



Metric Learning

Lecture 8

Stanislav Dereka

19.08.2022

План лекции

01

Что такое Metric Learning?

02

Метрики качества в
задачах Metric Learning

03

Обучение Metric Learning
моделей: contrastive
подход

04

Обучение Metric Learning
моделей:
классификационный
подход

План лекции

01

Что такое Metric Learning?

02

Метрики качества в
задачах Metric Learning

03

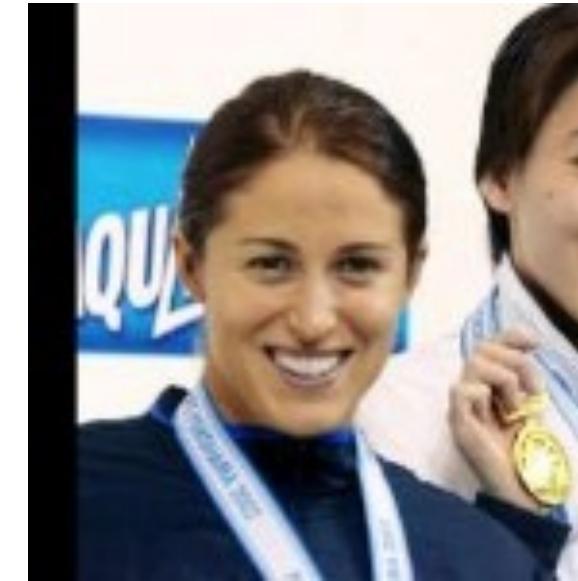
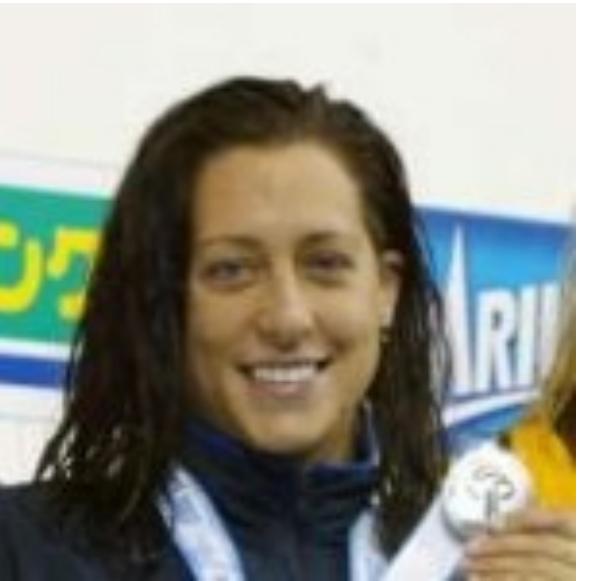
Обучение Metric Learning
моделей: contrastive
подход

04

Обучение Metric Learning
моделей:
классификационный
подход

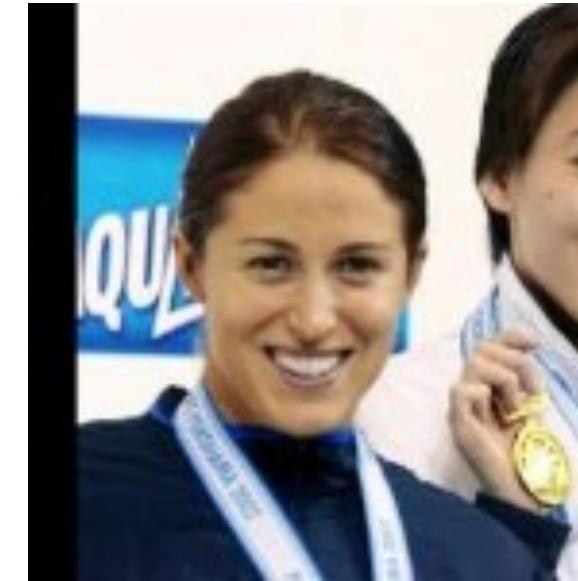
Face verification

Хотим создать модель, которая по двум фотографиям отвечает на вопрос: “Один и тот же человек на фотографиях?”



Face verification

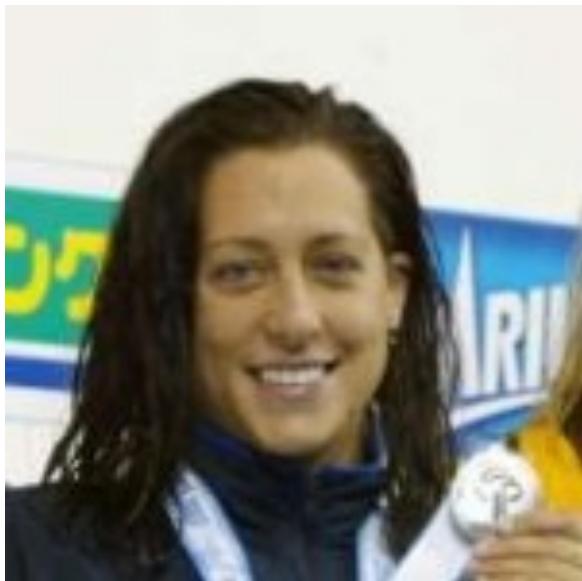
Хотим создать модель, которая по двум фотографиям отвечает на вопрос: “Один и тот же человек на фотографиях?”



Как бы вы решали такую задачу?

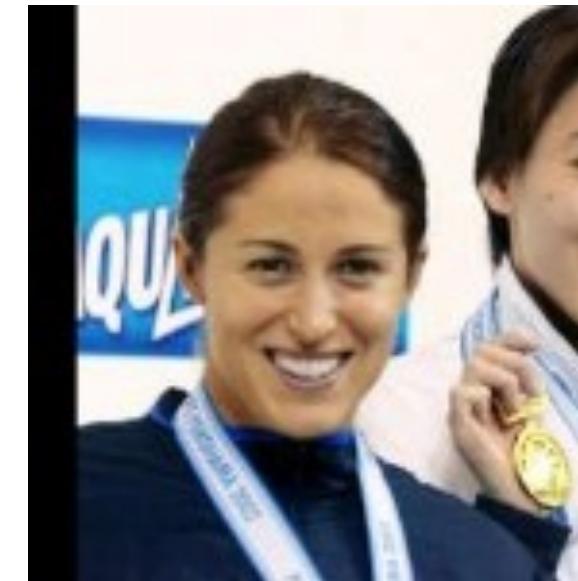
Face verification

Хотим создать модель, которая по двум фотографиям отвечает на вопрос: “Один и тот же человек на фотографиях?”



Вектор признаков
(эмбеддинг)

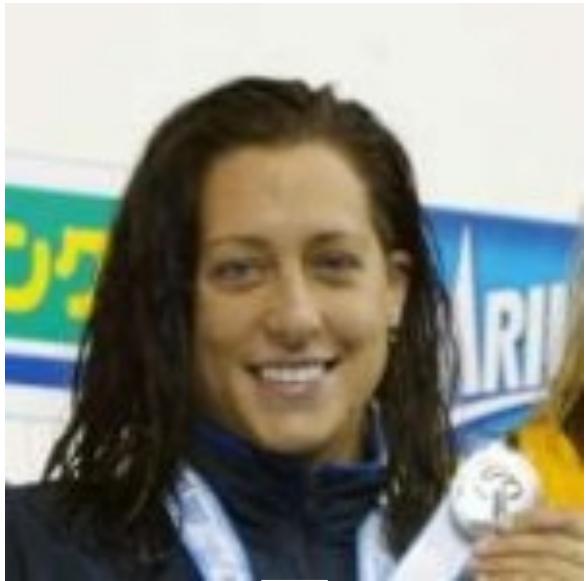
Расстояние между глаз Ширина рта



Расстояние между глаз Ширина рта

Face verification

Хотим создать модель, которая по двум фотографиям отвечает на вопрос: “Один и тот же человек на фотографиях?”



Вектор признаков
(эмбеддинг)

Расстояние между глаз Ширина рта

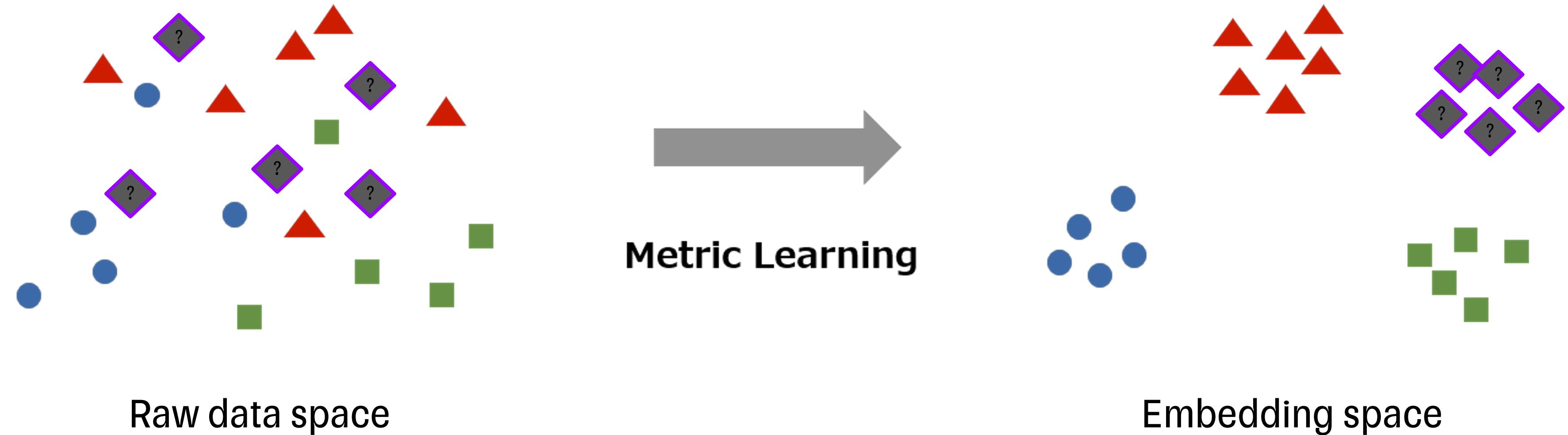


Расстояние между глаз Ширина рта

Требования к решению:

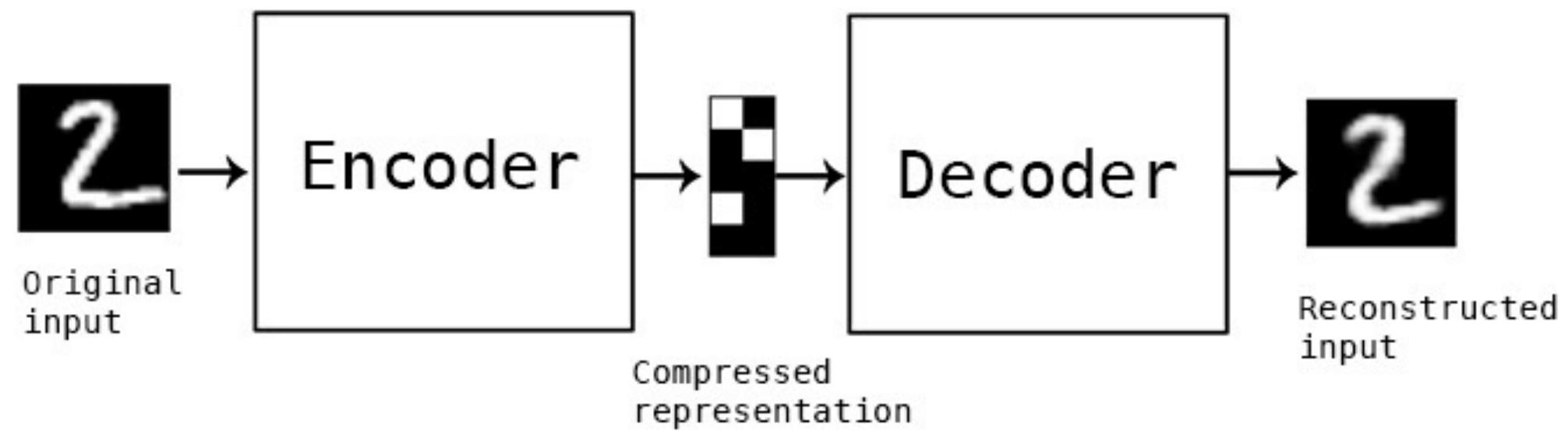
- Все изображения обрабатываются одинаково
- Поменьше признаков
- Возможность быстро сравнивать признаки

