# Glow: Generative Flow with Invertible 1×1 Convolutions

Нугаманов Эдуард

Москва, 2019

# Зачем нужны генеративные модели

- Позволяют моделировать данные
- Извлекают полезные признаки из тонн неразмеченных данных

# Виды генеративных моделей

- Likelihood-based:
  - Авторегрессионные (PixelRNN, PixelCNN)
  - VAE
  - Flow-based (NICE, Real NVP, GLOW)
- GANs

# Почему Flow-based?

- Точный вывод латентных переменных, расчет и оптимизация точного log-likelihood
- Вывод и сэмплирование эффективно параллелятся
- Получается хорошее скрытое представление
- Обратимые нейросети могут потреблять константное по отношению к глубине количество памяти [1].

# Пример работы GLOW

# Flows

- Найти такое преобразование $z = f(x)$ данных, что $P_Z(z)$ – простое

- $f$ - обратимая и $f = f_1 \circ f_2 \circ \cdots \circ f_K$ - (normalizing) flow

# Обучение Flow-based models

- Оптимизируем:

$$\log p_X(x) = \log p_Z(f(x)) + \sum_{i=1}^{K} \log \left| \det \left( \frac{dz_i}{dz_{i-1}} \right) \right|$$

(здесь $z_0 = x, z_K = z, z_i = f_i(z_{i-1})$)

