вассерштайн

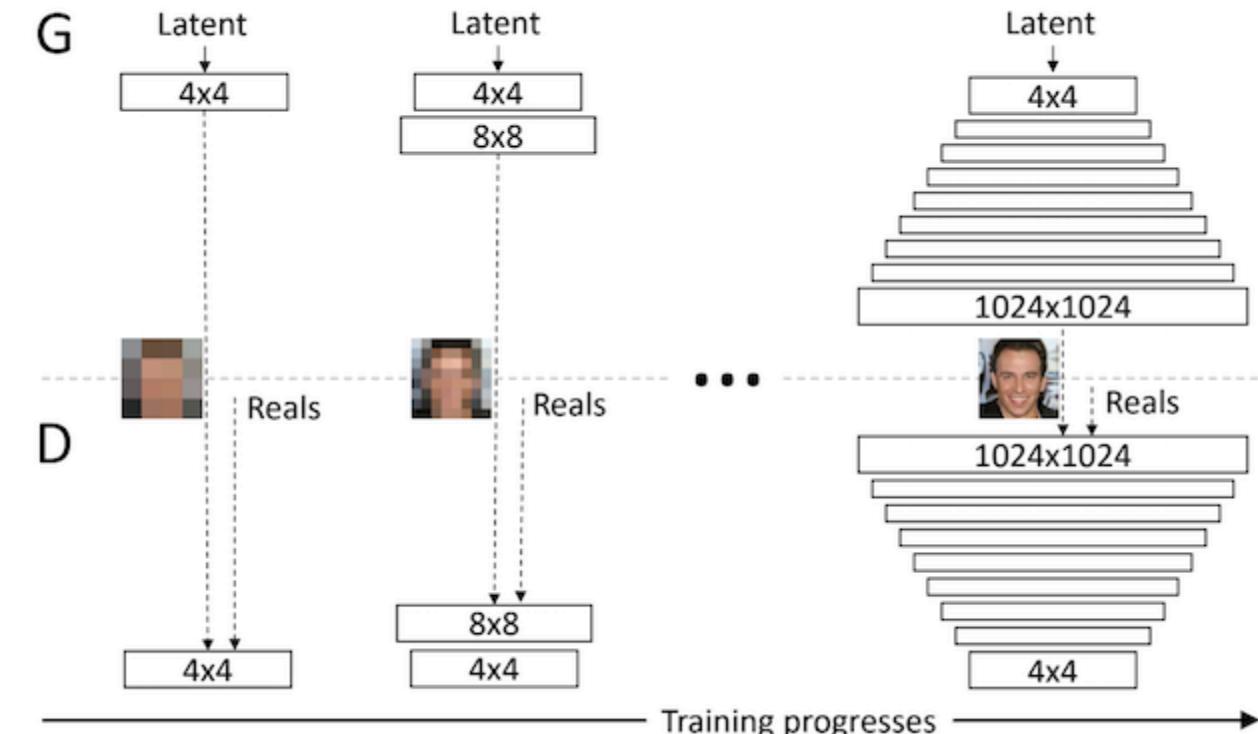
**Ключевая идея:**

Наращивать слои  
в процессе обучения



[https://research.nvidia.com/publication/2017-10\\_Progressive-Growing-of](https://research.nvidia.com/publication/2017-10_Progressive-Growing-of)

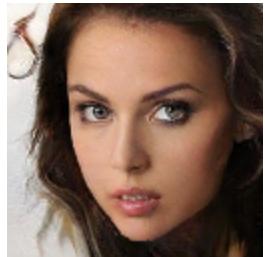
# Progressive GAN: архитектура



Generator	Act.	Output shape			Params
Latent vector	—	512	×	1	—
Conv $4 \times 4$	LReLU	512	×	4	4.2M
Conv $3 \times 3$	LReLU	512	×	4	2.4M
Upsample	—	512	×	8	—
Conv $3 \times 3$	LReLU	512	×	8	2.4M
Conv $3 \times 3$	LReLU	512	×	8	2.4M
Upsample	—	512	×	16	—
Conv $3 \times 3$	LReLU	512	×	16	2.4M
Conv $3 \times 3$	LReLU	512	×	16	2.4M
Upsample	—	512	×	32	—
Conv $3 \times 3$	LReLU	512	×	32	2.4M
Conv $3 \times 3$	LReLU	512	×	32	2.4M
Upsample	—	512	×	64	—
Conv $3 \times 3$	LReLU	256	×	64	1.2M
Conv $3 \times 3$	LReLU	256	×	64	590k
Upsample	—	256	×	128	—
Conv $3 \times 3$	LReLU	128	×	128	295k
Conv $3 \times 3$	LReLU	128	×	128	148k
Upsample	—	128	×	256	—
Conv $3 \times 3$	LReLU	64	×	256	74k
Conv $3 \times 3$	LReLU	64	×	256	37k
Upsample	—	64	×	512	—
Conv $3 \times 3$	LReLU	32	×	512	18k
Conv $3 \times 3$	LReLU	32	×	512	9.2k
Upsample	—	32	×	1024	—
Conv $3 \times 3$	LReLU	16	×	1024	4.6k
Conv $3 \times 3$	LReLU	16	×	1024	2.3k
Conv $1 \times 1$	linear	3	×	1024	51
Total trainable parameters				23.1M	

