МИСИС
УНИВЕРСИТЕТ

Генеративные модели. Автокодировщики

Корнет Мария Евгеньевна
старший преподаватель кафедры
Инженерная кибернетика

Модели глубокого обучения

Тип обучения	Примеры моделей	Типичные задачи
Обучение с учителем (Supervised Learning)	- CNN- RNN, LSTM, GRU- Transformer (BERT, T5)- MLP- ResNet, EfficientNet	- Классификация изображений и текста- Регрессия- Распознавание речи- Предсказание временных рядов
Обучение без учителя (Unsupervised Learning)	- Autoencoder- VAE- GAN- Self-supervised Transformers- Deep Clustering	- Сжатие данных- Генерация изображений или текста- Обнаружение аномалий- Кластеризация- Поиск скрытых структур
Полуобучение (Semi Supervised Learning)	- FixMatch- Pseudo-labeling- Consistency models	- Классификация с малым числом меток- Обучение на частично размеченных данных
Обучение с подкреплением (Reinforcement Learning)	- DQN- PPO- A3C- AlphaGo- RLHF (ChatGPT)	- Игры (Go, Atari)- Робототехника- Диалоговые агенты- Управление

Обучение без учителя

<table>
<tr><th colspan="4">Обучение без учителя
Unsupervised Learning</th></tr>
<tr><th>Невероятностные модели</th><th colspan="3">Вероятностные (генеративные) модели</th></tr>
<tr><td rowspan="4">• Sparse Coding
• Autoencoders
• k-means, …</td><th colspan="2">Плотность в явном виде (explicit density)</th><th>Плотность в неявном виде (implicit density)</th></tr>
<tr><th>Tractable Models</th><th colspan="2">Approximate density</th><td rowspan="3">• GAN
• Momet Matching Networks

Моделируем процесс сэмплирования, а не плотность</td></tr>
<tr><td rowspan="2">• Fully observed Belief Nets
• NADE
• MADE
• PixelRNN
• nonlinear ICA</td><th>Variational</th><th>Markov Chain</th></tr>
<tr><td>• VAE</td><td>• Boltzmann Machines
• Helmholtz Machines</td></tr>
</table>

Дьяконов А. https://alexanderdyakonov.wordpress.com/

Постановка задачи

Общая задача:

Имея выборку данных из какого-то неизвестного распределения *q(x)*, задача состоит в том, чтобы восстановить или аппроксимировать это самое распределение.

Задачи вероятностного моделирования: $p(y \mid x, \theta)$, где x – входные данные, y – целевая метка, θ – параметры модели. Такие модели называют **дискриминативные**.

Чтобы получить полное совместное распределение:

$$p(y, x \mid \theta) = p(y \mid x, \theta)\ p(x \mid \theta)$$

нам не хватает знания о том, как устроены сами данные, то есть о распределении $p(y \mid x, \theta)$. Именно его изучением и построением занимается **генеративное моделирование**.

Постановка задачи

Дана выборка $\{x_i\}_{i=1}^N$ одинаково распределенных случайных величин $x_i \sim q(x_i)$, $q(\cdot)$ - неизвестное распределение.

Требуется построить параметрическую модель $p\_\theta(\cdot)$, из которой можно *генерировать (сэмплировать)* новые объекты, следуя распределению $q(x)$.

Предполагается, что сэмплы, полученные из "хорошей" вероятностной модели, будут:

- Качественные, то есть получаемые случайные величины будут иметь распределение похожее на $q(\cdot)$;
- Разнообразные, то есть будут учтены разные моды;

- Вычислительно эффективные.

Генеративные модели

Генеративные модели — это такие модели, которые **учатся описывать распределение данных**, чтобы уметь:

- **Генерировать новые примеры**, похожие на те, что были в обучающем наборе.
- **Оценивать правдоподобие** новых данных (насколько они похожи на обучающие).

Подход 1: **Максимизация правдоподобия**

$$\prod_{x \in \text{train}} p_{\text{model}}(x; \theta) \to \max$$

Мы хотим найти параметры модели θ, которые **максимизируют вероятность (правдоподобие)** всех наблюдаемых данных.

Это то же самое, что **максимизировать логарифм** : $$\sum_{x \in \text{train}} \log p_{\text{model}}(x; \theta) \to \max$$

Генеративные модели

Подход 2:
Цель — добиться, чтобы **распределение на модели стало как можно ближе к истинному (в данных)**.

KL-дивергенция (от англ. *Kullback–Leibler divergence*) — это мера «расхождения» между двумя вероятностными распределениями.