Причем $p_Z(z)$ - тоже может меняться на обучении

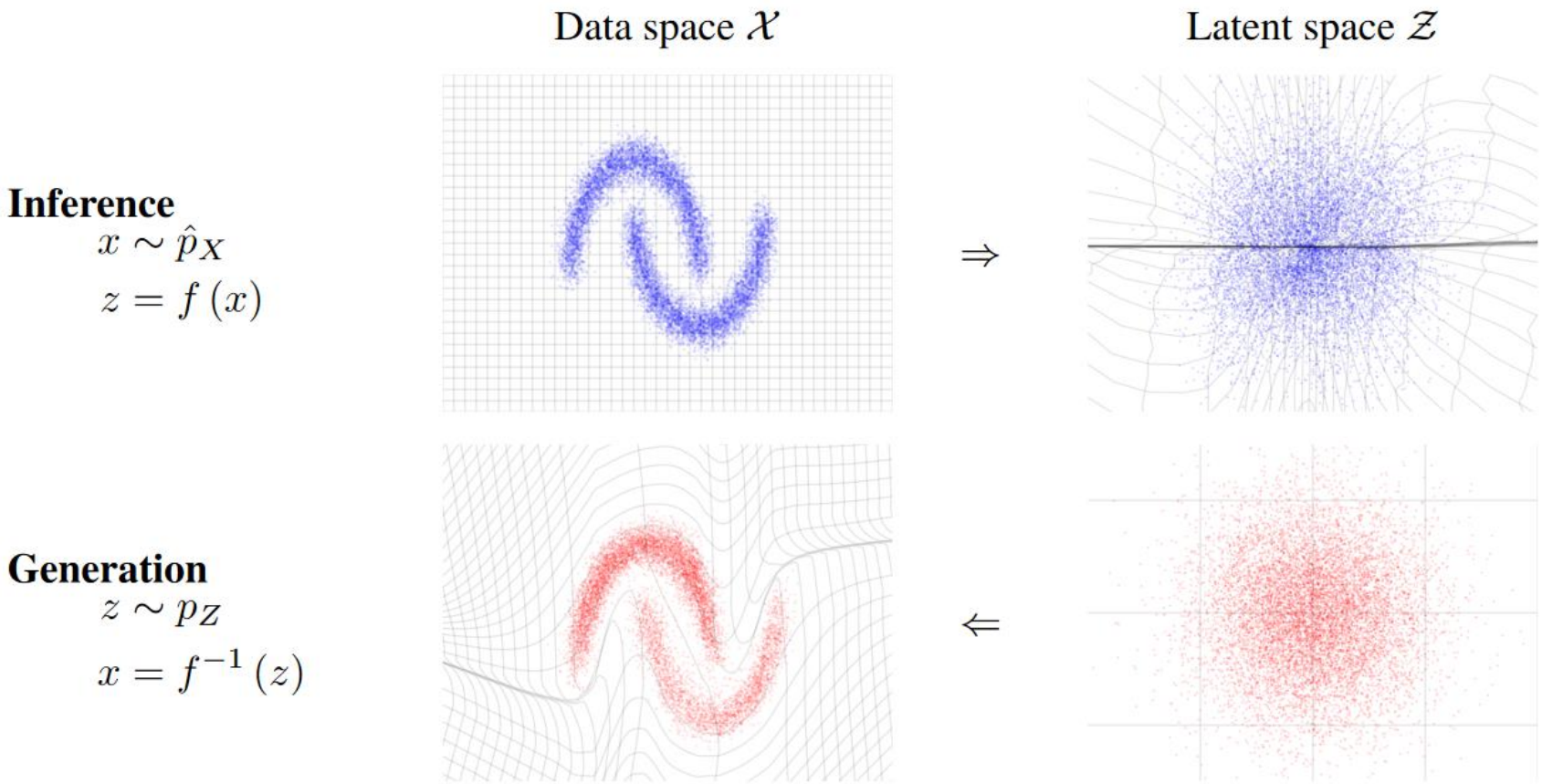
# Требования к потоку

- $\det\left(\frac{dz_i}{dz_{i-1}}\right)$должен легко вычисляться (треугольная матрица)
- $f^{-1}(z)$ тоже легко вычисляется

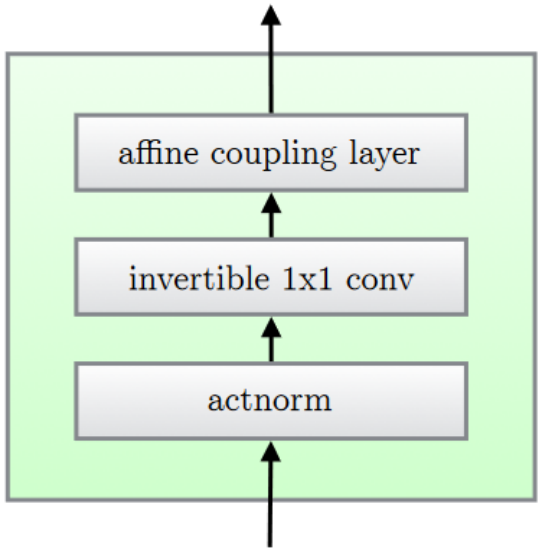
Тогда сэмплировать $x$ тоже легко:

$$\mathrm{z} \sim p_Z(z)$$
$$x = f^{-1}(z)$$

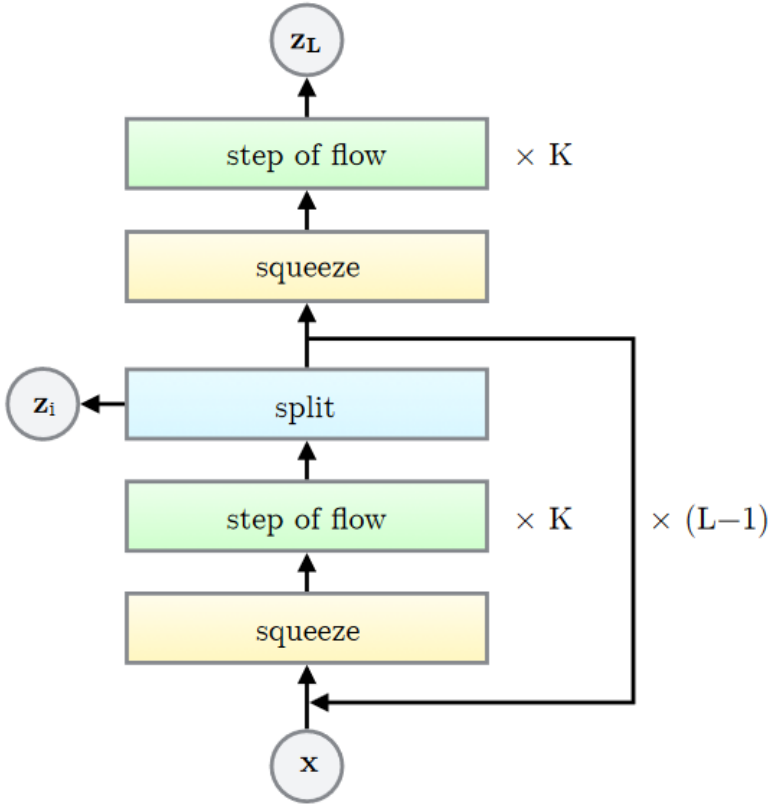
# Пример


Data space $\mathcal{X}$
Latent space $\mathcal{Z}$
**Inference**
$x \sim \hat{p}_X$
$z = f(x)$
⇒
**Generation**
$z \sim p_Z$
$x = f^{-1}(z)$
⇐

# Компоненты GLOW




(a) One step of our flow.