Discriminator	Act.	Output shape	Params
Input image	—	$3 \times 1024 \times 1024$	—
Conv $1 \times 1$	LReLU	$16 \times 1024 \times 1024$	64
Conv $3 \times 3$	LReLU	$16 \times 1024 \times 1024$	2.3k
Conv $3 \times 3$	LReLU	$32 \times 1024 \times 1024$	4.6k
Downsample	—	$32 \times 512 \times 512$	—
Conv $3 \times 3$	LReLU	$32 \times 512 \times 512$	9.2k
Conv $3 \times 3$	LReLU	$64 \times 512 \times 512$	18k
Downsample	—	$64 \times 256 \times 256$	—
Conv $3 \times 3$	LReLU	$64 \times 256 \times 256$	37k
Conv $3 \times 3$	LReLU	$128 \times 256 \times 256$	74k
Downsample	—	$128 \times 128 \times 128$	—
Conv $3 \times 3$	LReLU	$128 \times 128 \times 128$	148k
Conv $3 \times 3$	LReLU	$256 \times 128 \times 128$	295k
Downsample	—	$256 \times 64 \times 64$	—
Conv $3 \times 3$	LReLU	$256 \times 64 \times 64$	590k
Conv $3 \times 3$	LReLU	$512 \times 64 \times 64$	1.2M
Downsample	—	$512 \times 32 \times 32$	—
Conv $3 \times 3$	LReLU	$512 \times 32 \times 32$	2.4M
Conv $3 \times 3$	LReLU	$512 \times 32 \times 32$	2.4M
Downsample	—	$512 \times 16 \times 16$	—
Conv $3 \times 3$	LReLU	$512 \times 16 \times 16$	2.4M
Conv $3 \times 3$	LReLU	$512 \times 16 \times 16$	2.4M
Downsample	—	$512 \times 8 \times 8$	—
Conv $3 \times 3$	LReLU	$512 \times 8 \times 8$	2.4M
Conv $3 \times 3$	LReLU	$512 \times 8 \times 8$	2.4M
Downsample	—	$512 \times 4 \times 4$	—
Minibatch stddev	—	$513 \times 4 \times 4$	—
Conv $3 \times 3$	LReLU	$512 \times 4 \times 4$	2.4M
Conv $4 \times 4$	LReLU	$512 \times 1 \times 1$	4.2M
Fully-connected	linear	$1 \times 1 \times 1$	513
Total trainable parameters			<b>23.1M</b>

# Progressive GAN: наши результаты



Feature	TensorFlow version	Original Theano version
Branch	master (this branch)	original-theano-version
Multi-GPU support	Yes	No
FP16 mixed-precision support	Yes	No
Performance	High	Low
Training time for CelebA-HQ	2 days (8 GPUs) 2 weeks (1 GPU)	1–2 months
Repro CelebA-HQ results	Yes – very close	Yes – identical
Repro LSUN results	Yes – very close	Yes – identical
Repro CIFAR-10 results	No	Yes – identical
Repro MNIST mode recovery	No	Yes – identical
Repro ablation study (Table 1)	No	Yes – identical

# Progressive GAN: наши результаты



# StyleGAN

**Ключевая идея:**

Подавать не один вектор шума, а два, и не только на «вход», а на каждый слой

Expected training times for the default configuration using Tesla V100 GPUs:

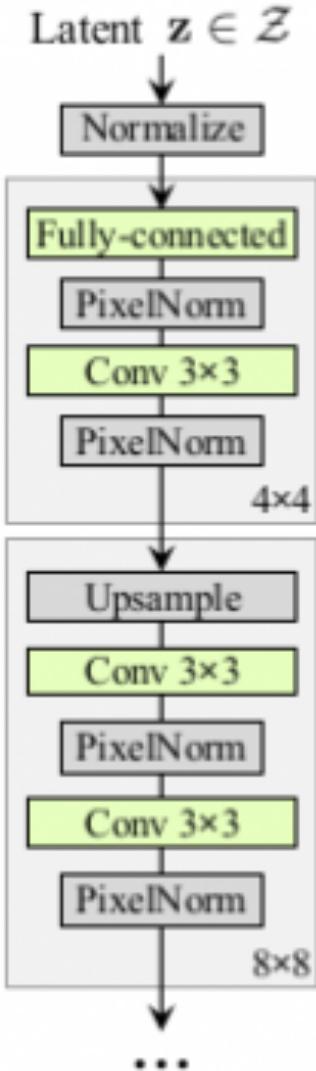
GPUs	1024×1024	512×512	256×256
1	41 days 4 hours	24 days 21 hours	14 days 22 hours
2	21 days 22 hours	13 days 7 hours	9 days 5 hours
4	11 days 8 hours	7 days 0 hours	4 days 21 hours
8	6 days 14 hours	4 days 10 hours	3 days 8 hours