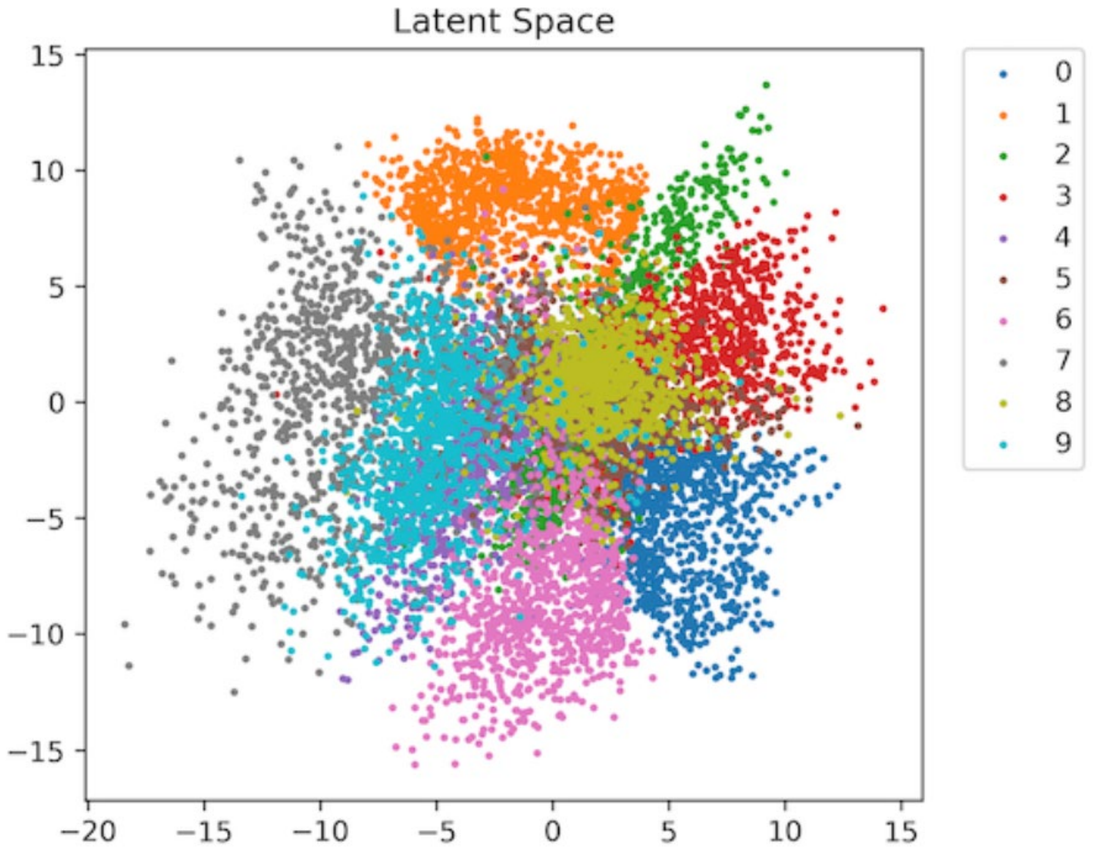
$$D_{KL}(P \parallel Q) = \int P(x) \log \frac{P(x)}{Q(x)}\, dx$$

Минимизация KL-дивергенции — это **эквивалент** максимизации правдоподобия:

$$D_{\mathrm{KL}}(p_{\mathrm{data}} \parallel p_{\mathrm{model}}) = \int p_{\mathrm{data}}(x) \ln \frac{p_{\mathrm{model}}(x)}{p_{\mathrm{data}}(x)}\, dx \to \min$$