Metric learning approach



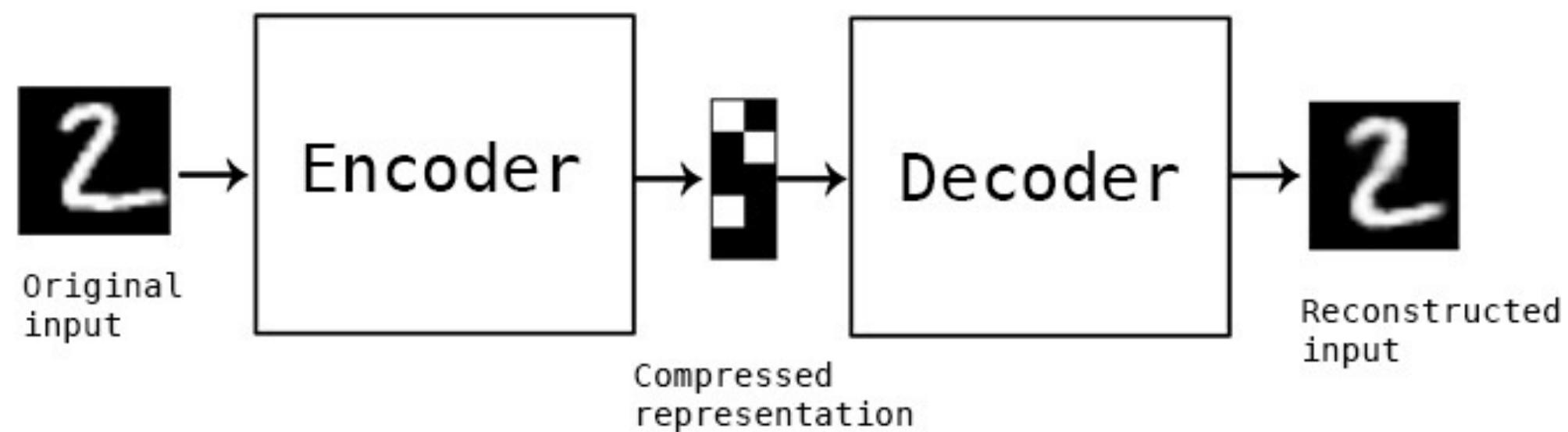
Похожее встречалось в курсе ранее

Латентное пространство в автоэнкодерах

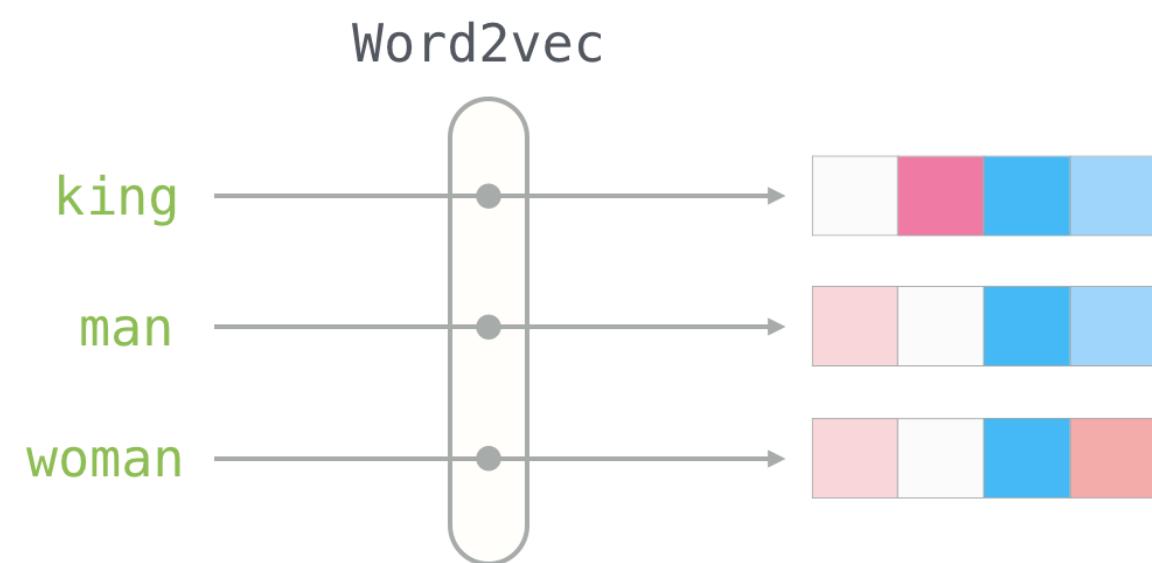


Похожее встречалось в курсе ранее

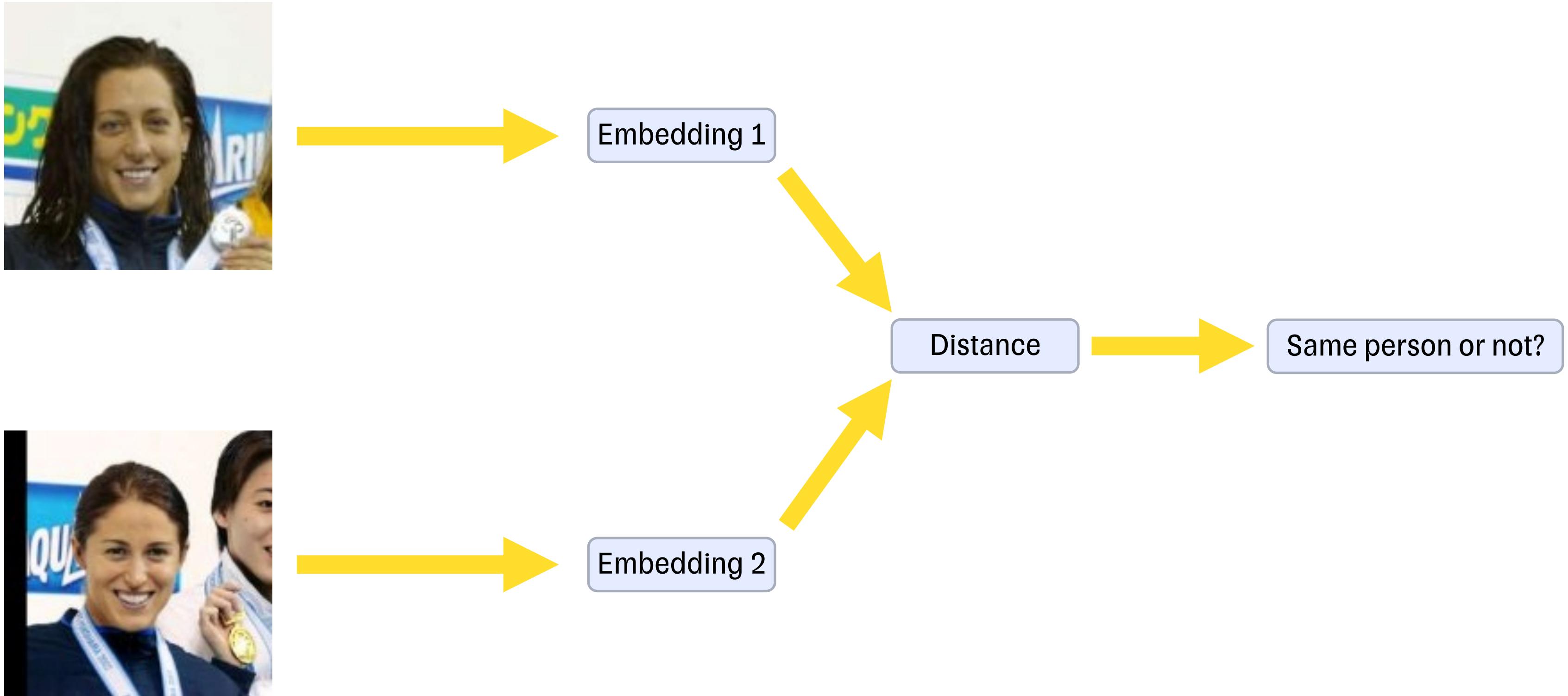
Латентное пространство в автоэнкодерах



Word2Vec



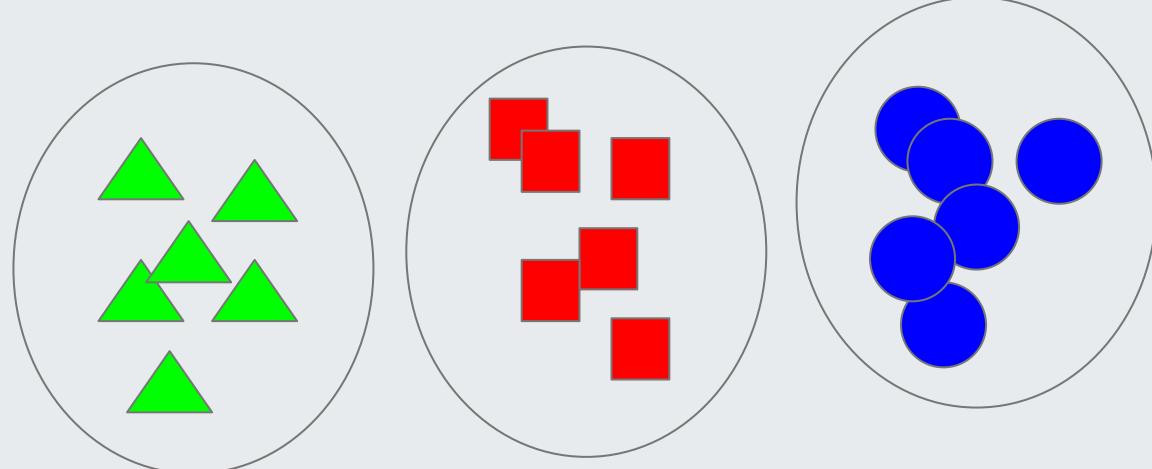
Face verification



Open-set classification

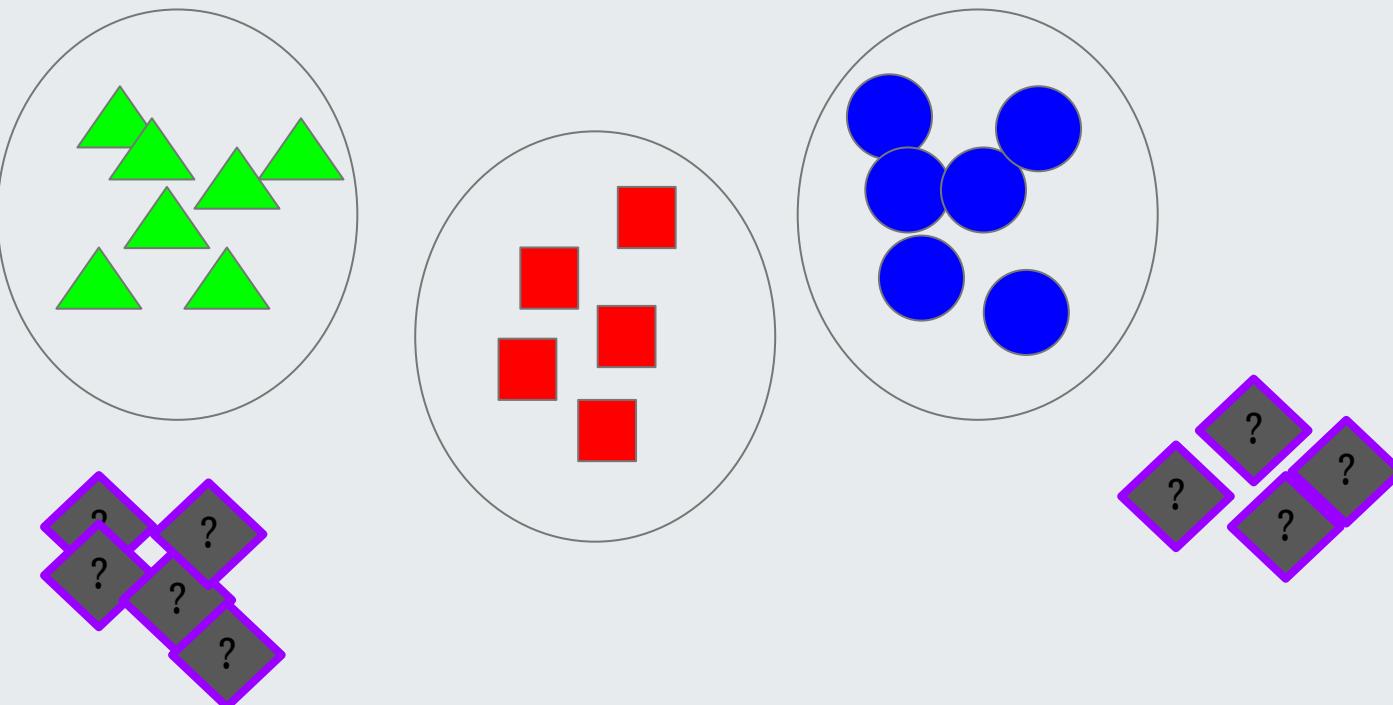
Closed-set classification

- Train on known classes
- Test and validate on known classes
- Infer on known classes

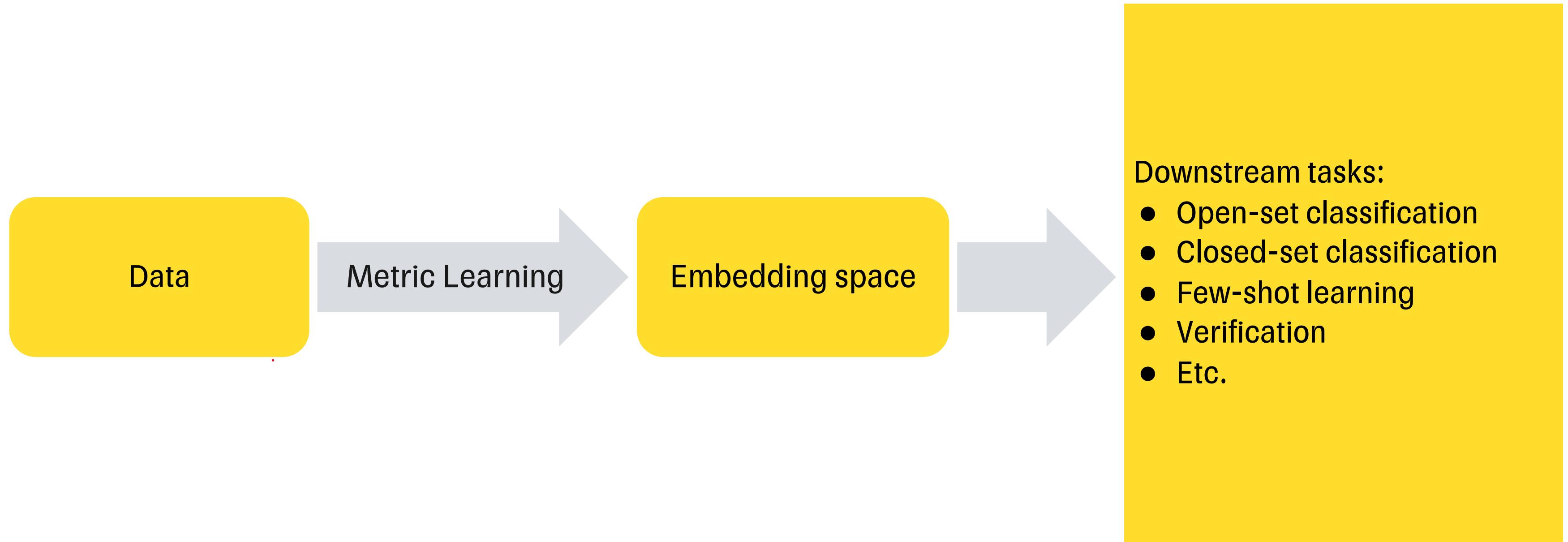


Open-set classification

- Train on known classes
- Test and validate on known + *unknown* classes
- Infer on known + *unknown* classes



Metric learning approach



План лекции

01

Что такое Metric Learning?

03

Обучение Metric Learning
моделей: contrastive
подход

02

Метрики качества в
задачах Metric Learning

04

Обучение Metric Learning
моделей:
классификационный
подход

Метрики качества

Что значит хорошее пространство эмбеддингов?

Recall@K

- Хорошо, когда ближайший сосед того же класса

Recall@K

- Для выбранного семпла считаются top K ближайших соседей в пространстве эмбеддингов

Recall@K

- Для выбранного семпла считаются top K ближайших соседей в пространстве эмбеддингов
- Для каждого соседа выставляется метка 0 - другой класс, 1 - тот же класс, что и у исходного семпла

Recall@K

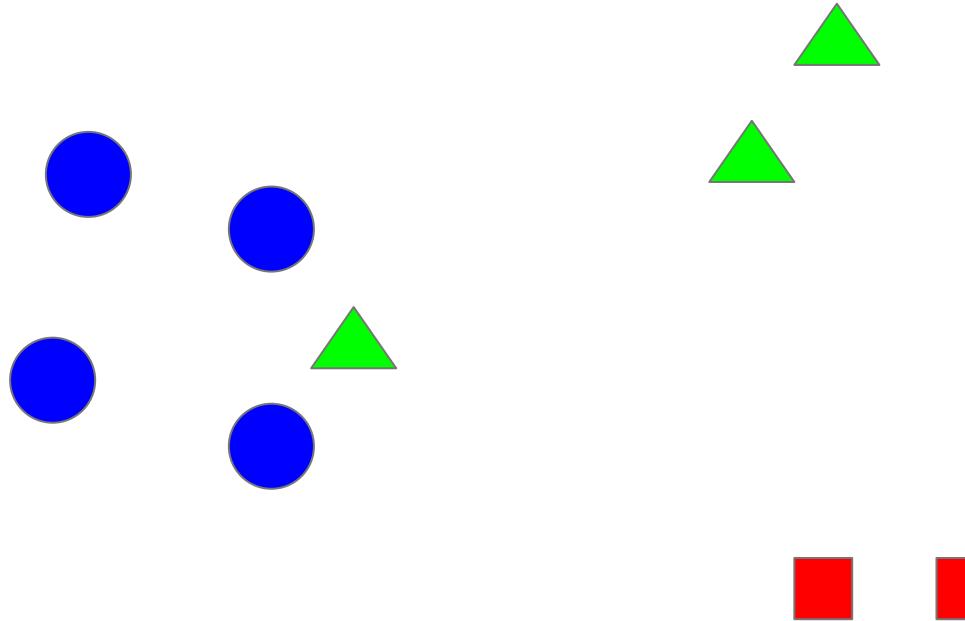
- Для выбранного семпла считаются top K ближайших соседей в пространстве эмбеддингов
- Для каждого соседа выставляется метка 0 - другой класс, 1 - тот же класс, что и у исходного семпла
- Для семпла $R@K = 1$, если есть хотя бы одна 1, $R@K = 0$ если все 0

Recall@K

- Для выбранного семпла считаются top K ближайших соседей в пространстве эмбеддингов
- Для каждого соседа выставляется метка 0 - другой класс, 1 - тот же класс, что и у исходного семпла
- Для семпла $R@K = 1$, если есть хотя бы одна 1, $R@K = 0$ если все 0
- Все посемпловые $R@K$ усредняются на всём тестовом датасете

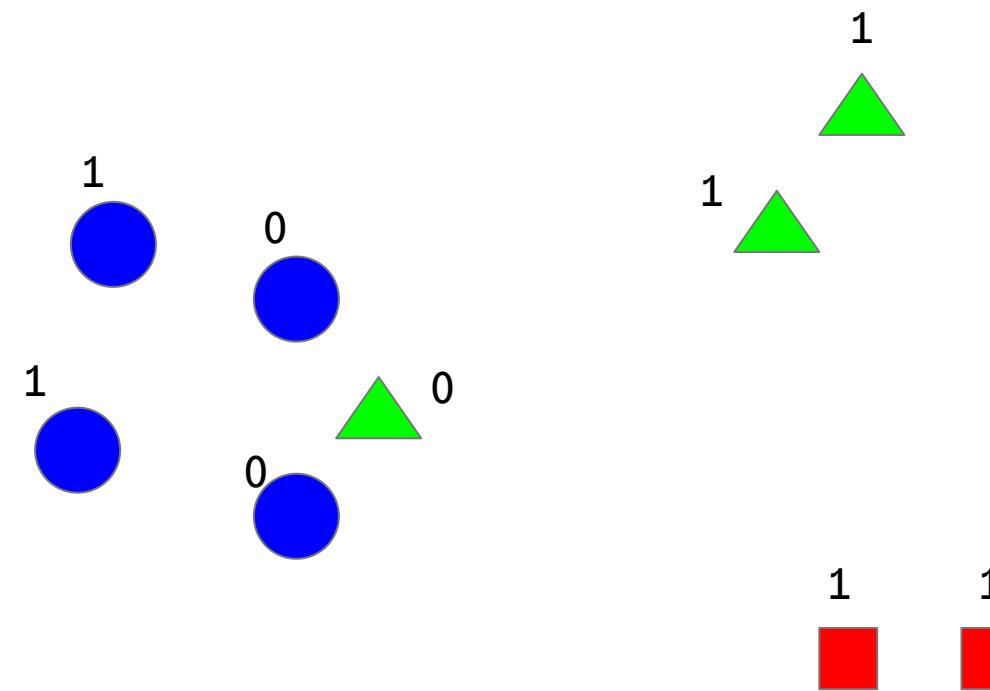
Recall@K

- Для выбранного семпла считаются top K ближайших соседей в пространстве эмбеддингов
- Для каждого соседа выставляется метка 0 - другой класс, 1 - тот же класс, что и у исходного семпла
- Для семпла $R@K = 1$, если есть хотя бы одна 1, $R@K = 0$ если все 0
- Все посемпловые $R@K$ усредняются на всём тестовом датасете