<https://arxiv.org/abs/1812.04943>

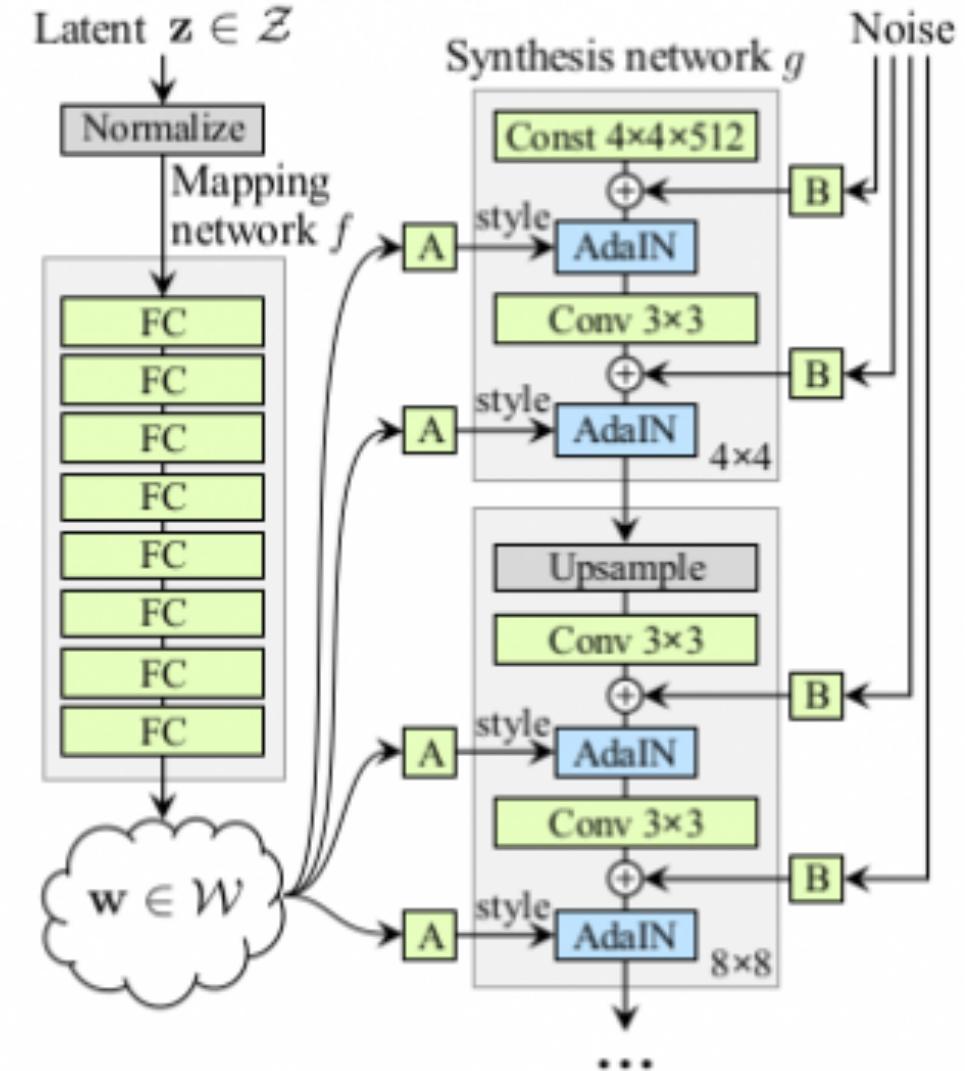


# StyleGAN: архитектура

Method	CelebA-HQ	FFHQ
A Baseline Progressive GAN [30]	7.79	8.04
B + Tuning (incl. bilinear up/down)	6.11	5.25
C + Add mapping and styles	5.34	4.85
D + Remove traditional input	5.07	4.88
E + Add noise inputs	<b>5.06</b>	4.42
F + Mixing regularization	5.17	<b>4.40</b>