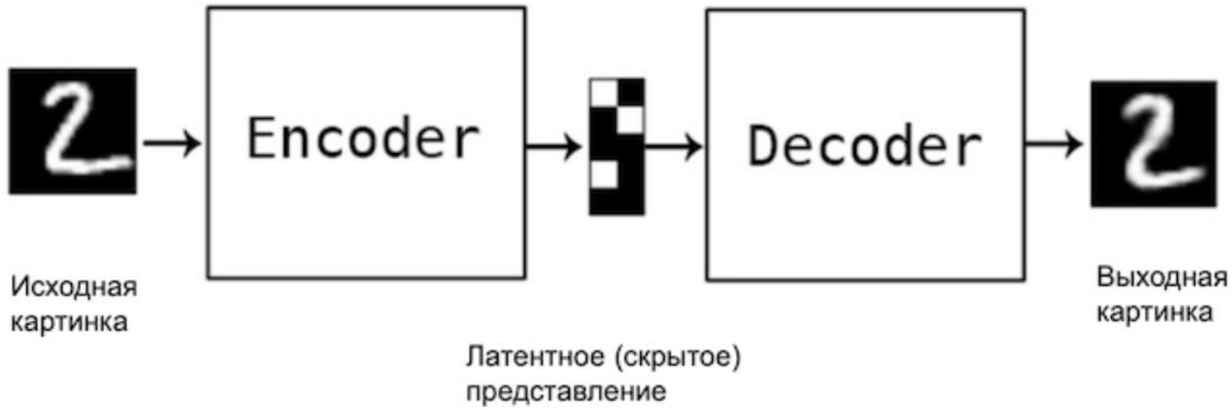
Латентное пространство

Латентное пространство — это внутреннее представление объектов в виде **векторов**, которые захватывают **семантику** и **структуру** данных, но не совпадают напрямую с исходными пикселями, словами или аудиосигналами.



Автоэнкодер

Автоэнкодер — это нейронная сеть, состоящая из двух симметричных частей:



1.Энкодер (Encoder): Принимает на вход исходные данные (изображение, временной ряд и т.д.) и сжимает их до компактного латентного представления, выделяя самые важные признаки.

2.Декодер (Decoder): Принимает это латентное представление и пытается восстановить из него исходные данные как можно точнее.

Автоэнкодер

Автоэнкодер (AE) обучается в режиме self-supervised learning. Это значит, что для его обучения не нужно иметь разметку данных.

Применение:

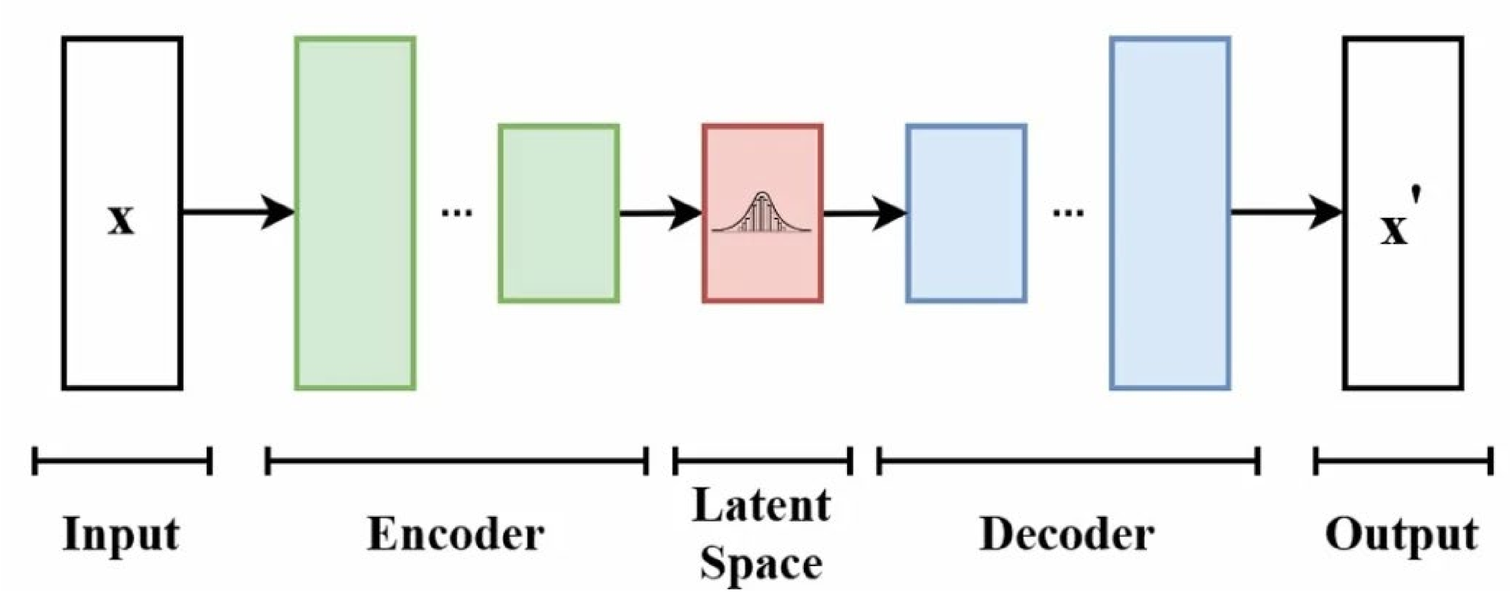
- Сжатие и хранение информации.
- Кластеризация объектов на основе их латентных представлений.
- Поиск похожих объектов (например, изображений).
- Генерация новых объектов (например, изображений).
- Поиск аномалий.