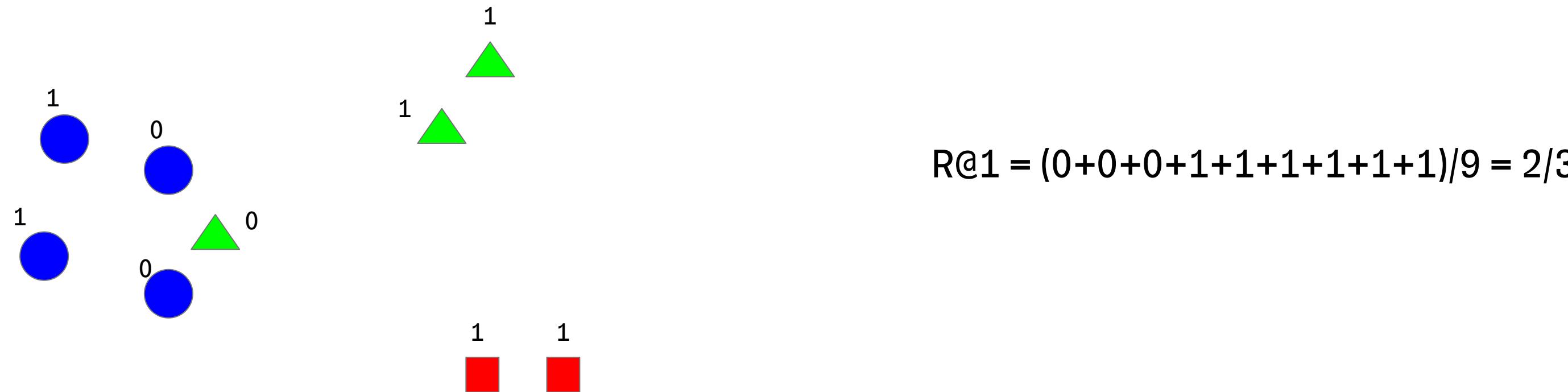
Recall@K

- Для выбранного семпла считаются top K ближайших соседей в пространстве эмбеддингов
- Для каждого соседа выставляется метка 0 - другой класс, 1 - тот же класс, что и у исходного семпла
- Для семпла $R@K = 1$, если есть хотя бы одна 1, $R@K = 0$ если все 0
- Все посемпловые $R@K$ усредняются на всём тестовом датасете



Recall@K

- Для выбранного семпла считаются top K ближайших соседей в пространстве эмбеддингов
- Для каждого соседа выставляется метка 0 - другой класс, 1 - тот же класс, что и у исходного семпла
- Для семпла $R@K = 1$, если есть хотя бы одна 1, $R@K = 0$ если все 0
- Все посемпловые $R@K$ усредняются на всём тестовом датасете



Метрики качества

Что значит хорошее пространство эмбеддингов?

F1 score

- Хорошо, когда при кластеризации полученные кластеры совпадают с исходными классами

F1 score

- Требует кластеризации

F1 score

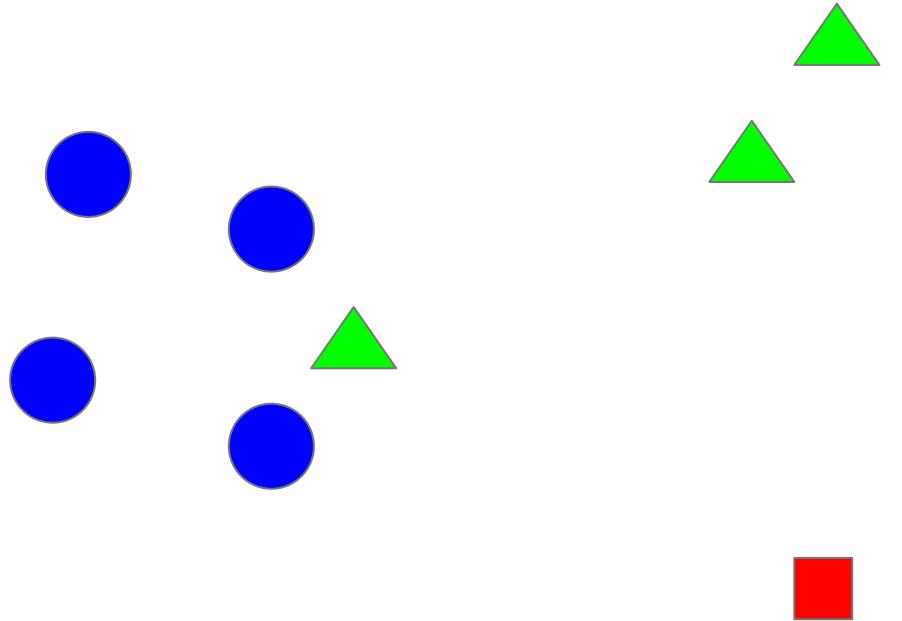
- Требует кластеризации
- Для пар в кластере выставляются метки 0 (разного класса) и 1 (одного класса)

F1 score

- Требует кластеризации
- Для пар в кластере выставляются метки 0 (разного класса) и 1 (одного класса)
- Далее считается сама метрика F1

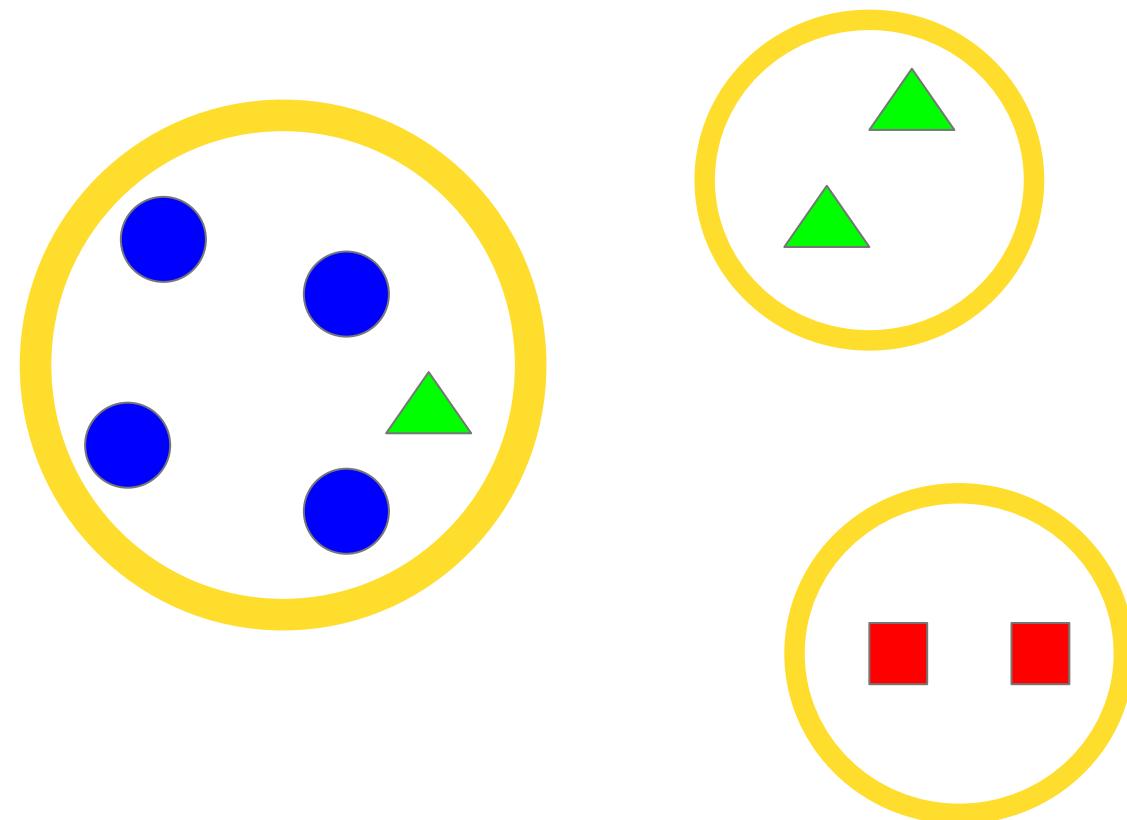
F1 score

- Требует кластеризации
- Для пар в кластере выставляются метки 0 (разного класса) и 1 (одного класса)
- Далее считается сама метрика F1

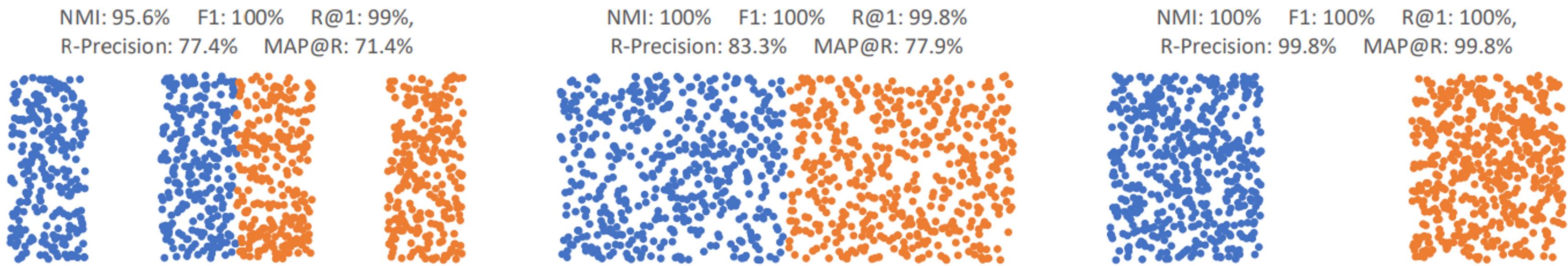


F1 score

- Требует кластеризации
- Для пар в кластере выставляются метки 0 (разного класса) и 1 (одного класса)
- Далее считается сама метрика F1



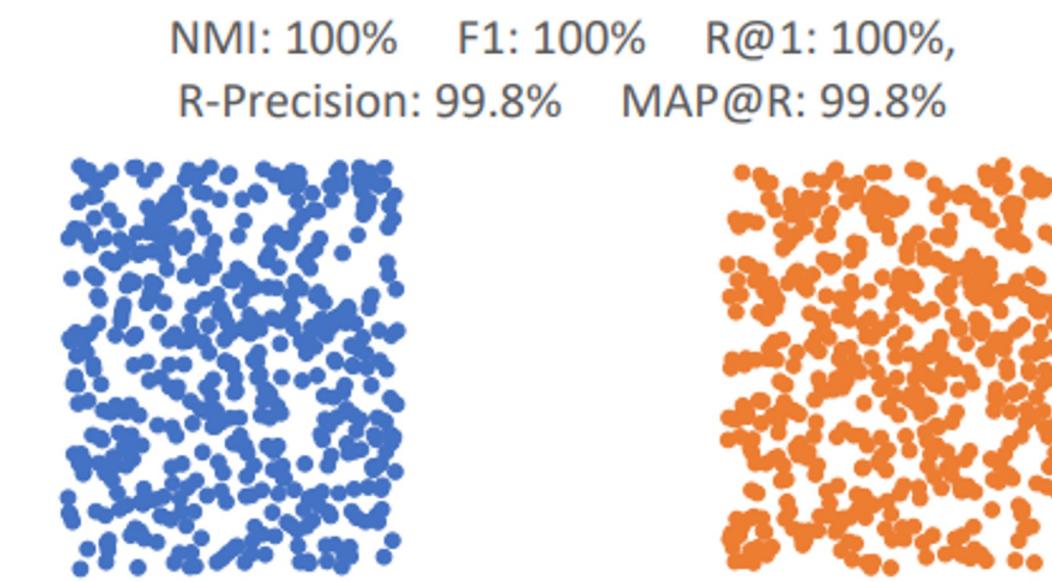
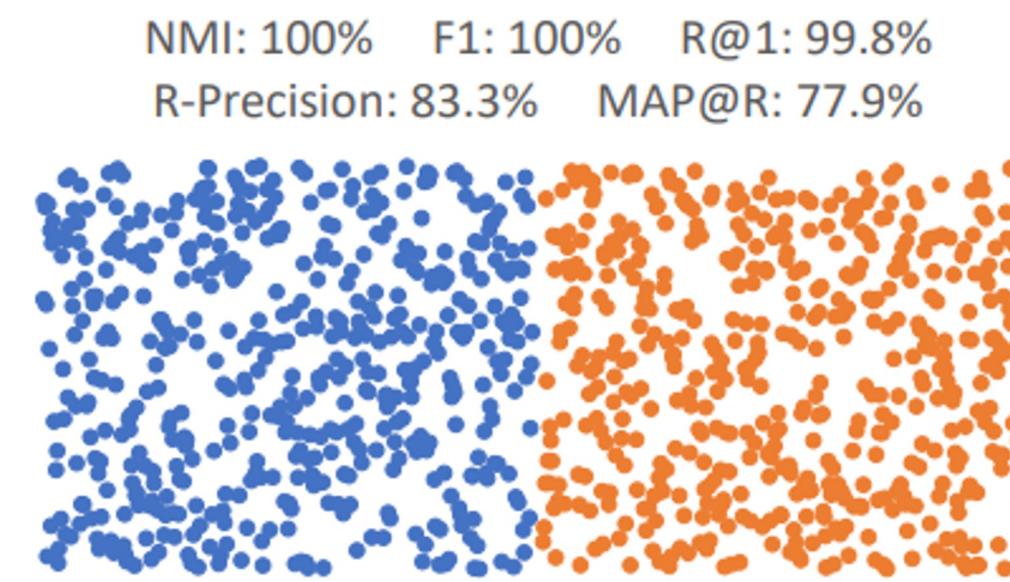
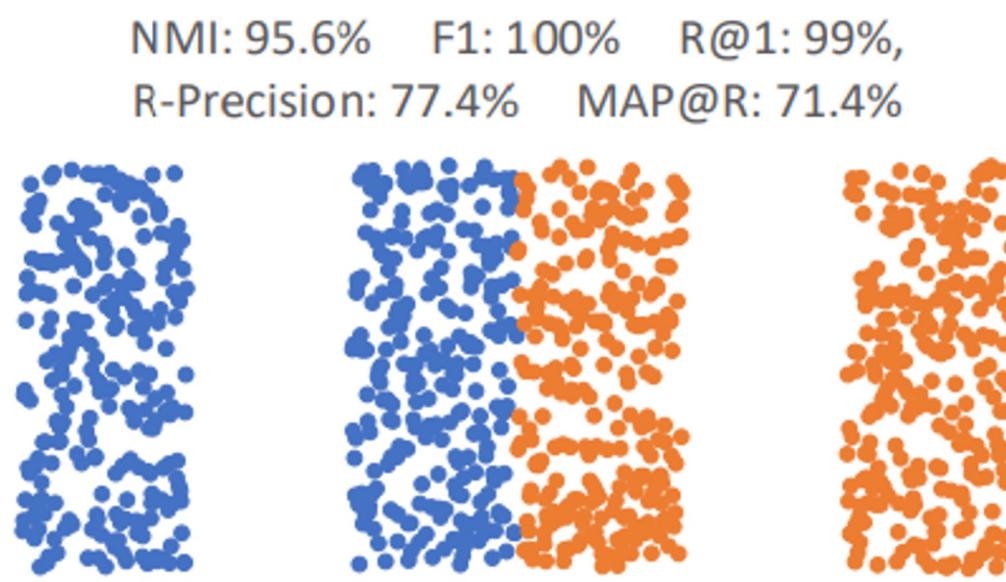
Проблемы R@K, F1



Проблемы:

- Зависимость от алгоритма кластеризации и его *random seed*
- Неинформативность (см картинку)

Проблемы R@K, F1



MAP@R

MAP@R

R-precision = r/K , К число ближайших соседей, из которых r того же класса, что и исходный сэмпл

MAP@R

R-precision = r/K, К число ближайших соседей, из которых r того же класса, что и исходный сэмпл