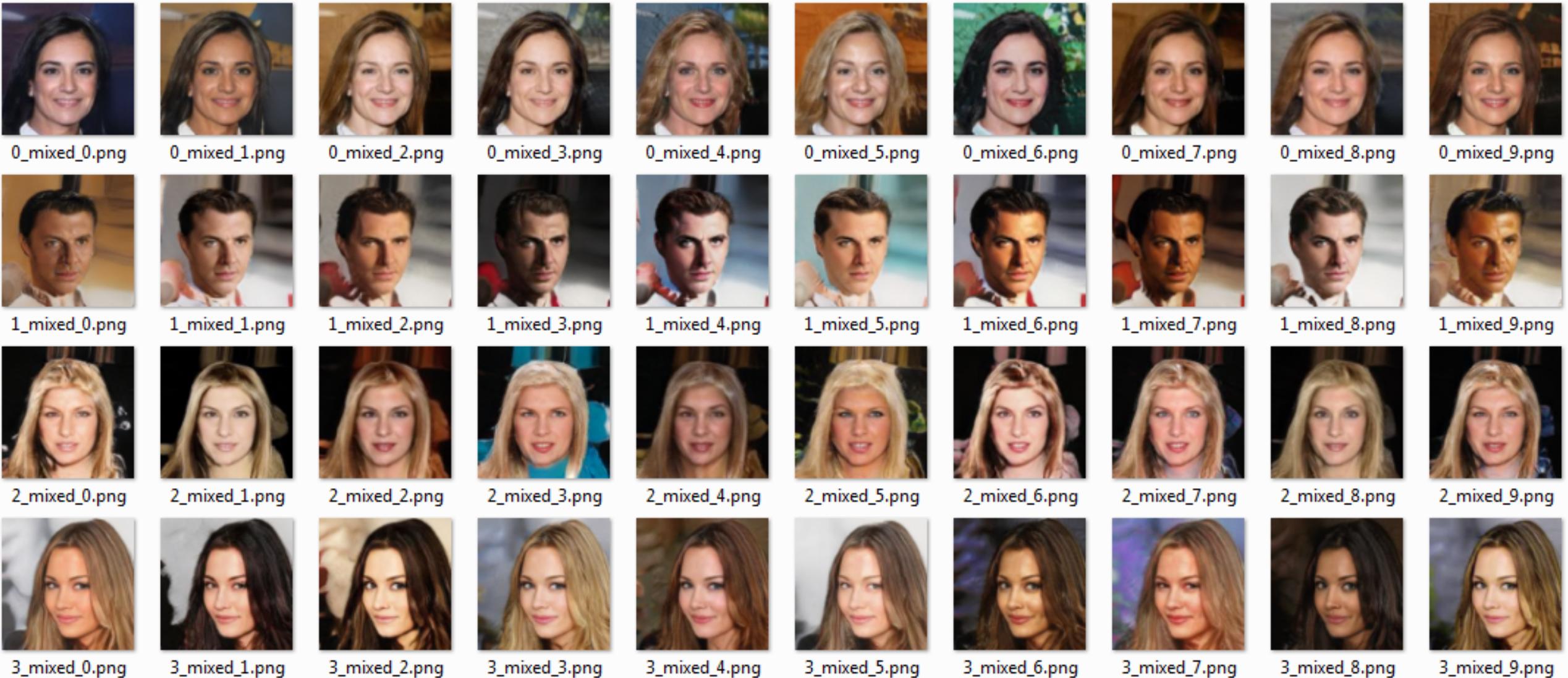


(a) Traditional



(b) Style-based generator

# StyleGAN: наш результат



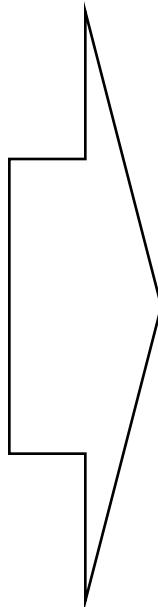
# Идея из TLGAN (transparent latent)

## Ключевая идея:

Над вектором шума, можно надстроить регрессию (логистическую, линейную...) обучение которой приводит к выявлению закономерности соответствия шума и изображения.

 В настоящее время не удается отобразить рисунок.

# StyleGAN + TLGAN: наш результат



“Лицо” StyleGAN

изменение возраста

**Угол**



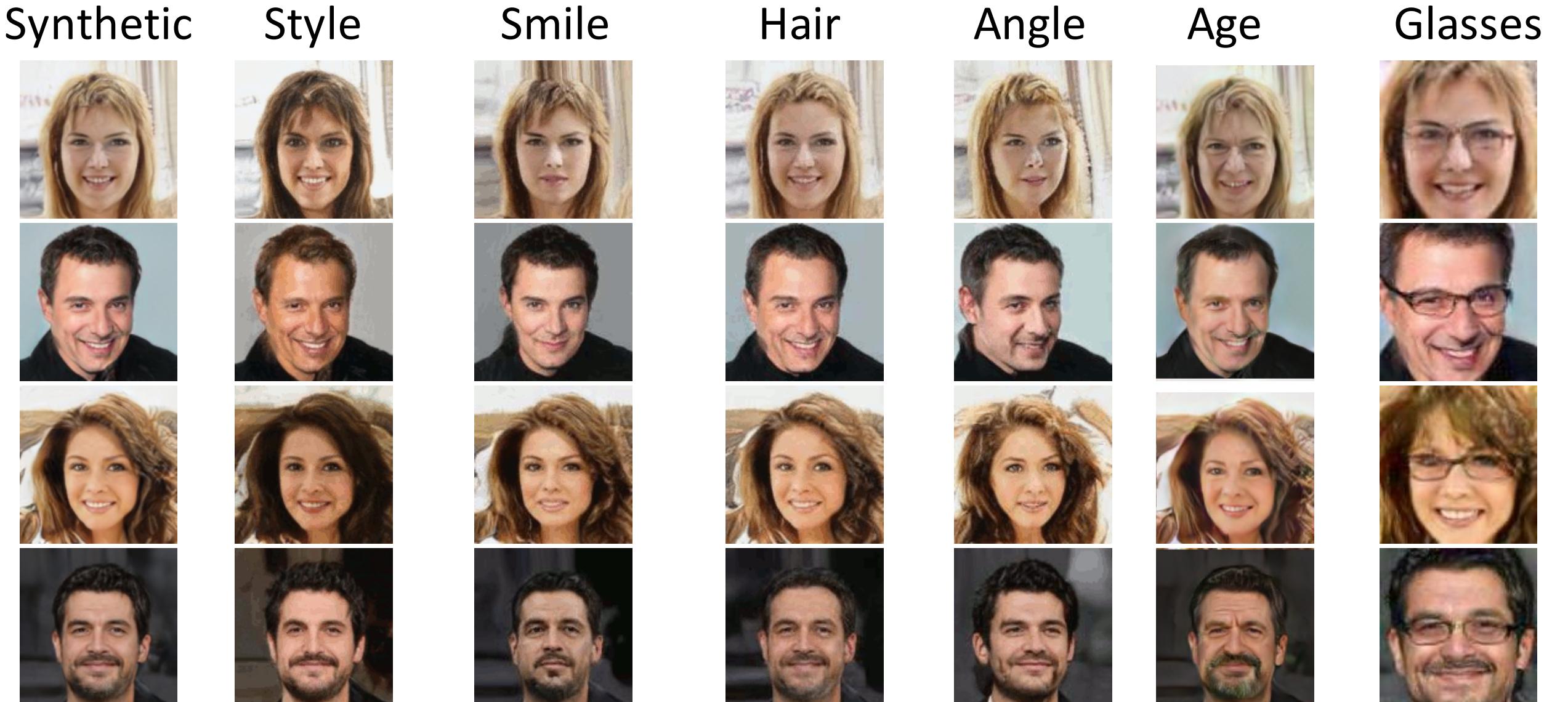
**Очки**



Чёлка



# StyleGAN + TLGAN: наш результат



# StyleGAN encoder

## **Важная идея!!!**

Можно подобрать такой вектор, пропускание которого через GAN будет давать наиболее близкий результат!