Ключевая идея для детекции аномалий:
Энкодер учится распределению «нормальных» данных, на которых он тренируется. Для типичных, похожих на обучающие, данных он сможет построить хорошее латентное представление и успешно их восстановить (loss будет small).
Если же на вход подается **аномалия** (выброс, который сеть раньше не видела), латентное представление окажется некорректным, и декодер плохо его восстановит. **Ошибка реконструкции (loss) будет высокой.**

Вариационный автоэнкодер

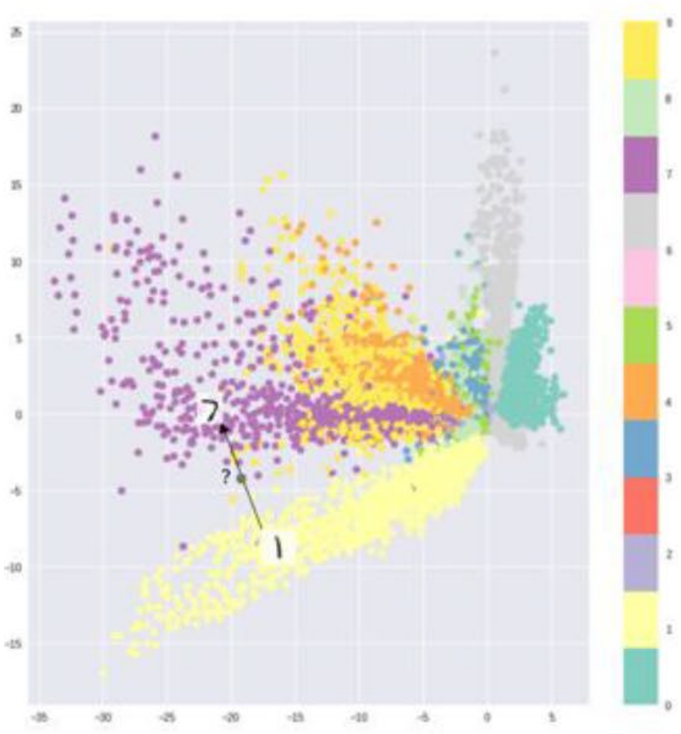
Вариационный автоэнкодер (Variational Autoencoder, VAE).
Основная идея: Обучать автокодировщик так, чтобы скрытые переменные имели какое-то распределение.

Энкодер — принимает данные и кодирует их в параметры распределения латентного вектора: среднее (μ) и дисперсию (σ^2).
Декодер — получает выборку из распределения $q(\cdot)$ и восстанавливает данные обратно.