(b) Multi-scale architecture (Dinh et al., 2016).

# Actnorm

- Обычно для лучшего обучения глубоких моделей используем Batch normalization
- Недостаток BN: на маленьких батчах много шума, а для больших изображений нужные маленькие батчи

- Activation normalization: scaling и bias параметры.
- Инициализация: так, что активации имеют 0 среднее и 1 дисперсию
- Дальше обучаем как обычные параметры

# Coupling layers

- **General coupling layer:** $\dim(x) = D$; $I_1, I_2$ - разделение множества индексов $[1, D]$, $|I_1| = d$ (в GLOW делим каналы пополам).
- Тогда $y = (y_{I_1}, y_{I_2})$, где

$$y_{I_1} = x_{I_1}$$
$$y_{I_2} = g\left(X_{I_2}; m(x_{I_1})\right)$$

Где $g$ – *coupling law,* обратимое отображение относительно первого аргумента.

# Coupling layers: смысл

- Якобиан: $\frac{dy}{dx} = \begin{bmatrix} I_d & 0 \\ \frac{dy_{I_2}}{dx_{I_1}} & \frac{dy_{I_2}}{d_{x_{I_2}}} \end{bmatrix} = \frac{dy_{I_2}}{d_{x_{I_2}}}$, зависит только от $g$,

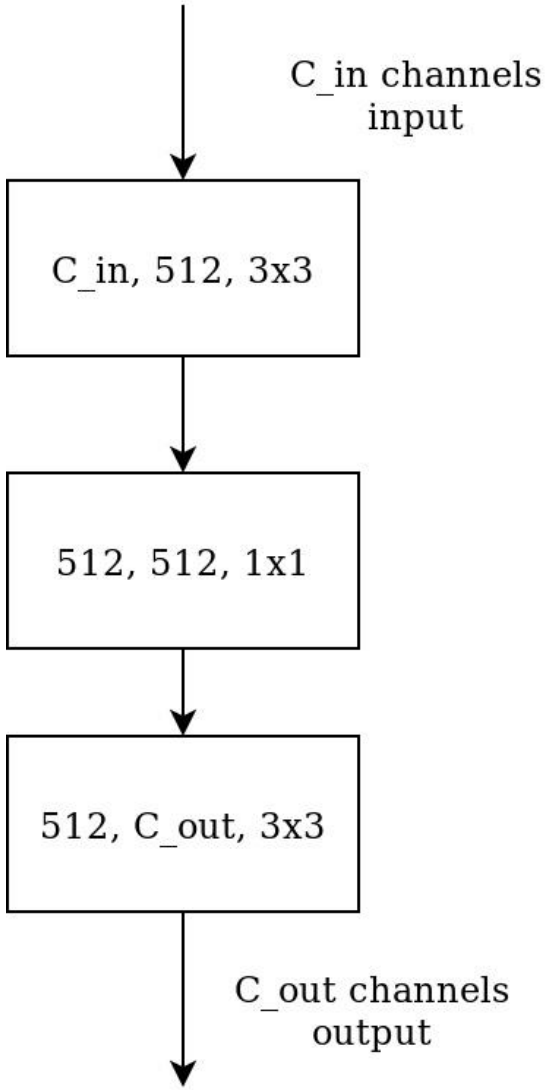
- Значит, $m$ может быть достаточно сложной (NN)

# Coupling layers: виды

- Additive coupling layer[2]: $g\big(a; m(b)\big) = a + m(b);$
- Affine coupling layer[3, 4]: $g\big(a; m(b)\big) = a \odot \exp s(b) + t(b)$

# Affine coupling layer: NN

- Нелинейное отображение, например, shallow CNN, как в ResNet



# Комбинирование coupling layers

Проблема: некоторые компоненты $x$ остаются неизменными

Решения:

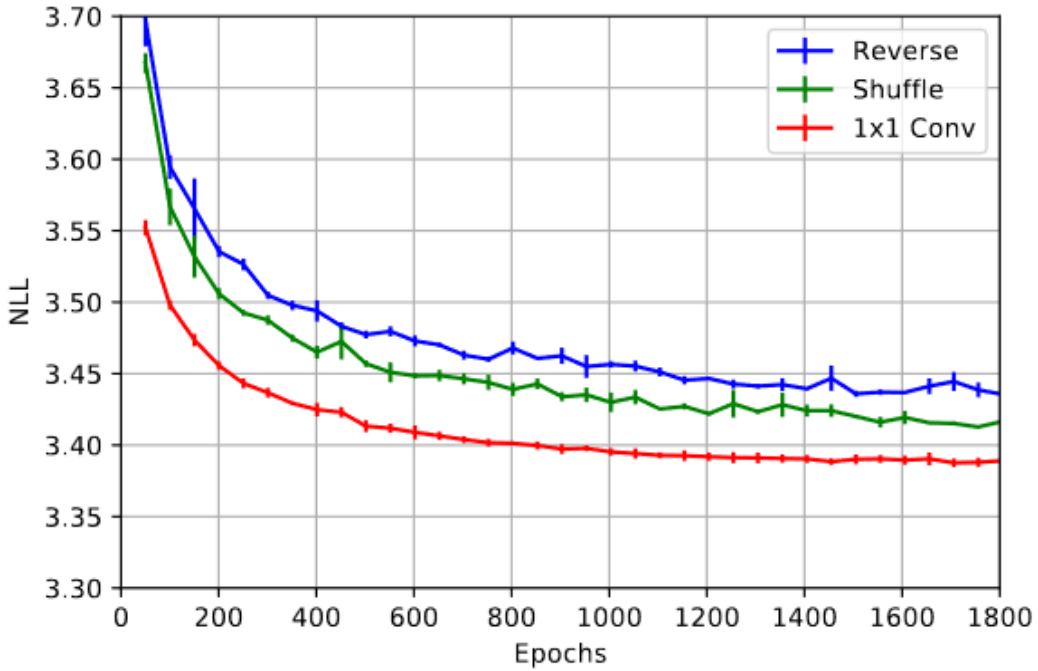
- Менять местами $I_1$ и $I_2$ (эквивалентно обратному порядку каналов);
- Случайная (фиксированная) перестановка каналов
- 1x1 invertible convolutions

# Invertible 1x1 convs

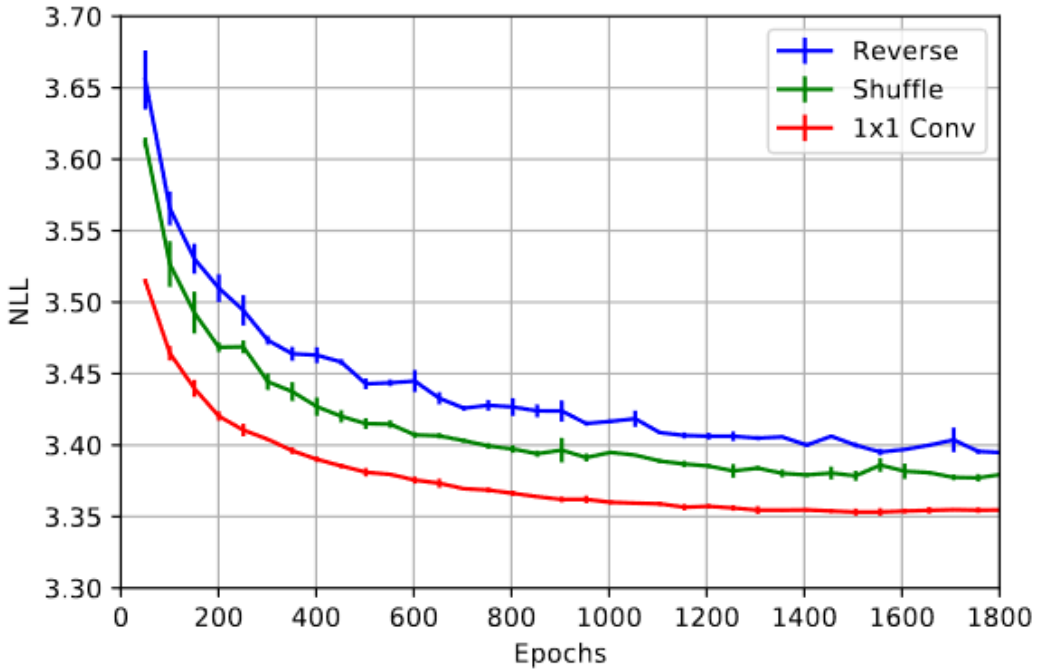
- Обобщение перестановок, когда число $n_{in\ channels} = n_{out\ channels}$
- Пусть $x$ – тензор размера $h \times w \times c$, $W$ – матрица весов свертки размера $c \times c$

$$\log \left| \det \left( \frac{dconv2D(x; W)}{dx} \right) \right| = h \cdot w \cdot \log|\det(W)|$$

# Сравнение подходов




(a) Additive coupling.