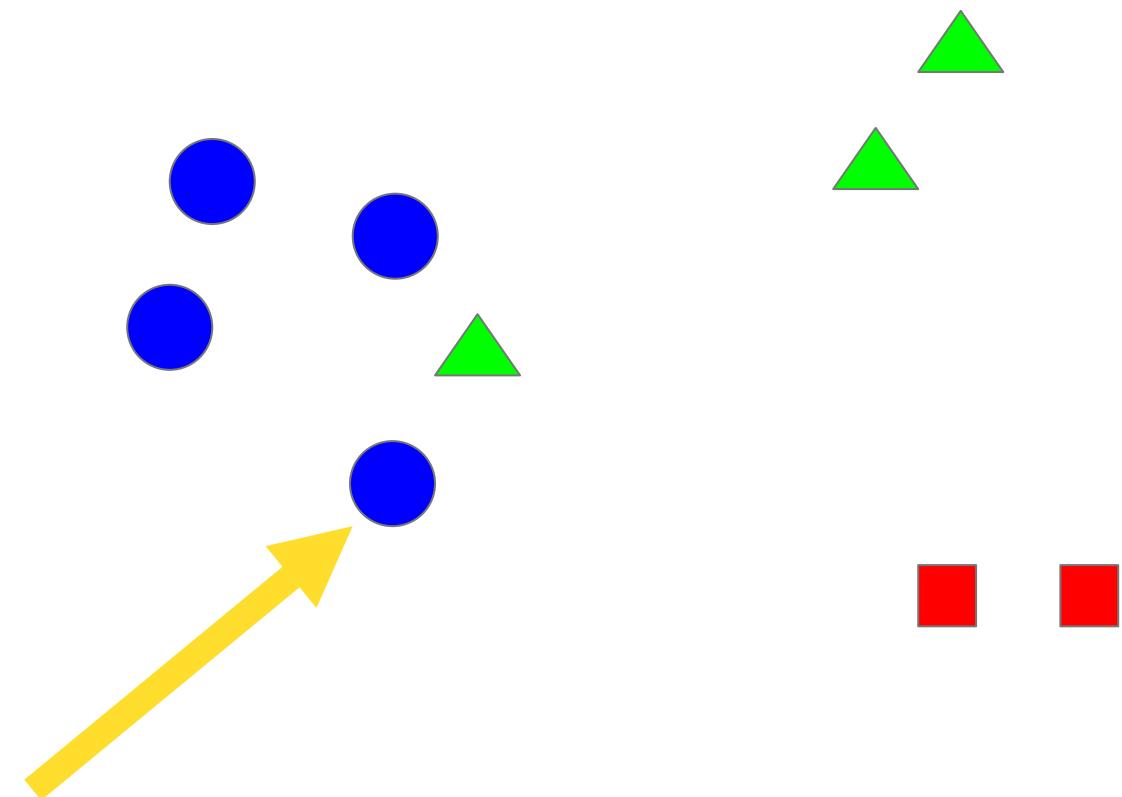
$$\text{MAP}@R = \frac{1}{R} \sum_{i=1}^R P(i)$$
$$P(i) = \begin{cases} \text{precision at } i, & \text{if the ith retrieval is correct} \\ 0, & \text{otherwise} \end{cases}$$

R = class_size - 1

MAP@R

$$\text{MAP}@R = \frac{1}{R} \sum_{i=1}^R P(i)$$

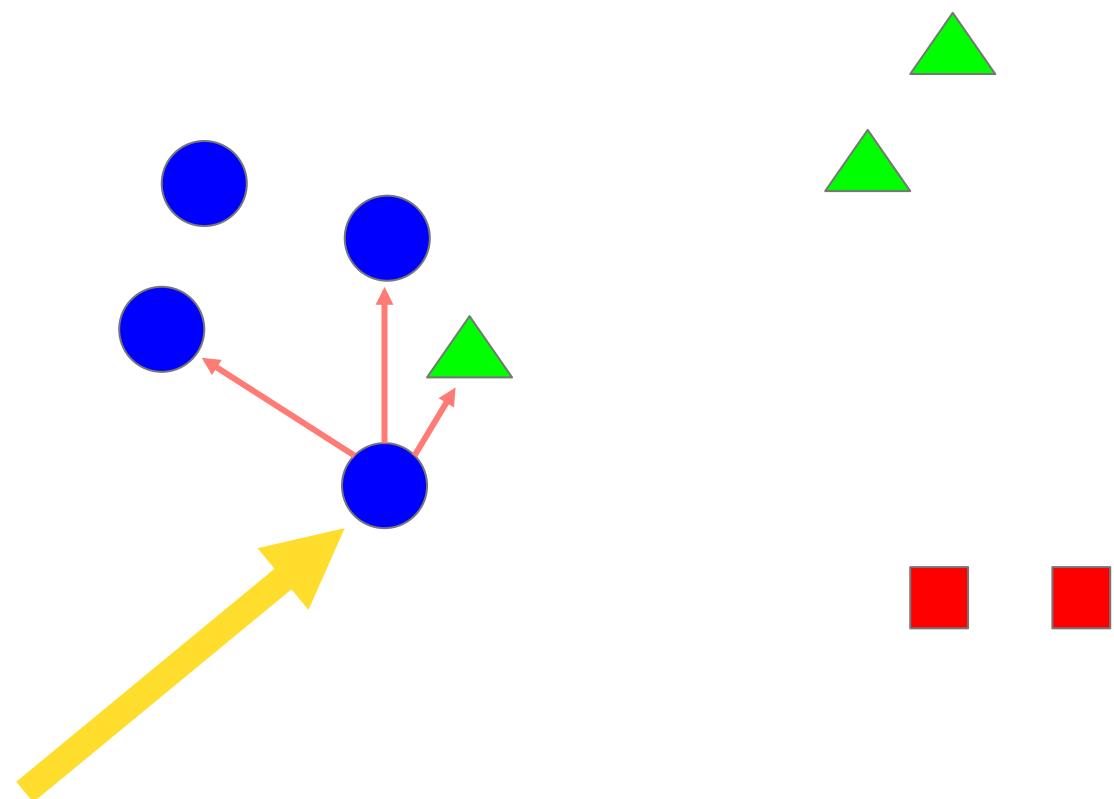
$$P(i) = \begin{cases} \text{precision at } i, & \text{if the } i\text{th retrieval is correct} \\ 0, & \text{otherwise} \end{cases}$$



MAP@R

$$\text{MAP}@R = \frac{1}{R} \sum_{i=1}^R P(i)$$

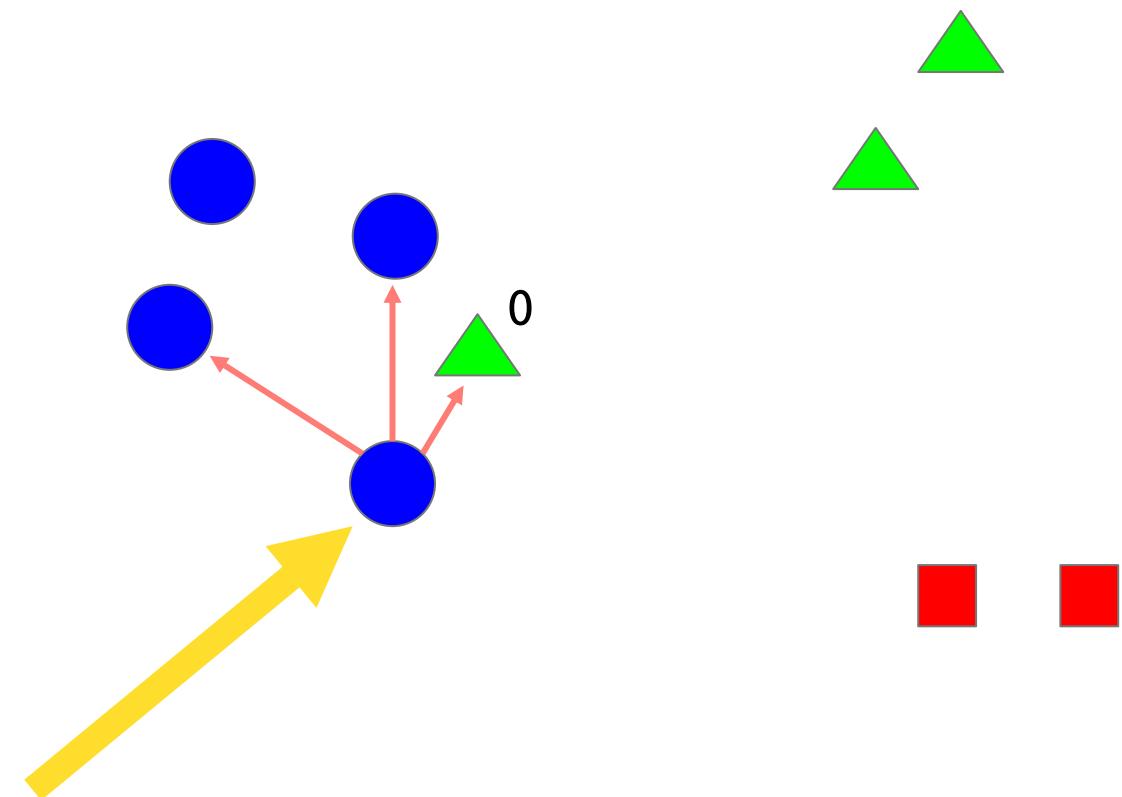
$$P(i) = \begin{cases} \text{precision at } i, & \text{if the } i\text{th retrieval is correct} \\ 0, & \text{otherwise} \end{cases}$$



MAP@R

$$\text{MAP}@R = \frac{1}{R} \sum_{i=1}^R P(i)$$

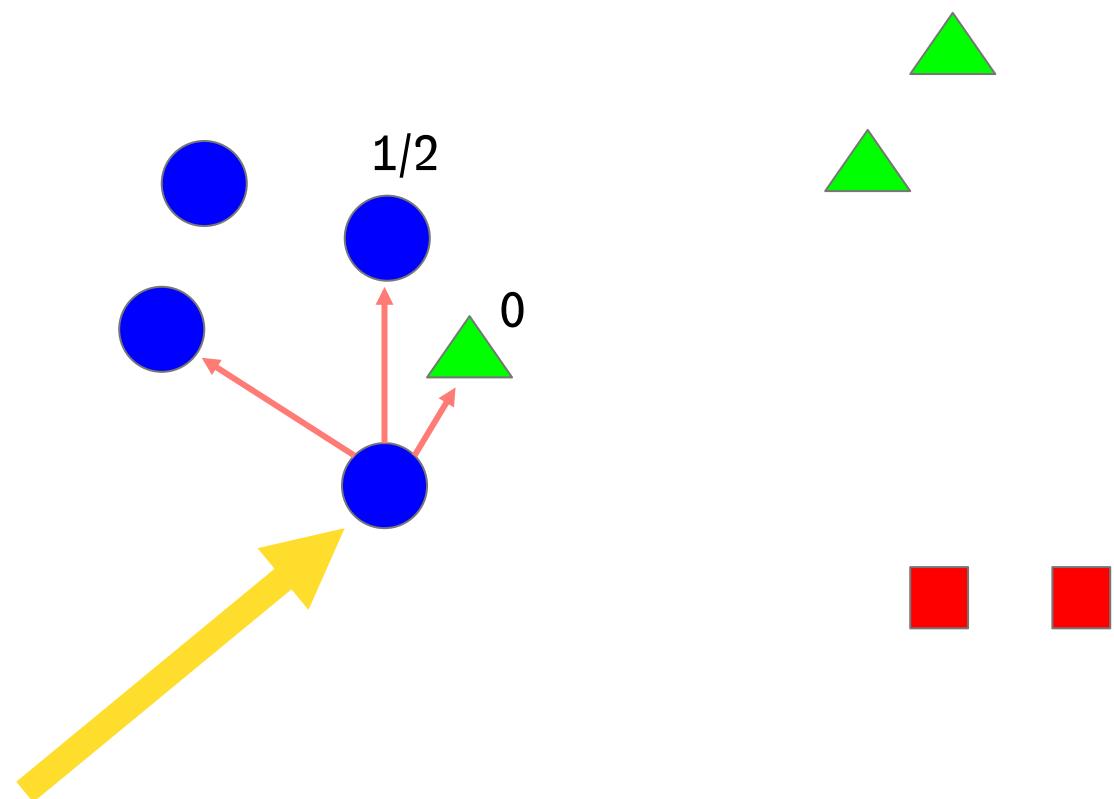
$$P(i) = \begin{cases} \text{precision at } i, & \text{if the } i\text{th retrieval is correct} \\ 0, & \text{otherwise} \end{cases}$$



MAP@R

$$\text{MAP}@R = \frac{1}{R} \sum_{i=1}^R P(i)$$

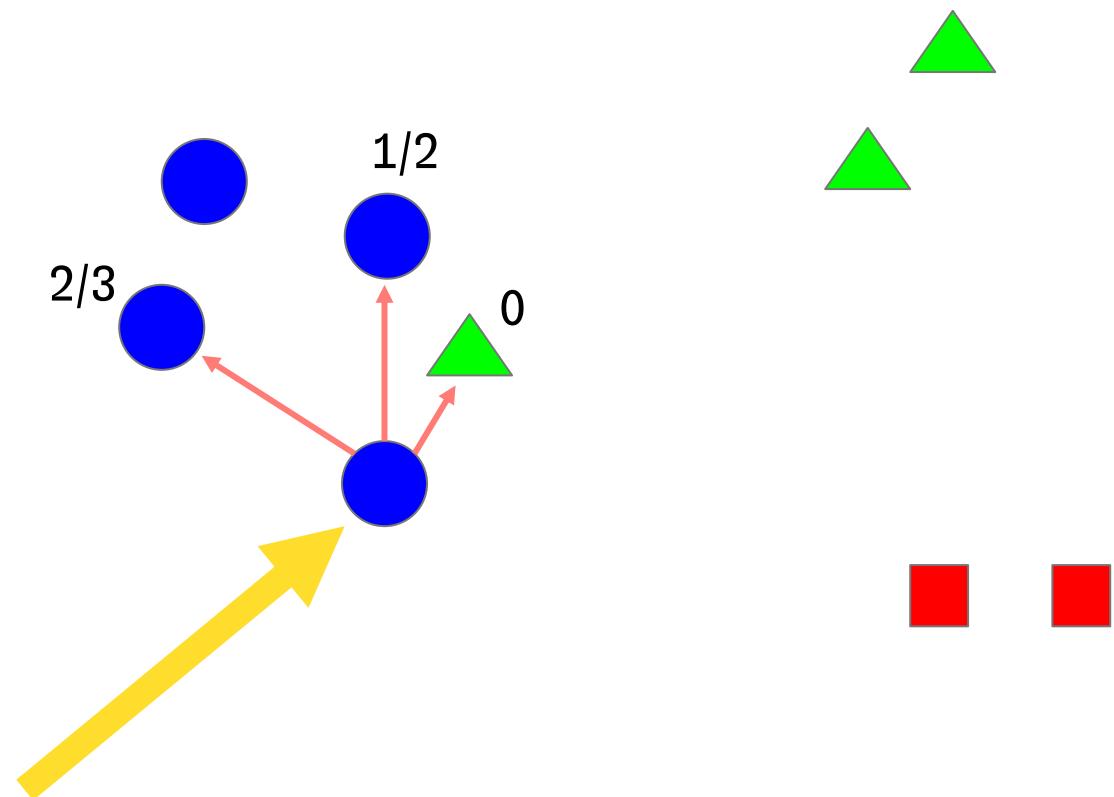
$$P(i) = \begin{cases} \text{precision at } i, & \text{if the } i\text{th retrieval is correct} \\ 0, & \text{otherwise} \end{cases}$$



MAP@R

$$\text{MAP}@R = \frac{1}{R} \sum_{i=1}^R P(i)$$

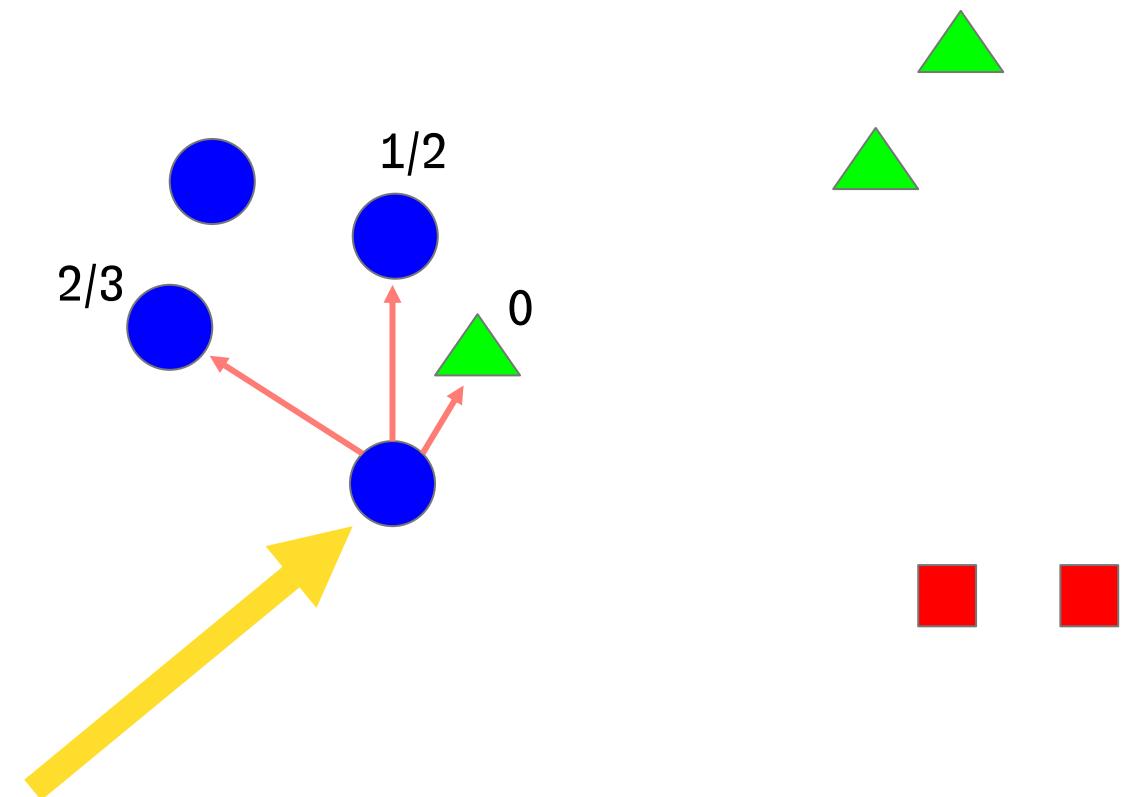
$$P(i) = \begin{cases} \text{precision at } i, & \text{if the } i\text{th retrieval is correct} \\ 0, & \text{otherwise} \end{cases}$$