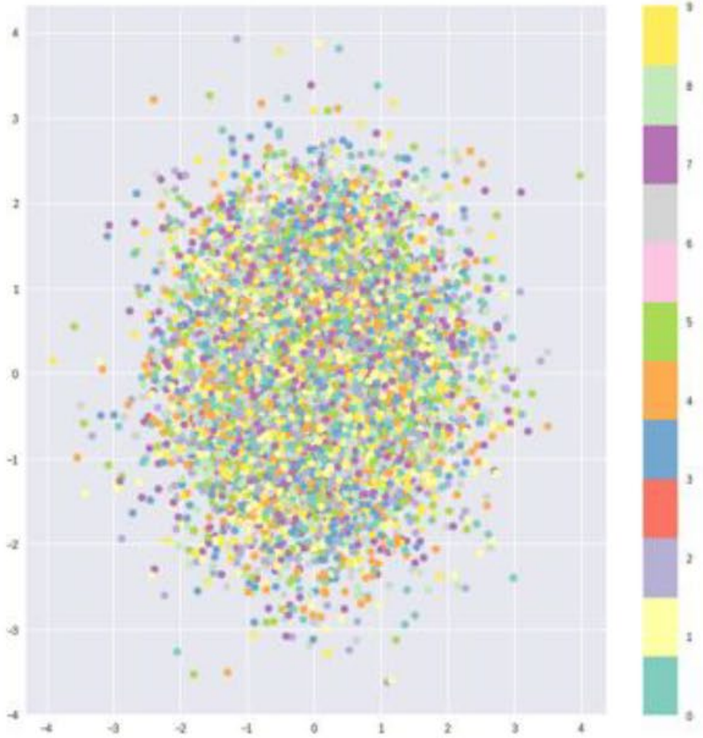


Латентные пространства

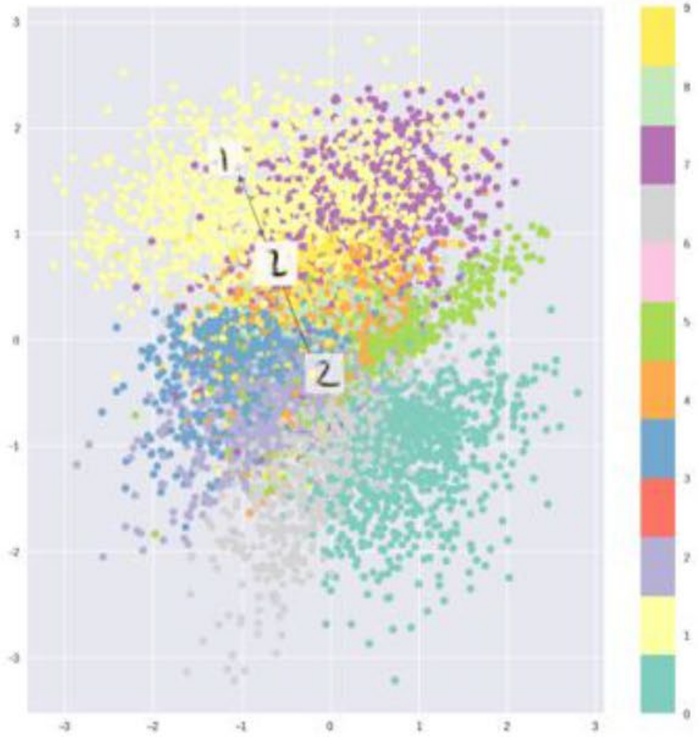
Only reconstruction loss



Only KL divergence

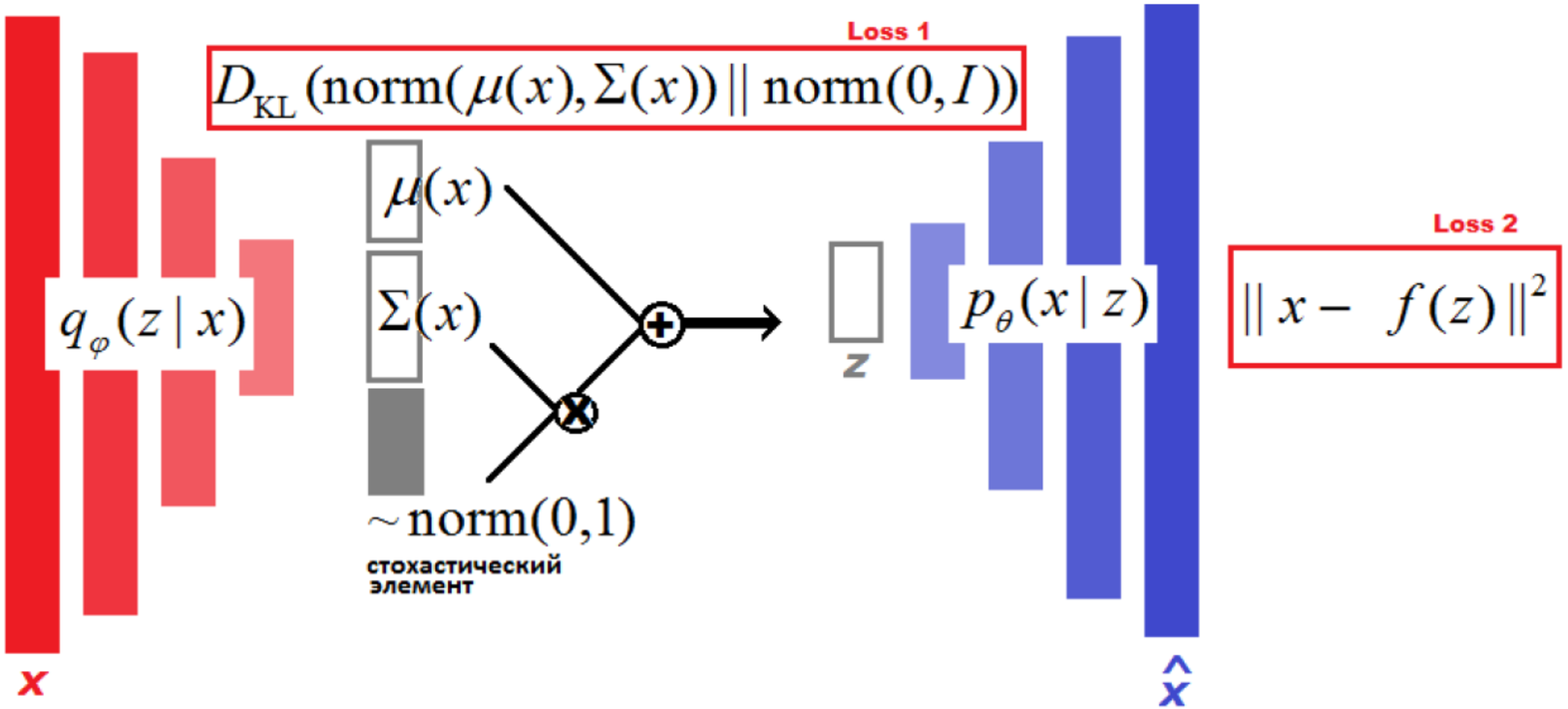


Combination



Ключевая техника VAE

Reparametrizaton trick: Как взять случайную выборку из распределения N(μ, σ²), чтобы этот процесс был дифференцируемым и модель можно было обучить градиентным спуском?



$$\mathrm{norm}(\mu,\Sigma) \sim \mu + \Sigma \cdot \mathrm{norm}(0,1)$$

Вариационный автоэнкодер

Функция потерь:

$$\mathcal{L}_{\text{CVAE}}(x, y) = \underbrace{\mathbb{E}_{q(z|x,y)}[\log p(x \mid z, y)]}_{\text{реконструкция}} - \underbrace{D_{KL}(q(z \mid x, y) \,\|\, p(z \mid y))}_{\text{KL-дивергенция}}$$