Пример: берём предобученную VGG16, переводим исходную и «желаемую» картинки в пространство признаков, проводим оптимизацию по минимизации расстояния в пространстве признаков. Это приведёт к тому что желаемая картинка приблизится максимально близко к исходной, но в отличии от нее будет не настоящей а сгенерённой.

<https://github.com/Puzer/stylegan-encoder>

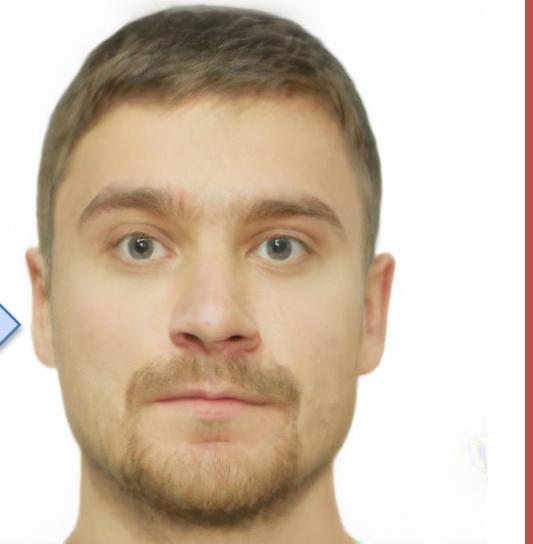
# StyleGAN encoder



Crop + align



Reconstruction



Это изображение получено подбором  