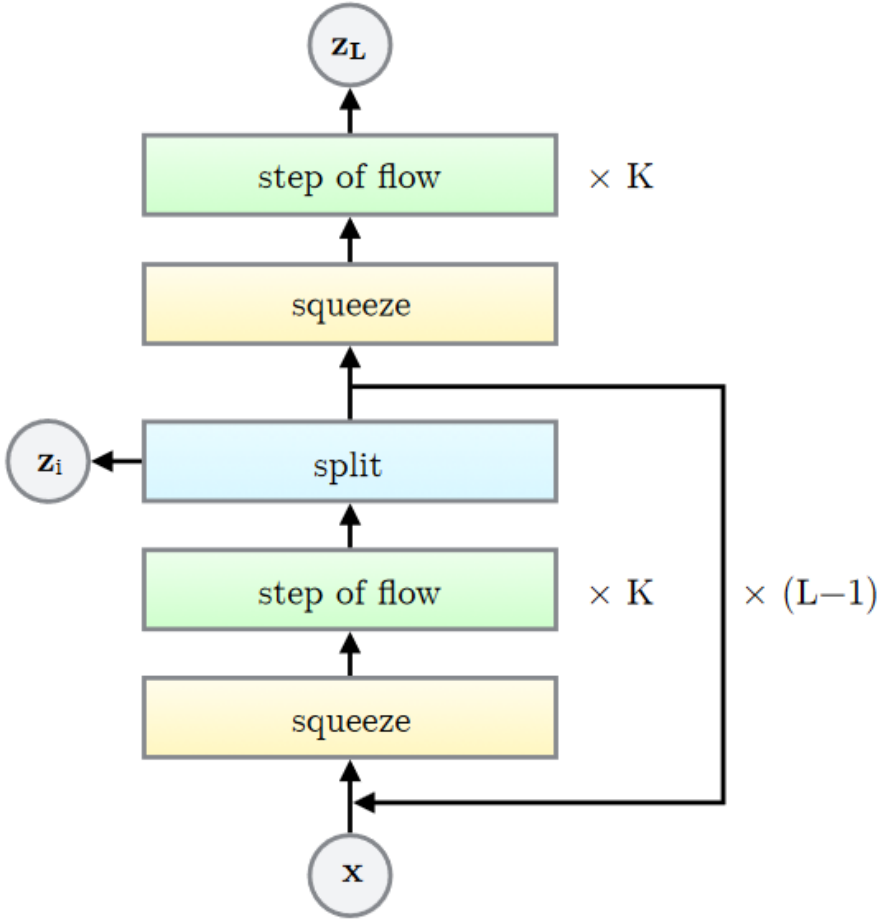

(b) Affine coupling.