MAP@R

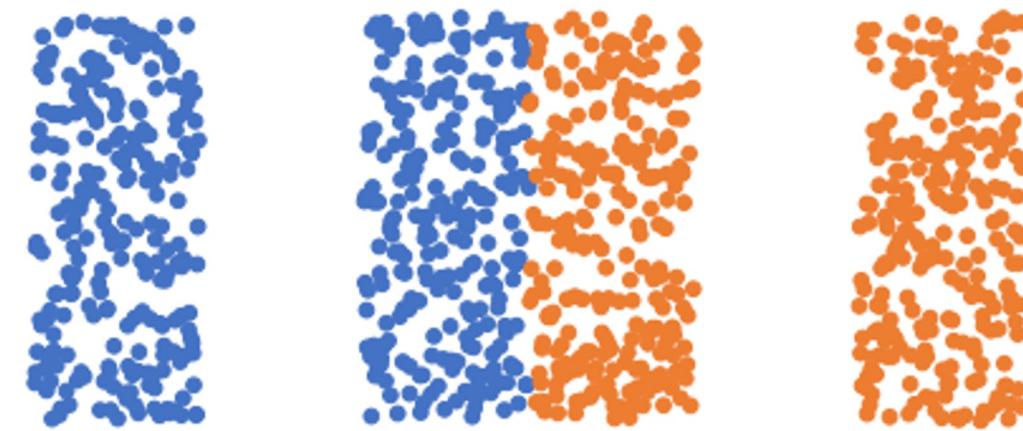
$$\text{MAP}@R = \frac{1}{R} \sum_{i=1}^R P(i)$$

$$P(i) = \begin{cases} \text{precision at } i, & \text{if the } i\text{th retrieval is correct} \\ 0, & \text{otherwise} \end{cases}$$

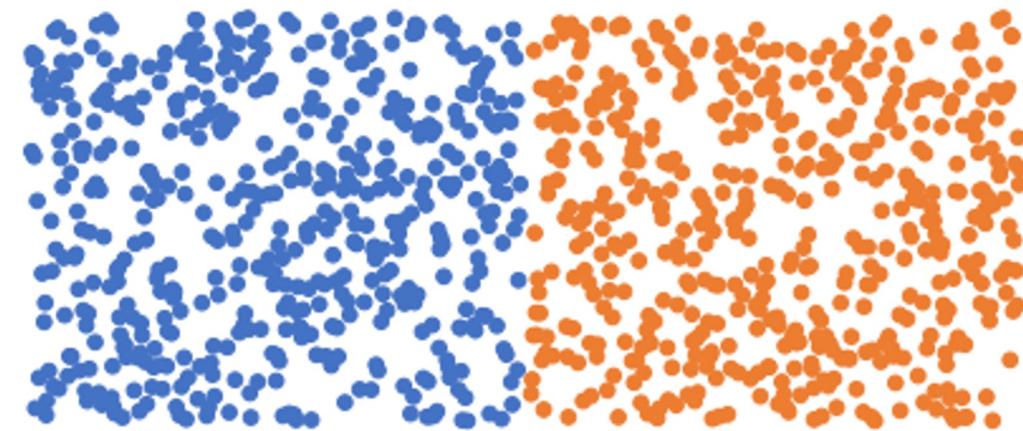


Проблемы R@K, F1

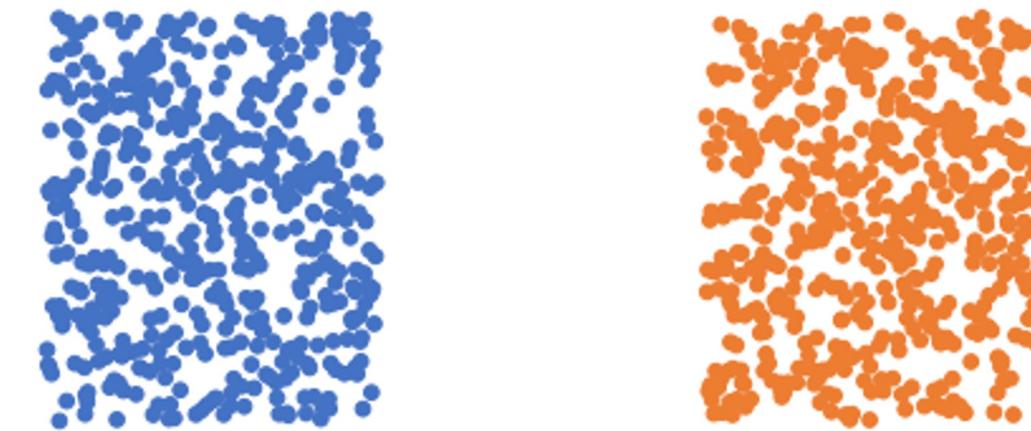
NMI: 95.6% F1: 100% R@1: 99%,
R-Precision: 77.4% MAP@R: 71.4%



NMI: 100% F1: 100% R@1: 99.8%
R-Precision: 83.3% MAP@R: 77.9%



NMI: 100% F1: 100% R@1: 100%,
R-Precision: 99.8% MAP@R: 99.8%



План лекции

01

Что такое Metric Learning?

02

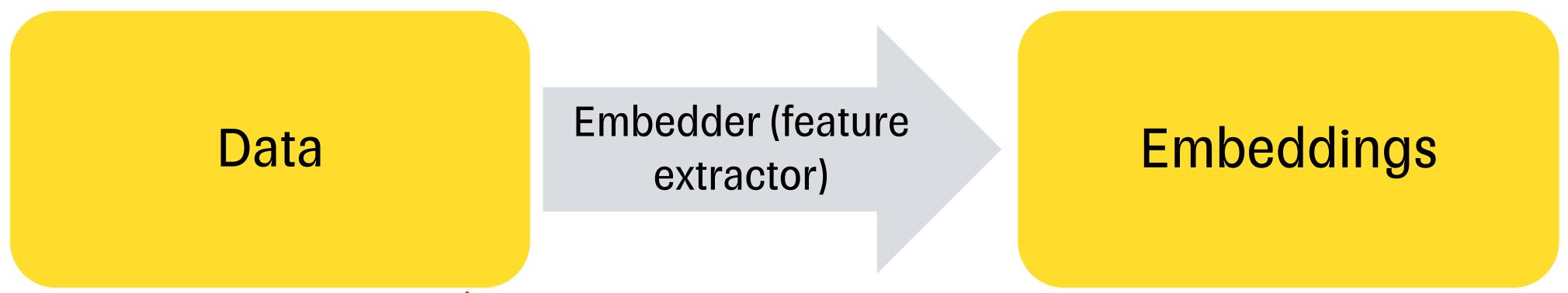
Метрики качества в
задачах Metric Learning

03

Обучение Metric Learning
моделей: contrastive
подход

04

Обучение Metric Learning
моделей:
классификационный
подход

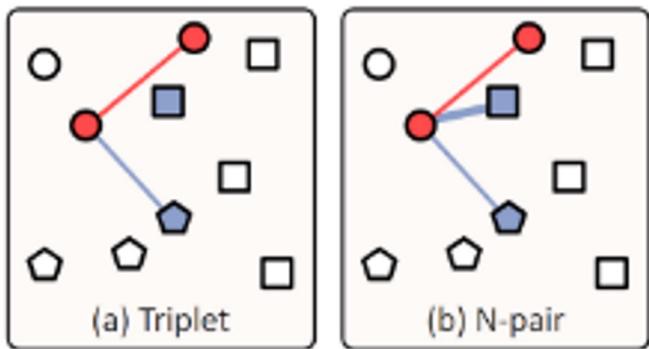




Metric learning supervised losses

Contrastive-based

- Direct distance optimization in embedding space

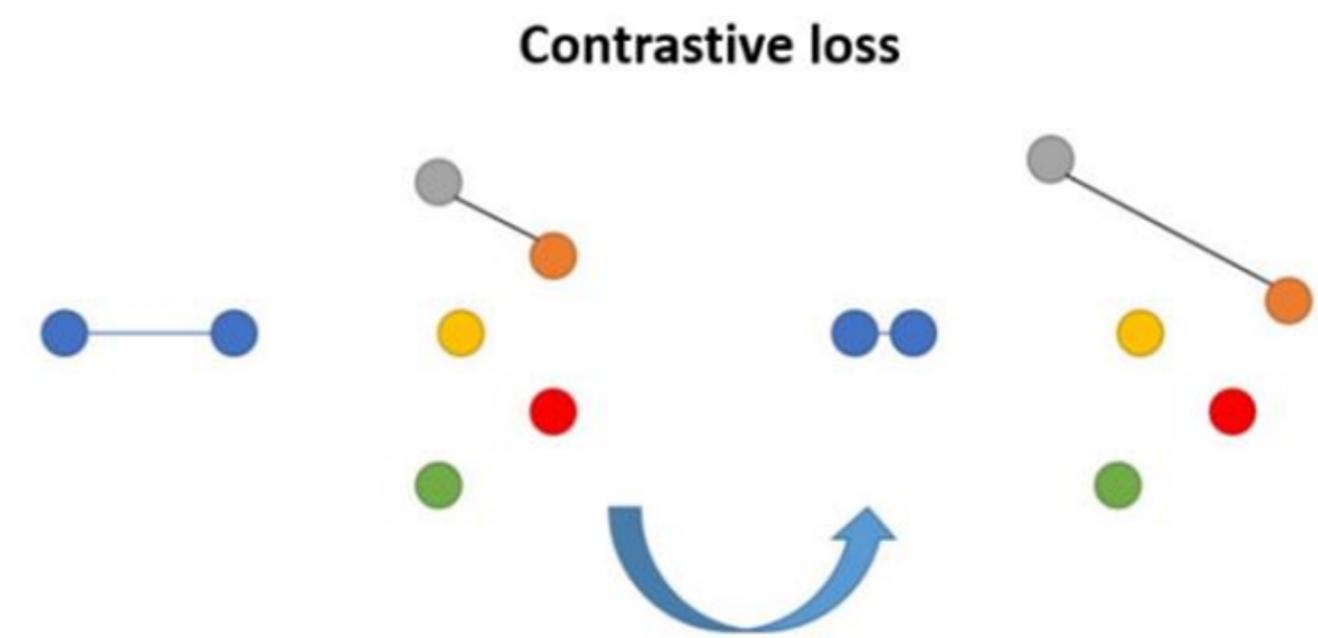


Classification-based

- Build embedding space via solving classification problem

Contrastive loss

- Positive pair - пара из объектов одного класса
- Negative pair - пара из объектов разных классов

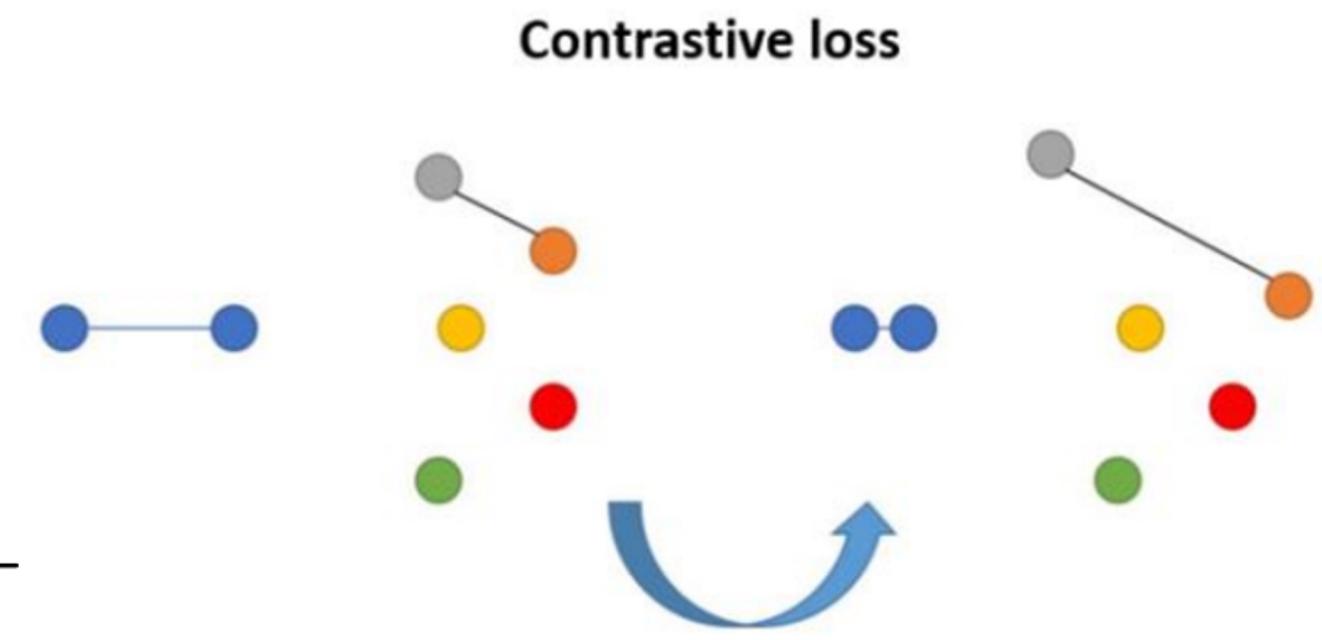


Contrastive loss

- Positive pair - пара из объектов одного класса
- Negative pair - пара из объектов разных классов
- Contrastive loss:

$$L_{\text{contrastive}} = [d_p - m_{\text{pos}}]_+ + [m_{\text{neg}} - d_n]_+$$

$m_{\text{pos}}, m_{\text{neg}}$ - positive and negative margins

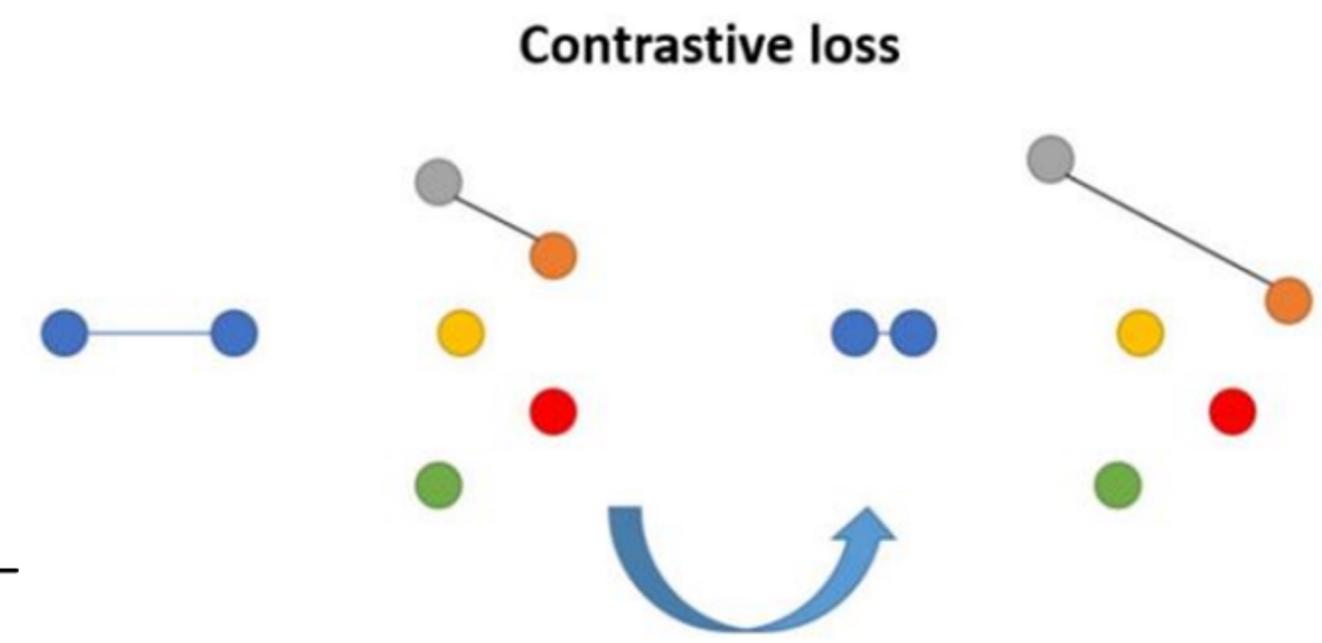


Contrastive loss

- Positive pair - пара из объектов одного класса
- Negative pair - пара из объектов разных классов
- Contrastive loss:

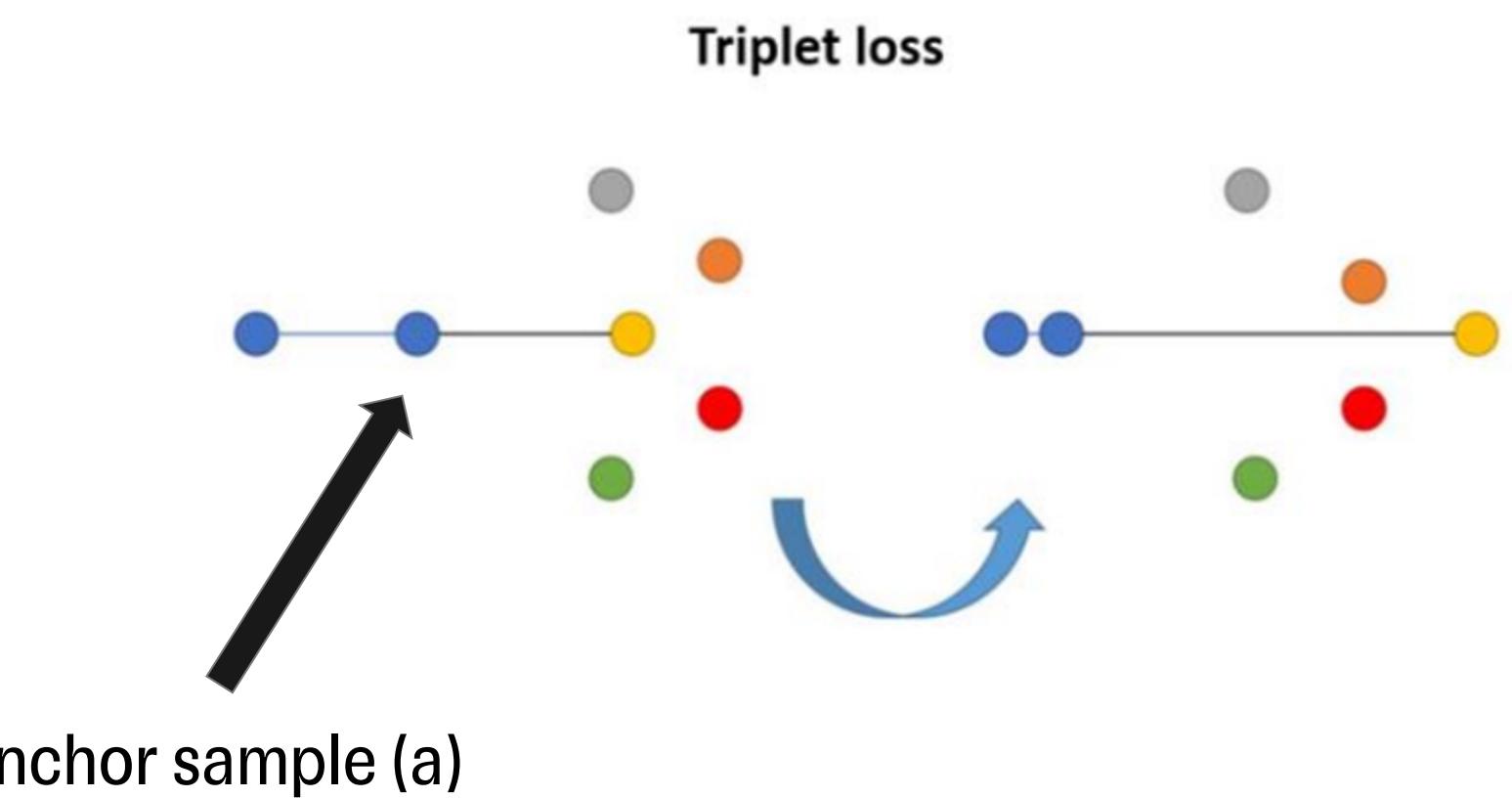
$$L_{\text{contrastive}} = [d_p - m_{\text{pos}}]_+ + [m_{\text{neg}} - d_n]_+$$

$m_{\text{pos}}, m_{\text{neg}}$ - positive and negative margins



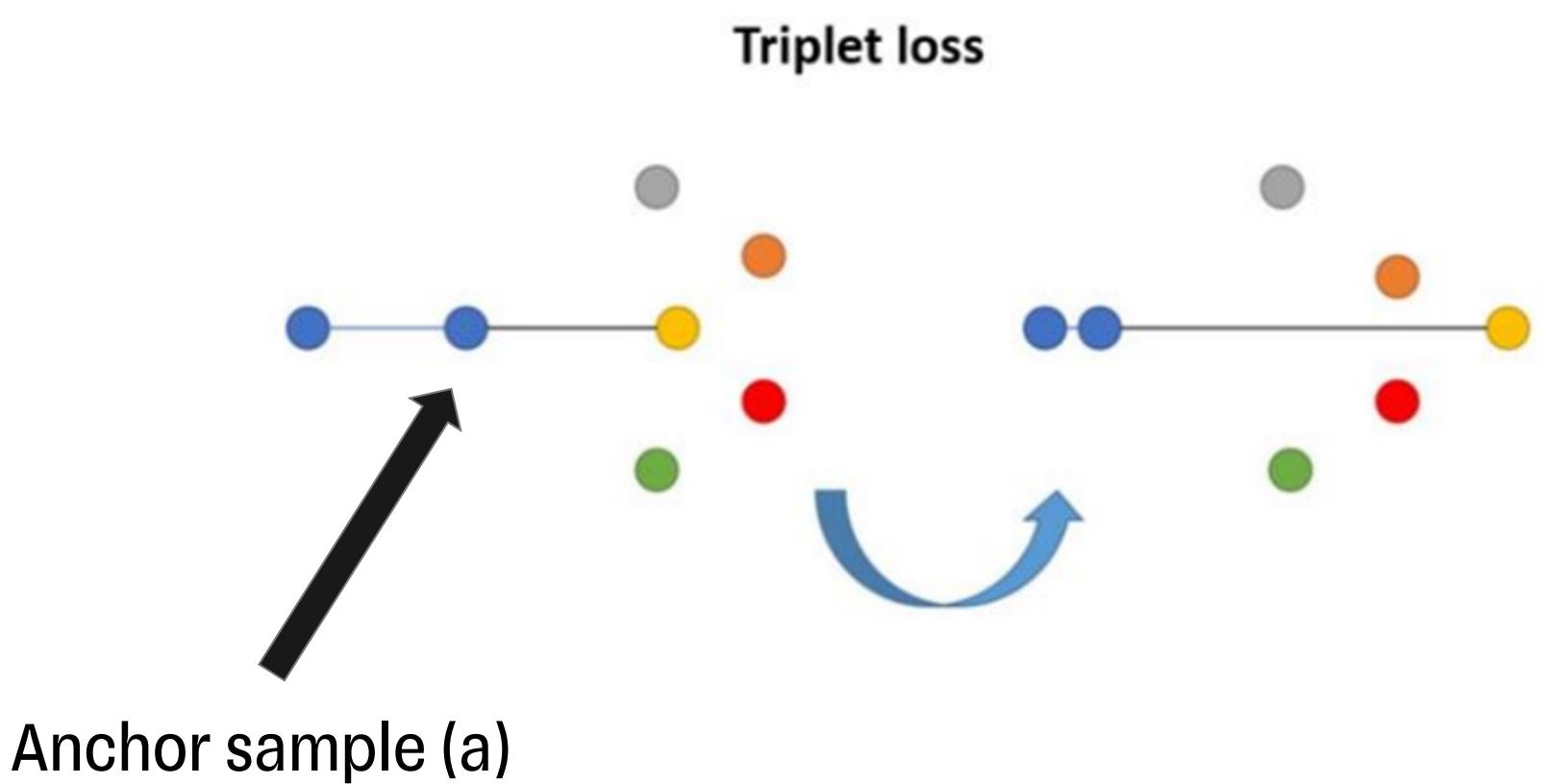
Проблема: одинаковые margin для всех пар, вне зависимости от их похожести

Triplet loss



Triplet loss

$$L_{\text{triplet}} = [d_{ap} - d_{an} + m]_+$$

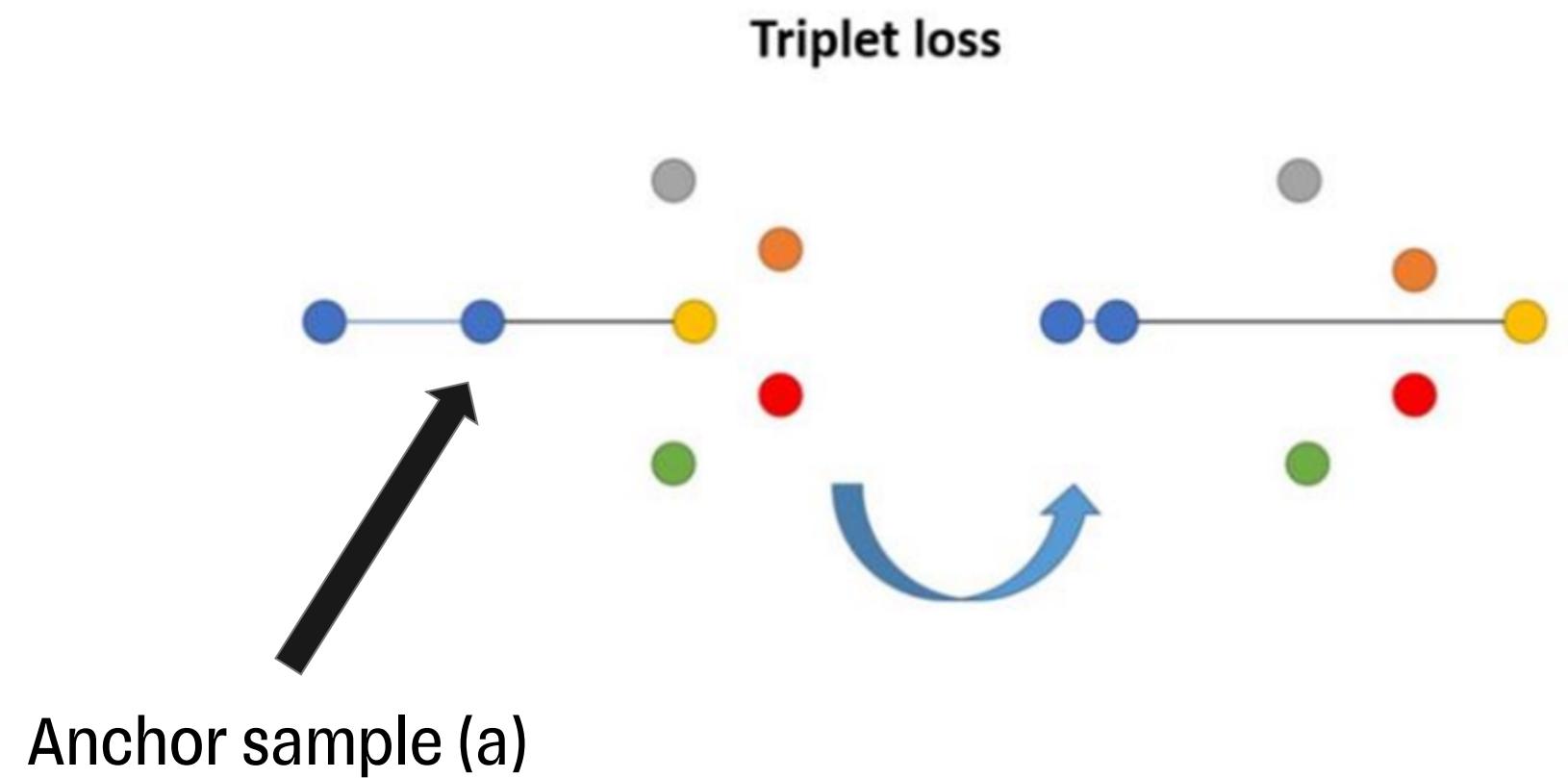


Triplet loss

$$L_{\text{triplet}} = [d_{ap} - d_{an} + m]_+$$

Общая проблема contrastive-based лоссов: при сэмплировании пар, размер датасета становится как бы n^2 (n - размер датасета), что приводит к увеличению времени, требуемого для обучения

Сложность обучения: $O(n^2)$ - contrastive, $O(n^3)$ - triplet



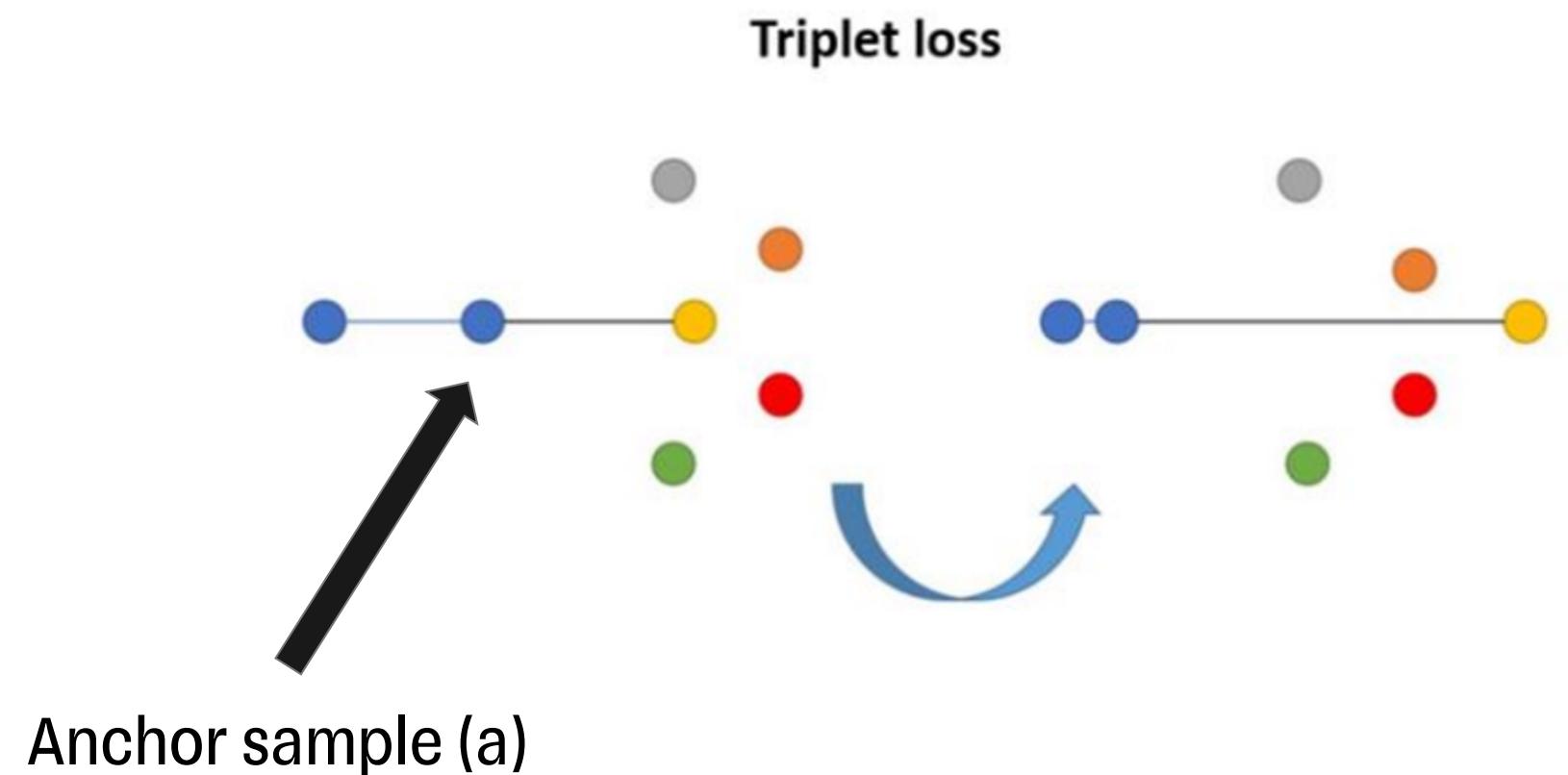
Triplet loss

$$L_{\text{triplet}} = [d_{ap} - d_{an} + m]_+$$

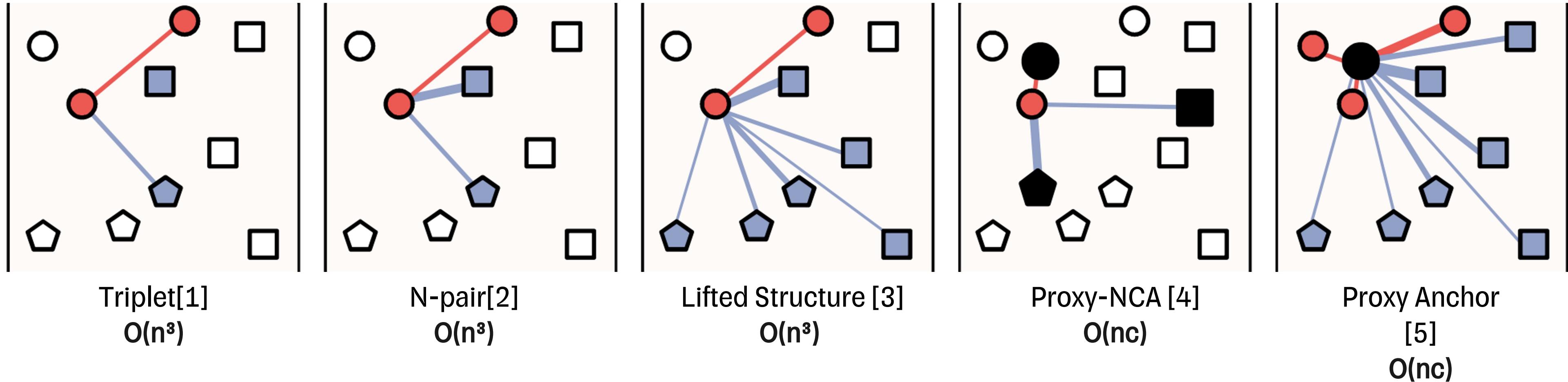
Общая проблема contrastive-based лоссов: при сэмплировании пар, размер датасета становится как бы n^2 (n - размер датасета), что приводит к увеличению времени, требуемого для обучения