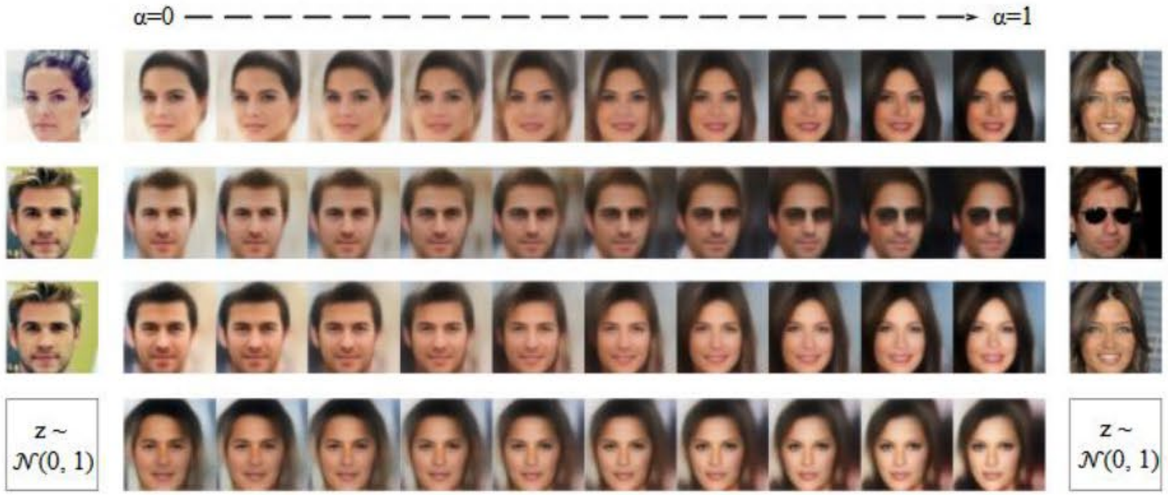


Figure 5. Linear interpolation for latent vector. Each row is the interpolation from left latent vector z_{left} to right latent vector z_{right}. e.g. $(1-\alpha)z_{left} + \alpha z_{right}$. The first row is the transition from a non-smiling woman to a smiling woman, the second row is the transition from a man without eyeglass to a man with eyeglass, the third row is the transition from a man to a woman, and the last row is the transition between two fake faces decoded from $z \sim \mathcal{N}(0, 1)$.