# Суммарно о каждой операции

| Description | Function | Reverse Function | Log-determinant |
|---|---|---|---|
| Actnorm.<br>See Section 3.1. | $\forall i,j : \mathbf{y}_{i,j} = \mathbf{s} \odot \mathbf{x}_{i,j} + \mathbf{b}$ | $\forall i,j : \mathbf{x}_{i,j} = (\mathbf{y}_{i,j} - \mathbf{b})/\mathbf{s}$ | $h \cdot w \cdot \mathtt{sum}(\log \lvert\mathbf{s}\rvert)$ |
| Invertible $1 \times 1$ convolution.<br>$\mathbf{W} : [c \times c]$.<br>See Section 3.2. | $\forall i,j : \mathbf{y}_{i,j} = \mathbf{W}\mathbf{x}_{i,j}$ | $\forall i,j : \mathbf{x}_{i,j} = \mathbf{W}^{-1}\mathbf{y}_{i,j}$ | $h \cdot w \cdot \log \lvert \det(\mathbf{W}) \rvert$<br>or<br>$h \cdot w \cdot \mathtt{sum}(\log \lvert\mathbf{s}\rvert)$<br>(see eq. (10)) |
| Affine coupling layer.<br>See Section 3.3 and<br>(Dinh et al., 2014) | $\mathbf{x}_a, \mathbf{x}_b = \mathtt{split}(\mathbf{x})$<br>$(\log \mathbf{s}, \mathbf{t}) = \mathtt{NN}(\mathbf{x}_b)$<br>$\mathbf{s} = \exp(\log \mathbf{s})$<br>$\mathbf{y}_a = \mathbf{s} \odot \mathbf{x}_a + \mathbf{t}$<br>$\mathbf{y}_b = \mathbf{x}_b$<br>$\mathbf{y} = \mathtt{concat}(\mathbf{y}_a, \mathbf{y}_b)$ | $\mathbf{y}_a, \mathbf{y}_b = \mathtt{split}(\mathbf{y})$<br>$(\log \mathbf{s}, \mathbf{t}) = \mathtt{NN}(\mathbf{y}_b)$<br>$\mathbf{s} = \exp(\log \mathbf{s})$<br>$\mathbf{x}_a = (\mathbf{y}_a - \mathbf{t})/\mathbf{s}$<br>$\mathbf{x}_b = \mathbf{y}_b$<br>$\mathbf{x} = \mathtt{concat}(\mathbf{x}_a, \mathbf{x}_b)$ | $\mathtt{sum}(\log(\lvert\mathbf{s}\rvert))$ |

# Multi-scale architecture[3]

- **Squeeze**: разделить изображение на блоки $2 \times 2 \times c$, reshape to $1 \times 1 \times 4c$, обмен пространственной информации на глубину каналов
- **Factor out**: Для экономии памяти и времени после каждого шага потока отделяем часть каналов, как «готовые»
- $z = (z_1, z_2, \ldots, z_L)$



# Результаты (bits per dim)

Table 2: Best results in bits per dimension of our model compared to RealNVP.

| Model | CIFAR-10 | ImageNet 32x32 | ImageNet 64x64 | LSUN (bedroom) | LSUN (tower) | LSUN (church outdoor) |
|---|---|---|---|---|---|---|
| RealNVP | 3.49 | 4.28 | 3.98 | 2.72 | 2.81 | 3.08 |
| Glow | **3.35** | **4.09** | **3.81** | **2.38** | **2.46** | **2.67** |