Сложность обучения: $O(n^2)$ - contrastive, $O(n^3)$ - triplet

Для решения этой проблемы придуманы основные подходы - введение proxy и стратегии сэмплирования пар/триплетов



Proxy



n - число сэмплов в датасете, c - число классов

Proxy - “представитель” класса, выбираемый либо случайно, либо из других соображений

[1] A unified embedding for face recognition and clustering (<https://arxiv.org/abs/1503.03832>)

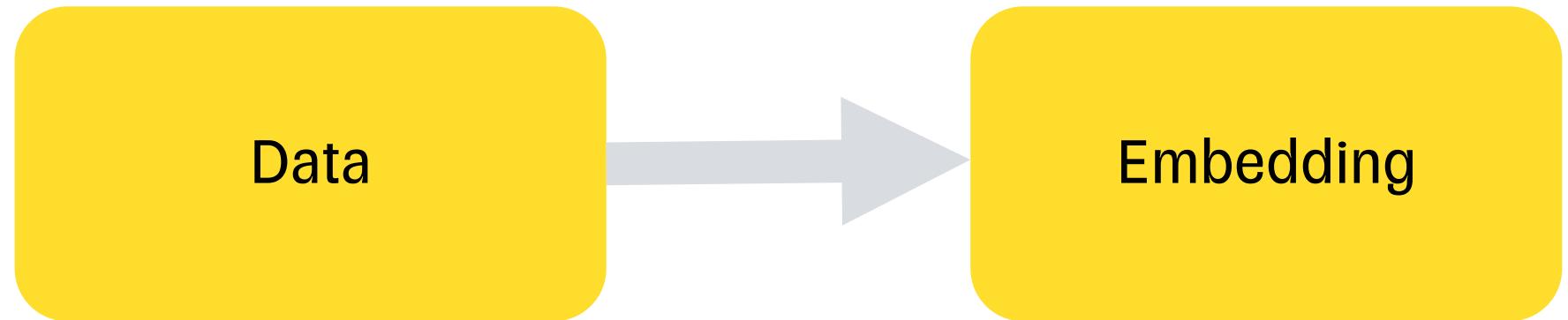
[2] Improved deep metric learning with multiclass n-pair loss objective (<https://papers.nips.cc/paper/2016/hash/6b180037abbebea991d8b1232f8a8ca9-Abstract.html>)

[3] Deep metric learning via lifted structured feature embedding (<https://arxiv.org/abs/1511.06452>)

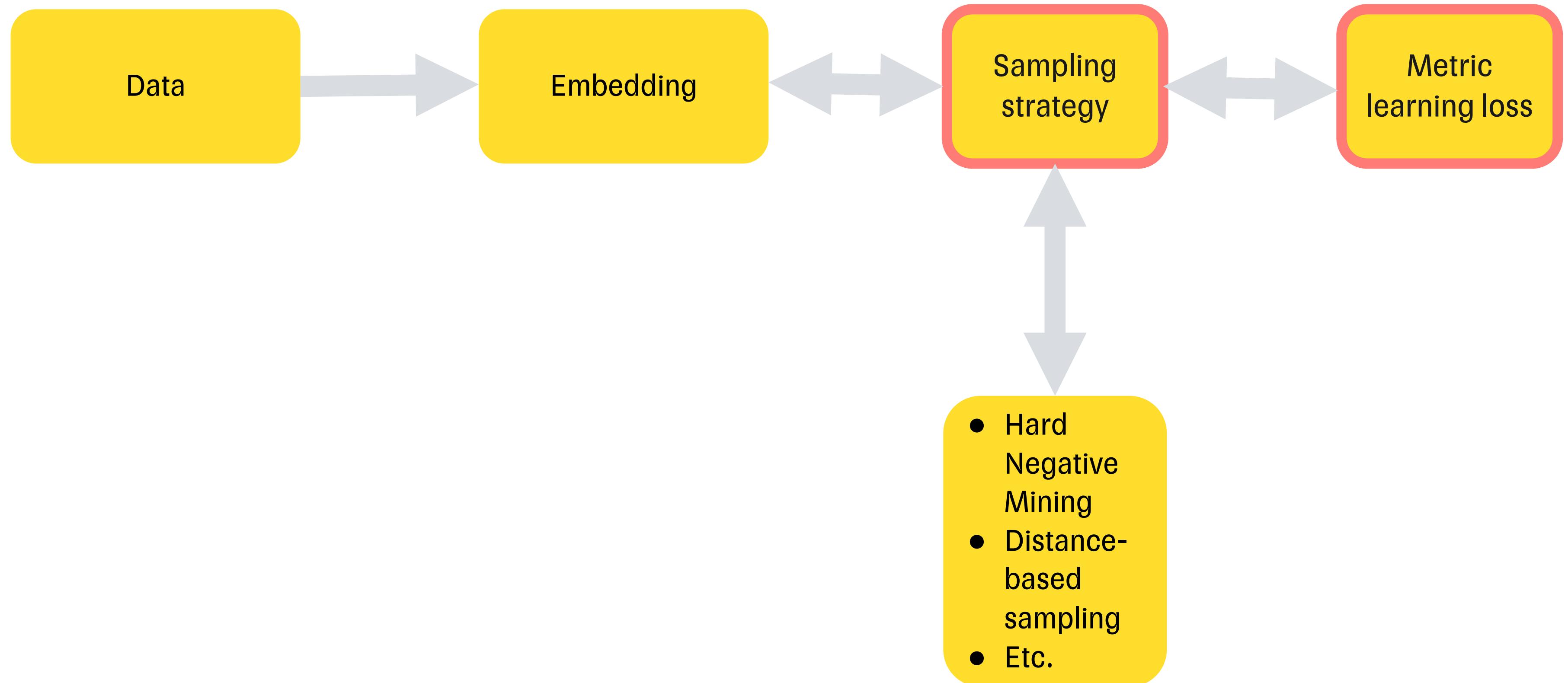
[4] No fuss distance metric learning using proxies (<https://arxiv.org/abs/1703.07464>)

[5] Proxy Anchor Loss for Deep Metric Learning (<https://arxiv.org/pdf/2003.13911.pdf>)

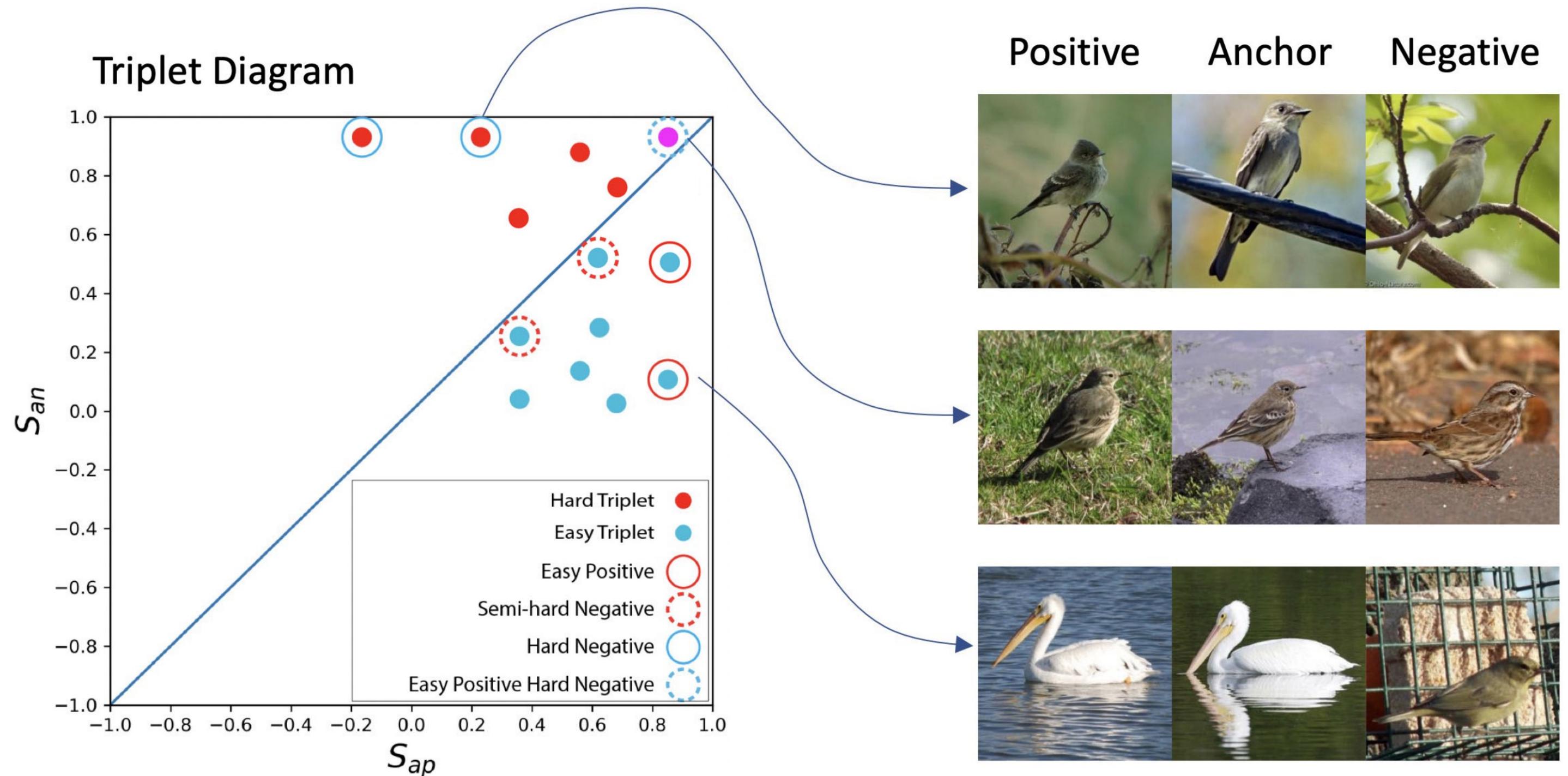
Sampling matters!



Sampling matters!

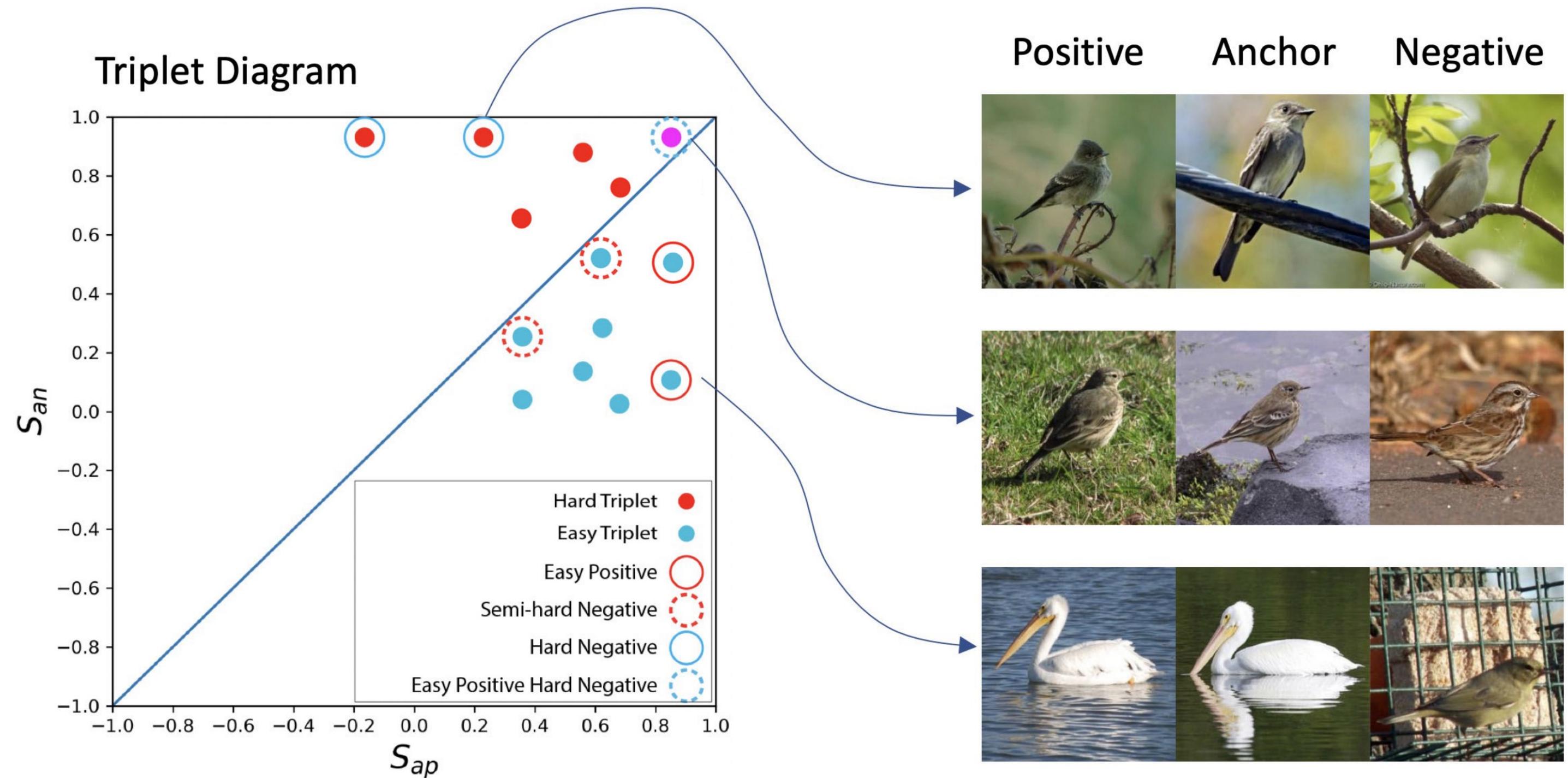


Hard negative mining



Hard negative mining

- Hard Negative Mining:
выбираем триплеты,
где negative близок к
anchor



План лекции

01

Что такое Metric Learning?