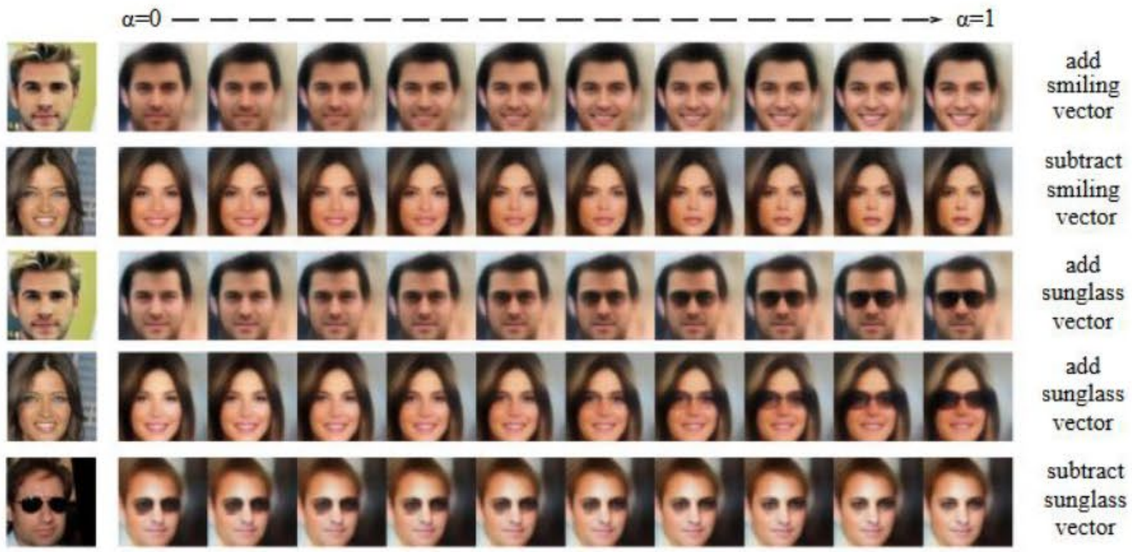
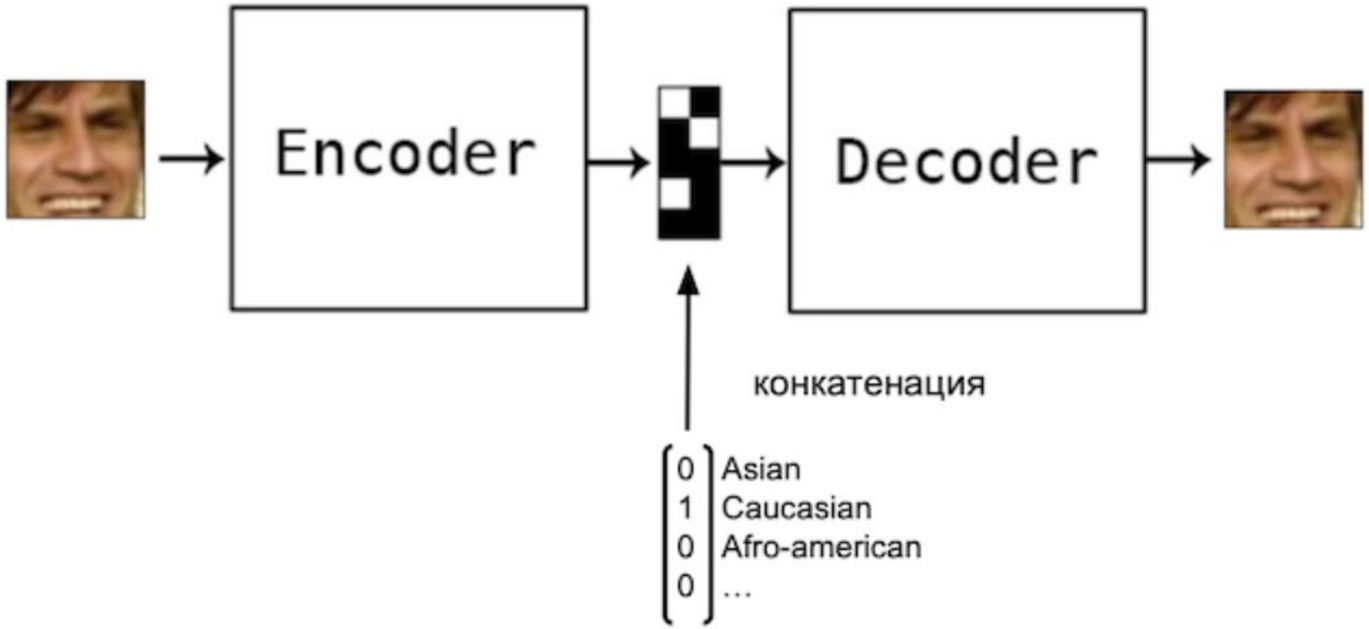