# Линейная интерполяция

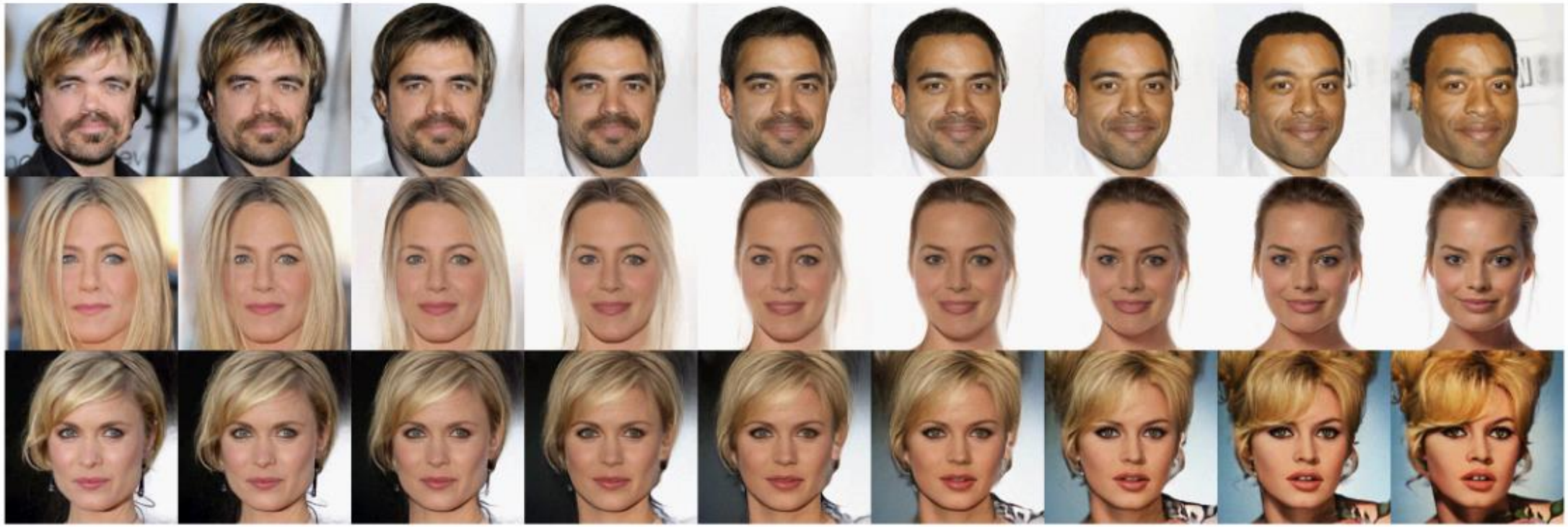
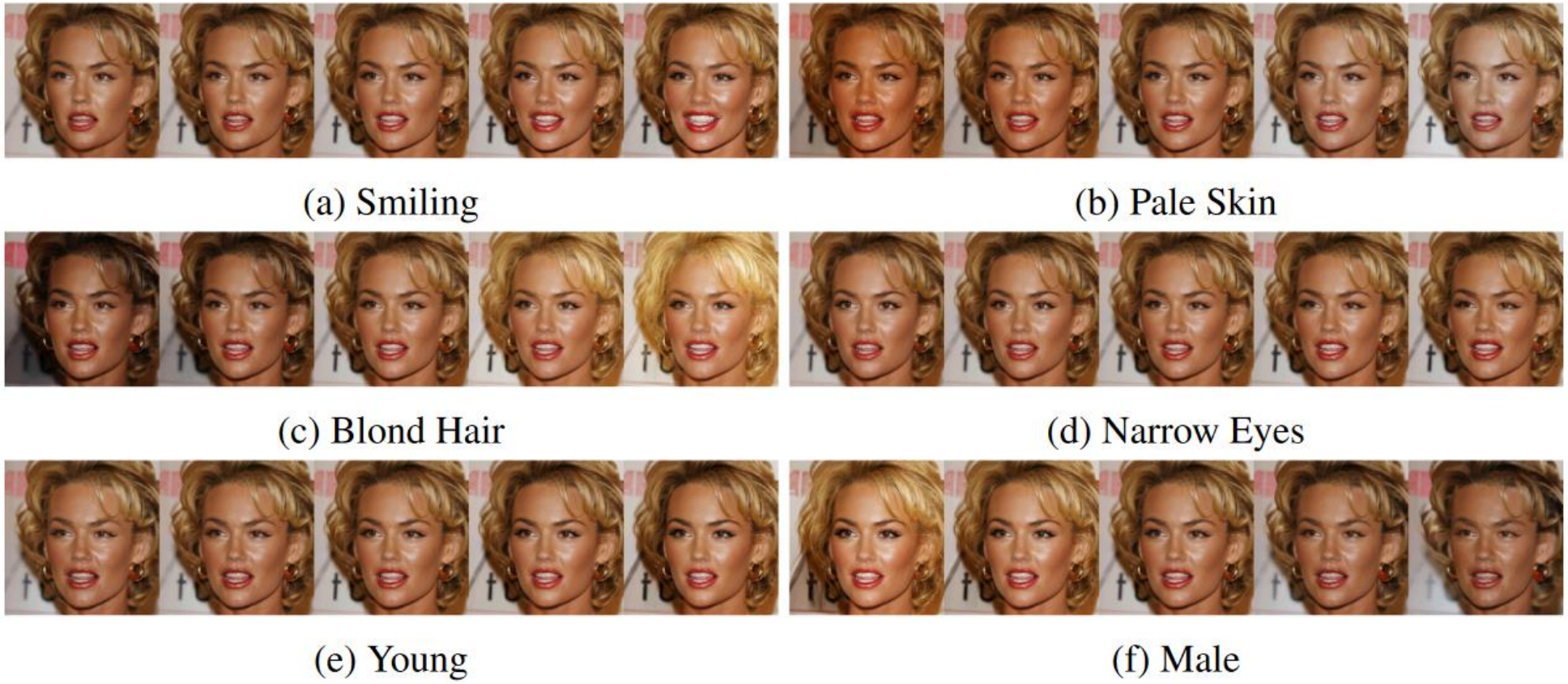