вектора в латентном пространстве

# StyleGAN + encoder + TLGAN



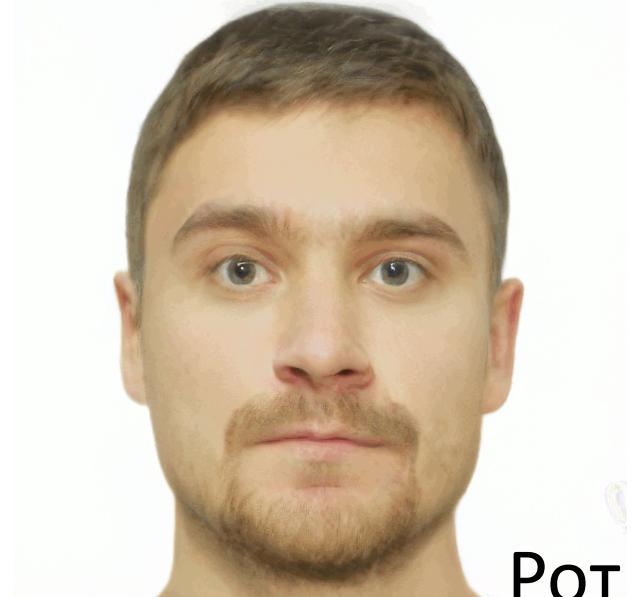
Угол



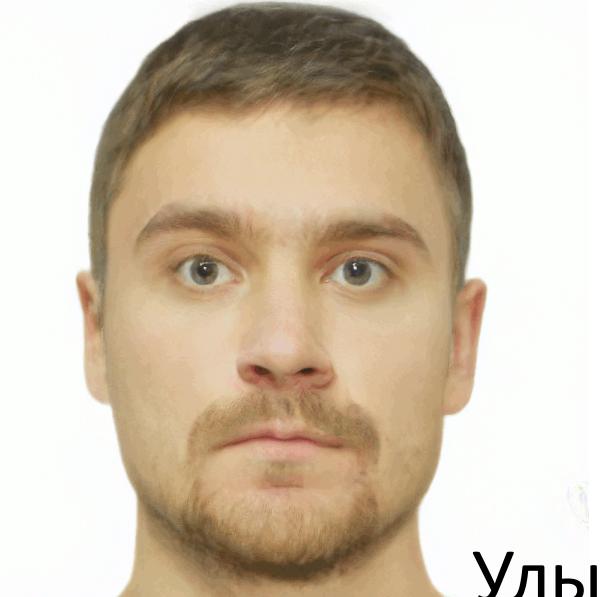
Очки



Возраст



Рот

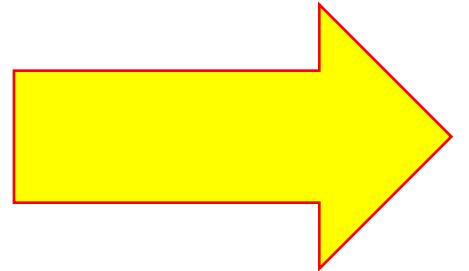


Улыбка



Чёлка

# StyleGAN + encoder + TLGAN



Вход низкого разрешения

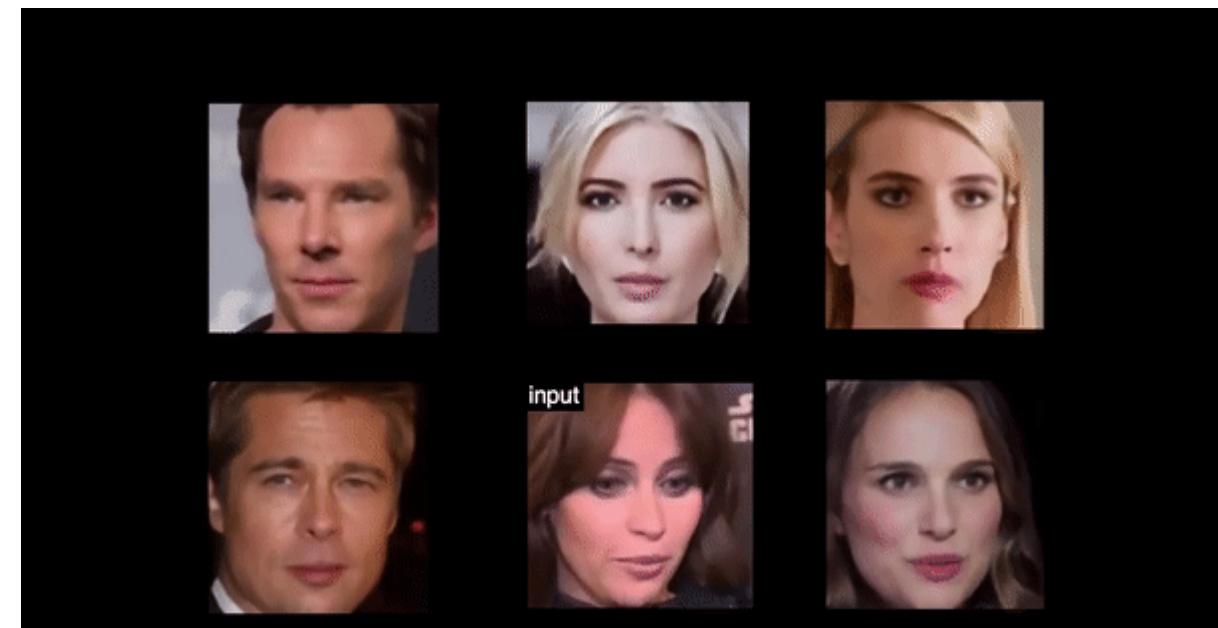
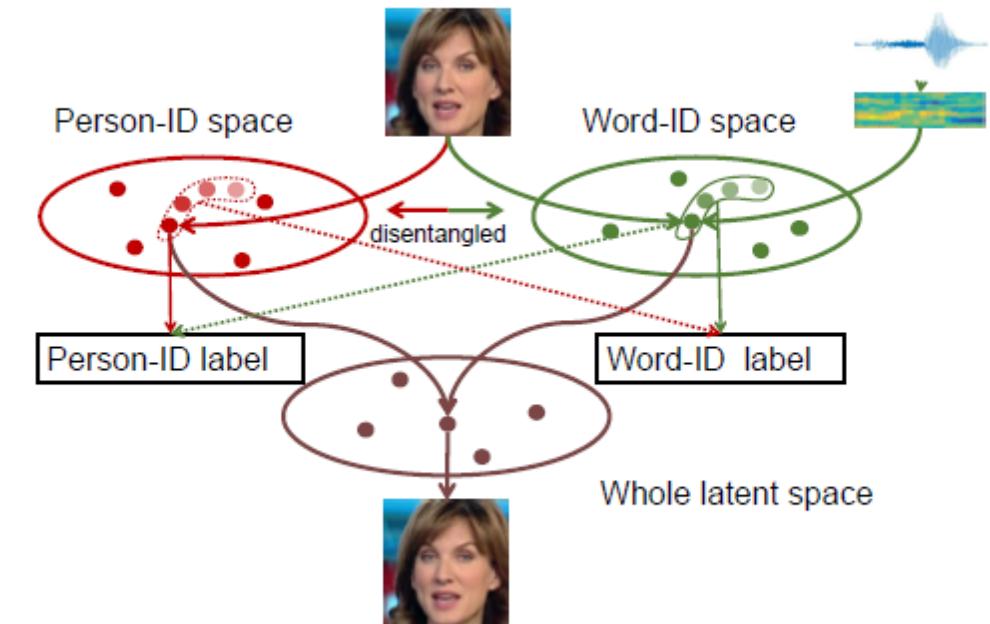


Реконструкция в высоком разрешении

# DAVS (Disentangled Audio-Visual System)

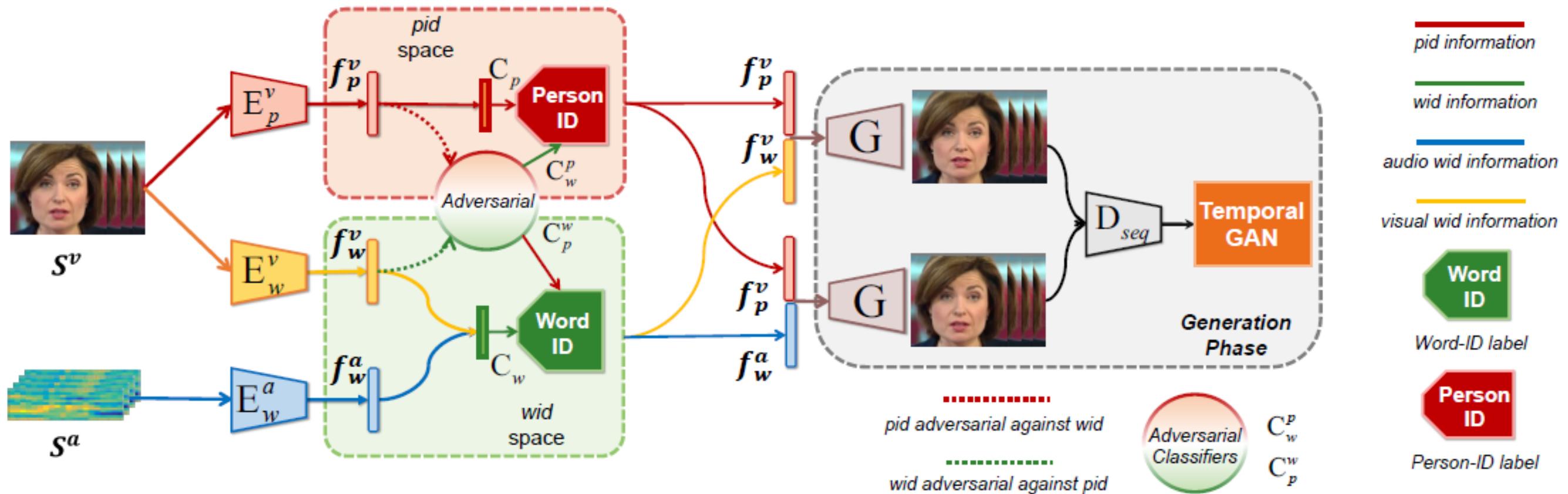
## Ключевая идея:

Система учится генерить изображение для пары «изображение + слово», где «слово» - некая звуковая единица

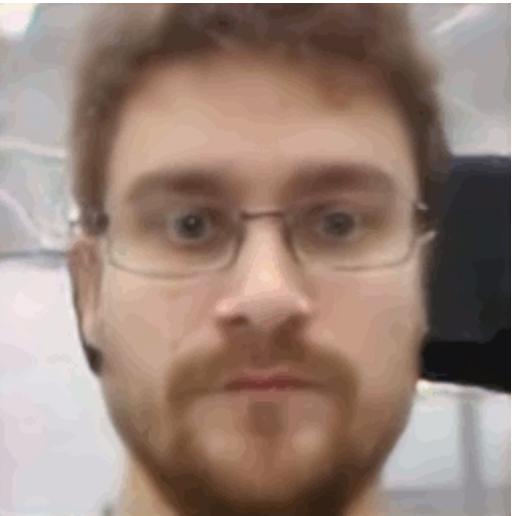


<https://arxiv.org/abs/1807.07860>

# DAVS



# DAVS: наши результаты



я



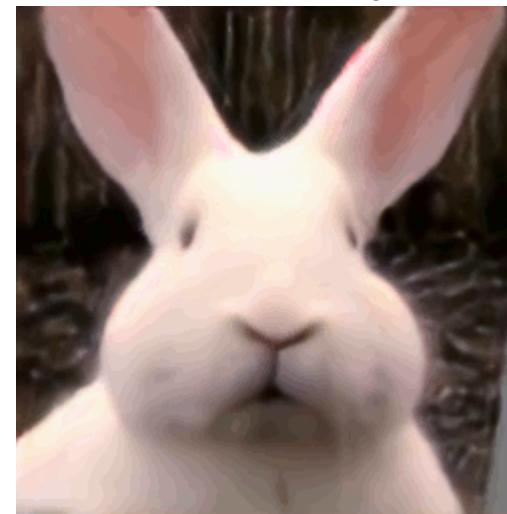
азиатское лицо



европейское лицо



мой кот



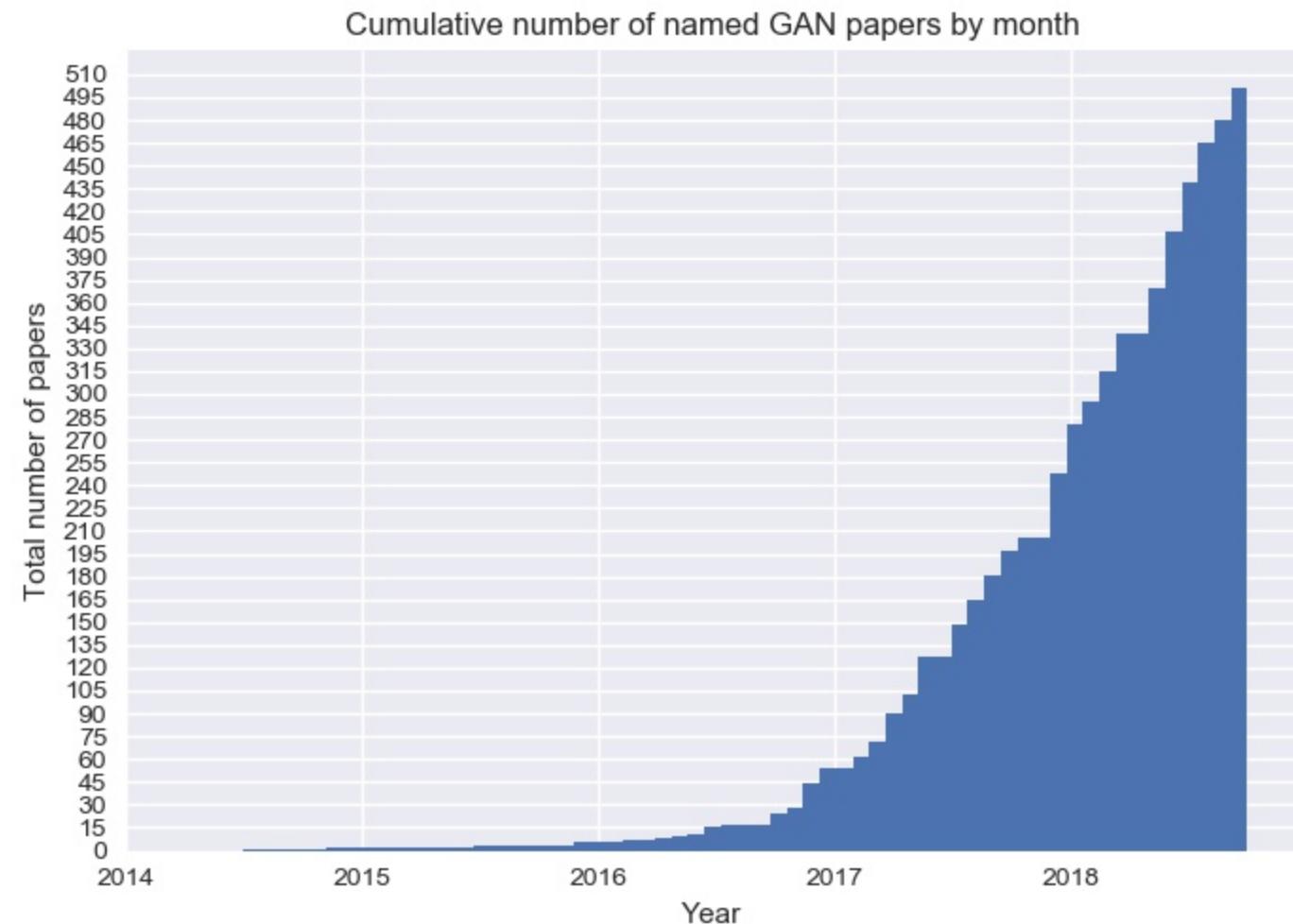
нарисованный кролик



# GANы, ещё GANы, ещё больше GANов...

[https://github.com/hindupuravinash/  
the-gan-zoo](https://github.com/hindupuravinash/the-gan-zoo)

Где-то в сентябре прошлого года  
автор забил на подсчёт количества  
и составление списка ганов.



# Новейшее: SMIT (Stochastic Multi-Label Image-to-Image Translation)

## Ключевая идея:

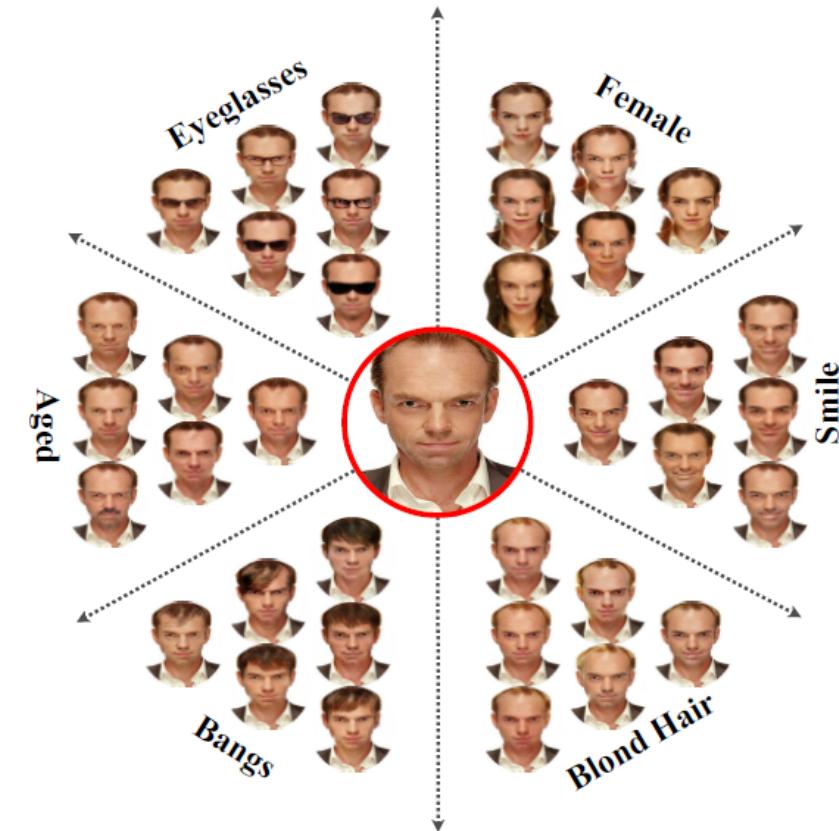
Объединить преимущества известных ганов

1. Передача меток класса как в StarGAN
2. Преобразование стиля посредством второго вектора как в StyleGAN
3. Энкодинг изображения и поиск направлений изменений атрибутов как TLGAN

...

## 5. Profit

	CycleGAN [55]	BiCycleGAN [56]	StarGAN [12]	MUNIT&alike [23, 3, 39]	DRIT [34]	GANimation [46]	SMIT (ours)
Unpaired Training	✓		✓	✓	✓	✓	✓
Multimodal Generation		✓		✓	✓		✓
Multiple Attributes			✓			✓	✓
One Single Generator			✓			✓	✓
Fine-grained Transformation				✓		✓	✓
Continuous Label Interpolation						✓	✓
Style Transformation				✓	✓		✓
Style Interpolation				✓	✓		✓
Attention Mechanism						✓	✓



<https://arxiv.org/pdf/1812.03704>

# Новейшее: SinGAN

Анимация по 1 картинке.

*Single training image*



*Random samples from a single image*



<https://arxiv.org/pdf/1905.01164>

# Новейшее: SinGAN

SinGAN: Learning a Generative Model  
from a Single Natural Image

Single Image Animation - Results

# Благодарю за внимание