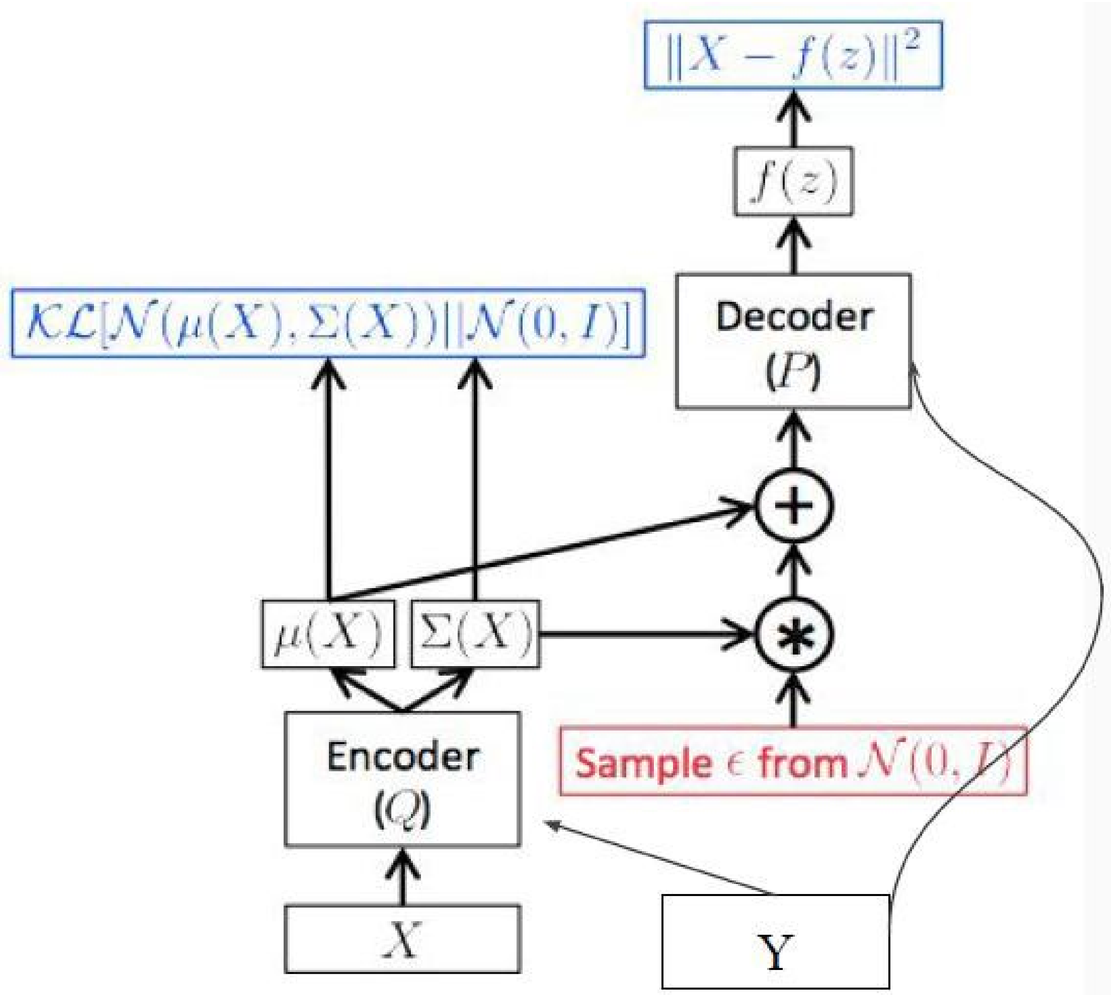
Figure 6. Vector arithmetic for visual attributes. Each row is the generated faces from latent vector z_{left} by adding or subtracting an attribute-specific vector, i.e., $z_{left} + \alpha\ z_{smiling}$, where $\alpha = 0, 0.1, \ldots, 1$. The first row is the transition by adding a smiling vector with a linear factor α from left to right, the second row is the transition by subtracting a smiling vector, the third and fourth row are the results by adding a eyeglass vector to the latent representation for a man and women, and the last row shows results by subtracting an eyeglass vector.

Conditional VAE

Conditional Variational Autoencoder (CVAE) — это расширение обычного VAE, которое позволяет **контролировать процесс генерации** с помощью **дополнительной информации (меток)**.



Conditional VAE



МИСИС
УНИВЕРСИТЕТ

Спасибо
за внимание!