Figure 5: Linear interpolation in latent space between real images

# Изменение атрибутов



(a) Smiling (b) Pale Skin

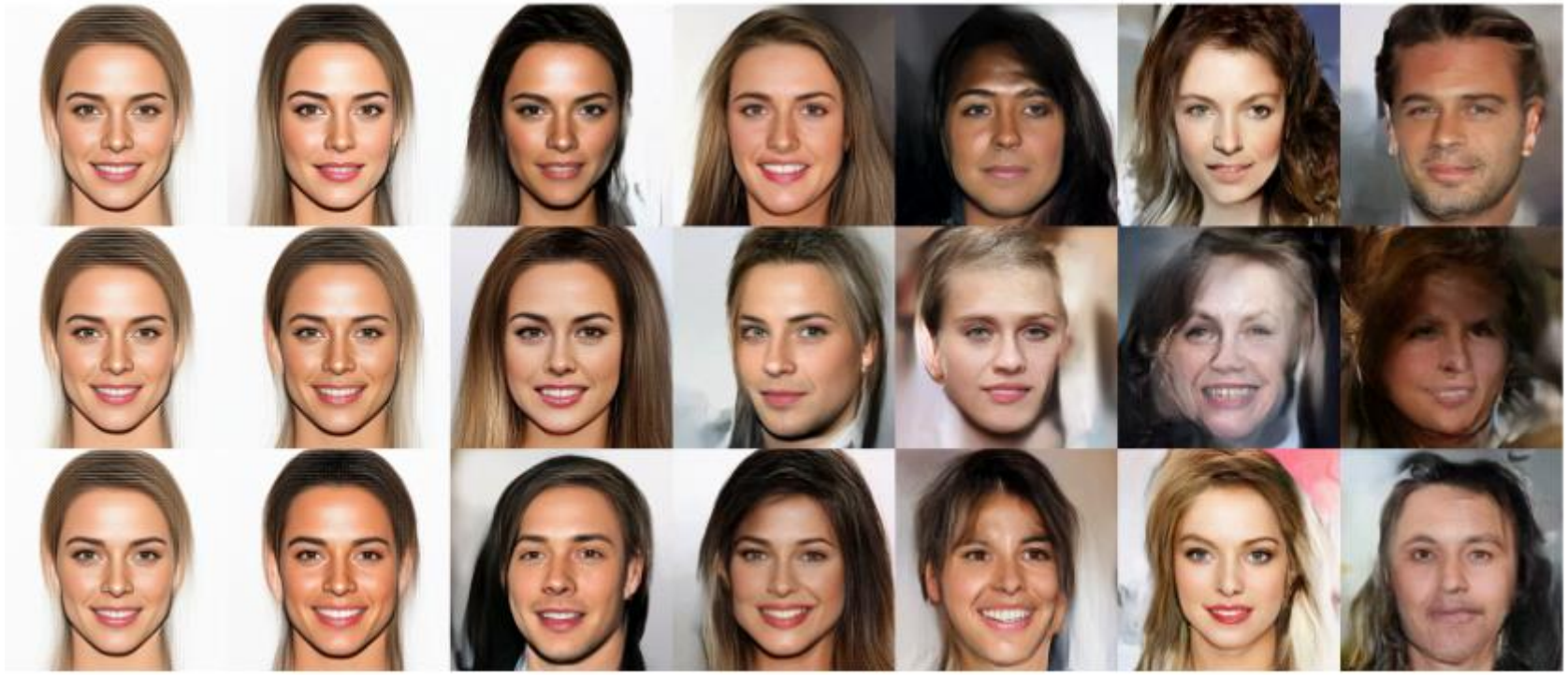
(c) Blond Hair (d) Narrow Eyes

(e) Young (f) Male

Figure 6: Manipulation of attributes of a face. Each row is made by interpolating the latent code of an image along a vector corresponding to the attribute, with the middle image being the original image

# LSUN bedrooms сэмплы

# Влияние температуры ($p_{X,T}(x) \propto p_X^{T^2}(x)$)



# Влияние глубины модели

# References

[1] Gomez, A. N., Ren, M., Urtasun, R., and Grosse, R. B. (2017). The reversible residual network: Backpropagation without storing activations. https://arxiv.org/abs/1707.04585

[2] Dinh, L., Krueger, D., and Bengio, Y. (2014). Nice: non-linear independent components estimation. https://arxiv.org/abs/1410.8516

[3] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using Real NVP. https://arxiv.org/abs/1605.08803

[4] Kingma, D.P., Dhariwal, P. (2018). Glow: Generative Flow with Invertible 1x1 Convolutions. https://arxiv.org/abs/1807.03039v2