02

Метрики качества в
задачах Metric Learning

03

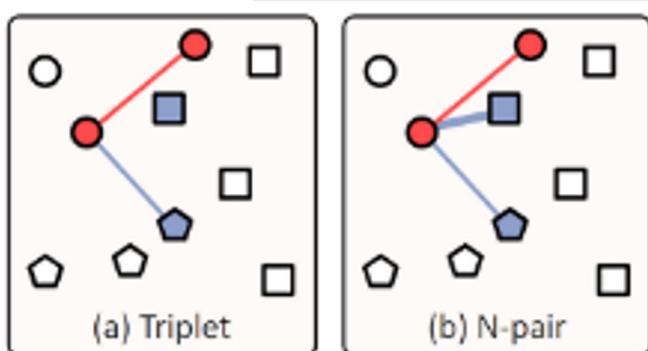
Обучение Metric Learning
моделей: contrastive
подход

04

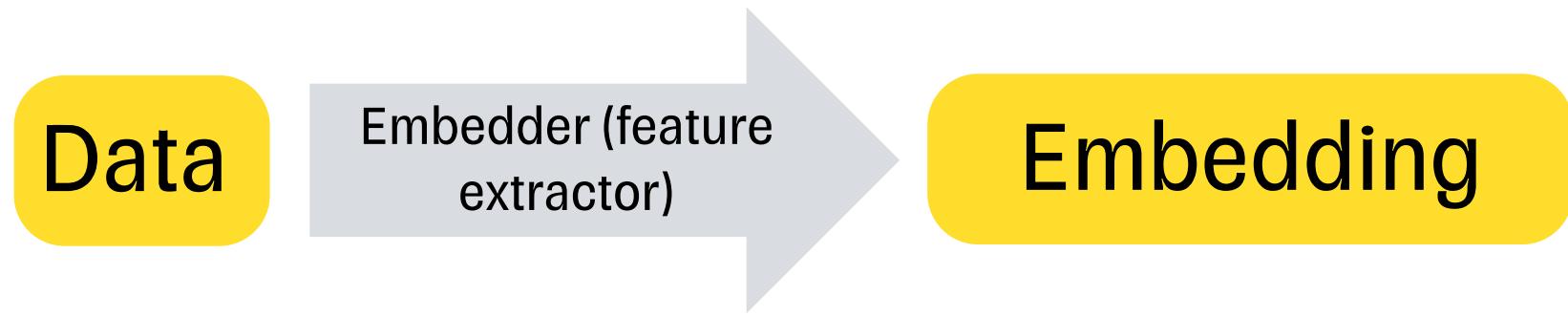
Обучение Metric Learning
моделей:
классификационный
подход

Metric learning supervised losses

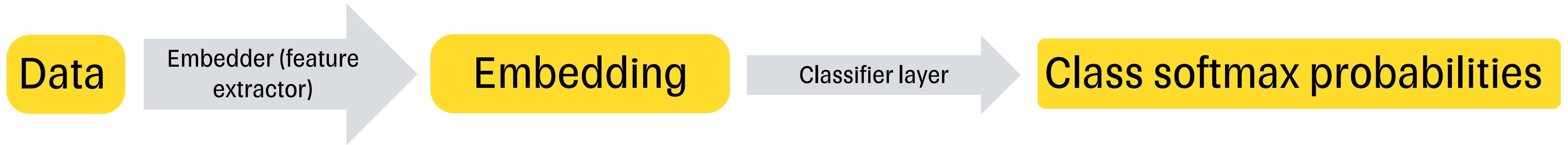
Contrastive-based	Classification-based
<ul style="list-style-type: none">▪ Direct distance optimization in embedding space	<ul style="list-style-type: none">▪ Build embedding space via solving classification problem
<ul style="list-style-type: none">▪ Contrastive (classic)▪ Triplet (classic)▪ Proxy-based▪ Distance learning▪ SNR loss (2019)▪ Proxy Anchor loss (2020)▪ Ranked List loss (2019)▪ Etc.	<ul style="list-style-type: none">▪ Softmax loss (classic)▪ Normalized softmax▪ Margin-based▪ Angular margin loss



Classification-based metric learning losses



Classification-based metric learning losses



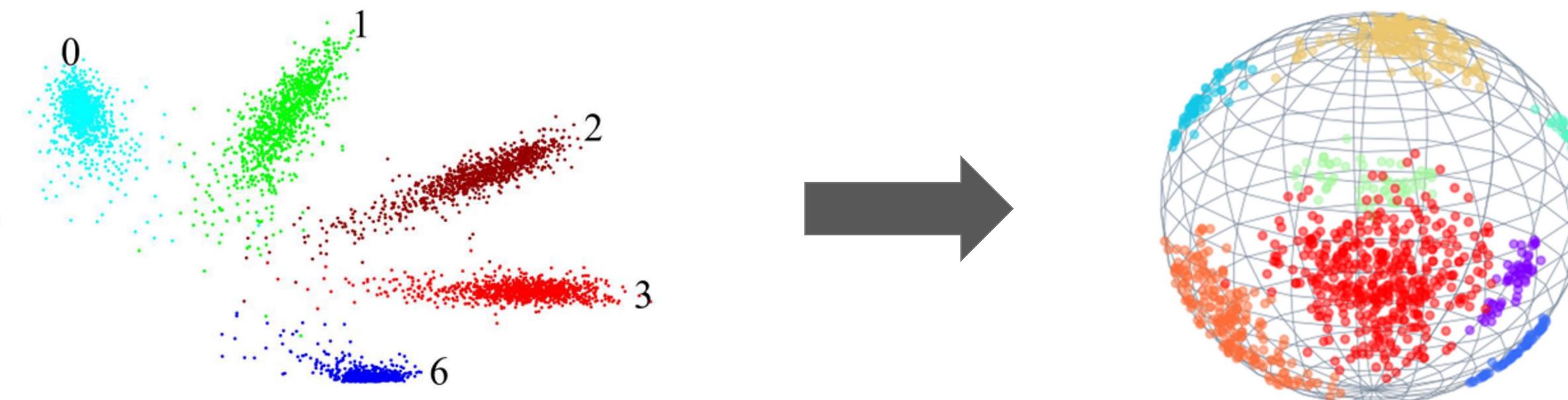
$$L_{softmax} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{xW_i^T + b_i}}{e^{xW_i^T + b_i} + \sum_{j=1, j \neq y_i}^n e^{xW_j^T + b_j}}$$

Classification-based metric learning losses

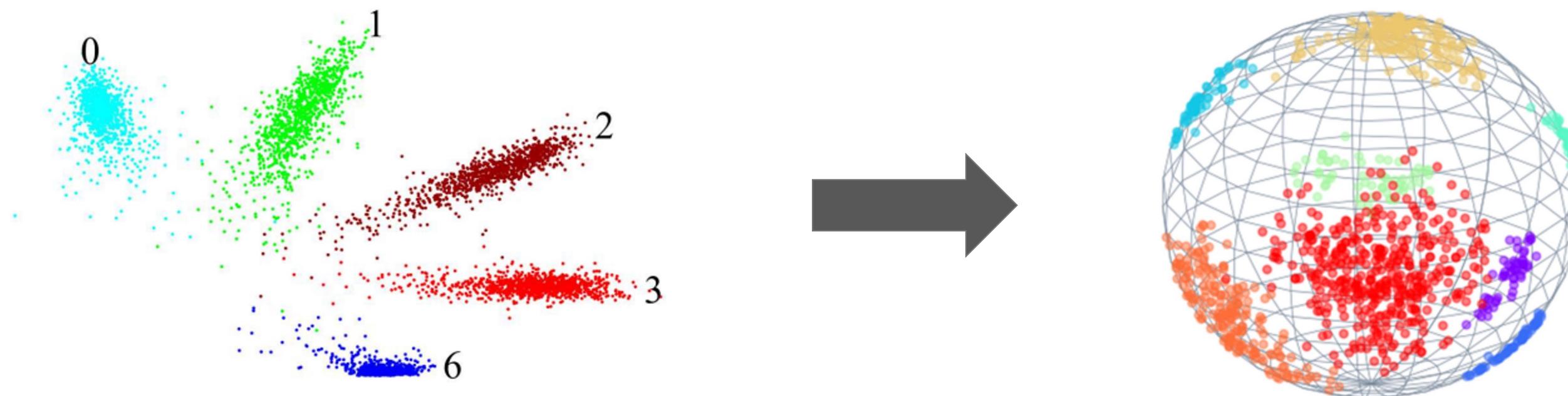
Мотивация: при обучении с Softmax на задачу классификации, перед последним линейным слоем по сути формируется пространство *embeddings*, на котором уже учится линейный классификатор



Spherical embeddings

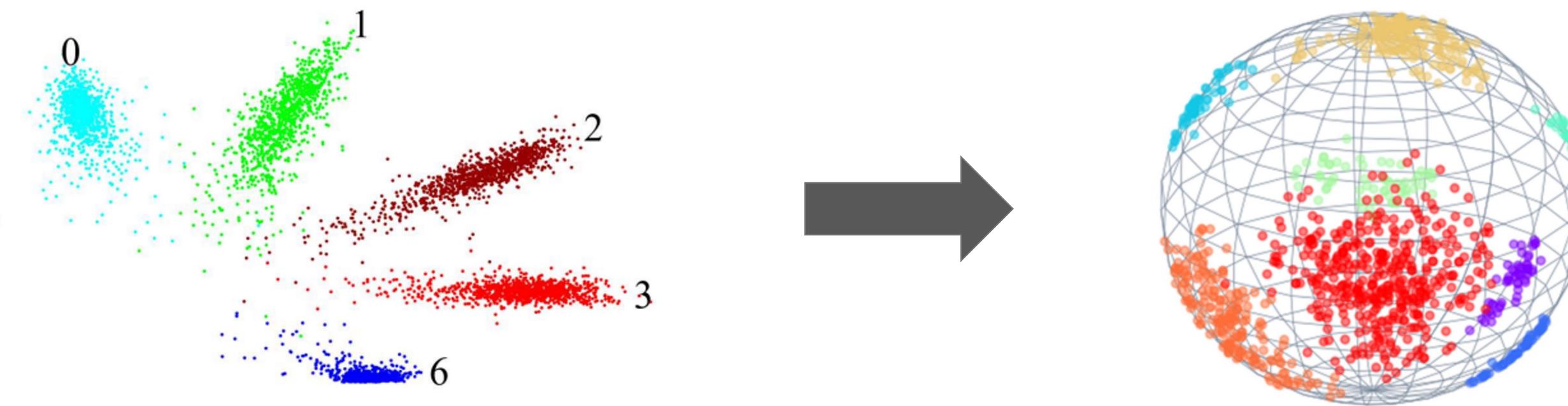


Normalized softmax



$$L_{softmax} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{xW_i^T + b_i}}{e^{xW_i^T + b_i} + \sum_{j=1, j \neq y_i}^n e^{xW_j^T + b_j}}$$

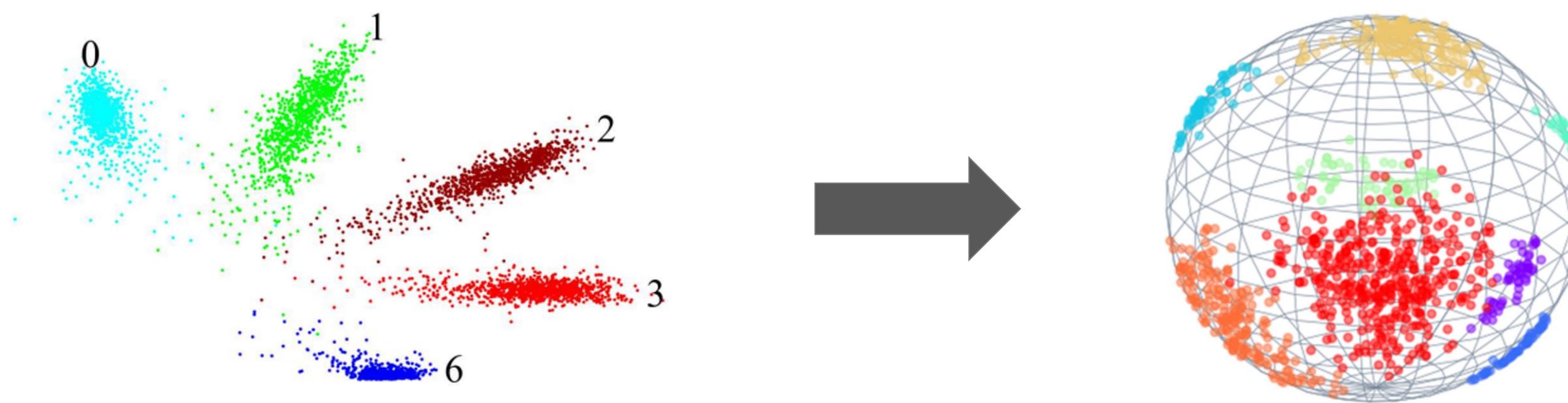
Normalized softmax



$$\|W_{y_i}\| = 1, \|x_i\| = 1$$

$$L_{softmax} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{xW_i^T + b_i}}{e^{xW_i^T + b_i} + \sum_{j=1, j \neq y_i}^n e^{xW_j^T + b_j}}$$

Normalized softmax

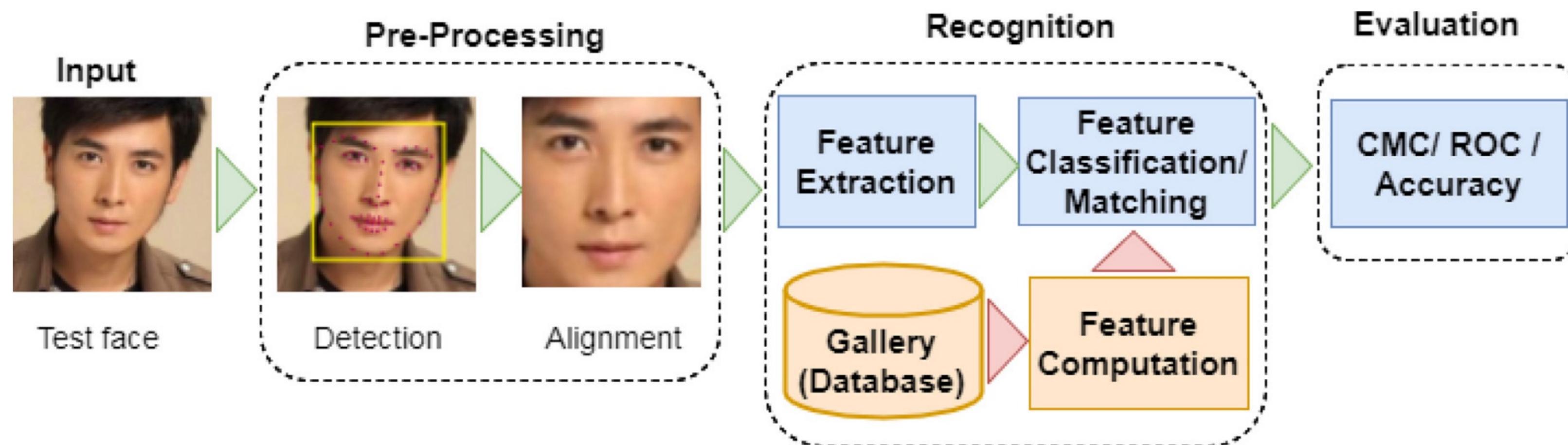


$$\|W_{y_i}\| = 1, \|x_i\| = 1$$

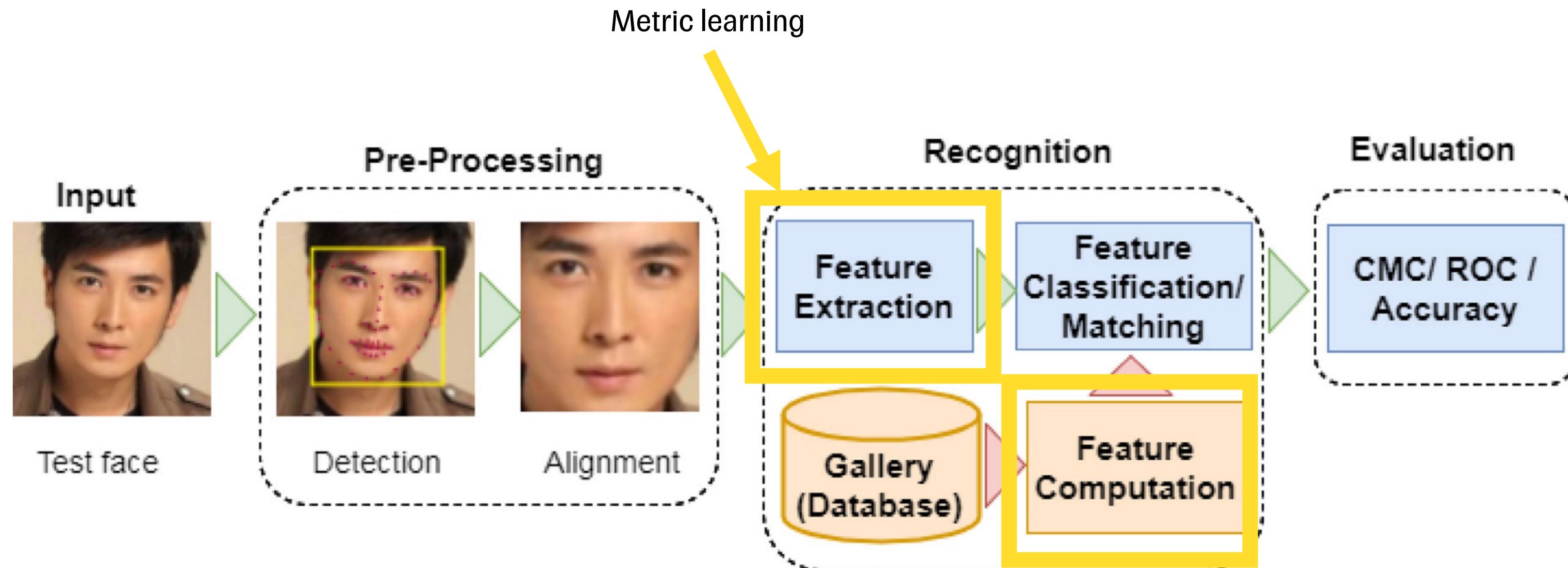
$$L_{softmax} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{xW_i^T + b_i}}{e^{xW_i^T + b_i} + \sum_{j=1, j \neq y_i}^n e^{xW_j^T + b_j}}$$

$$L_{Nsoftmax} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cos \theta_{y_i}}}{e^{s \cos \theta_{y_i}} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

Deep face recognition



Deep face recognition



Margin

$$L_{margin} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s\phi(\theta_j)}}{e^{s\phi(\theta_j)} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

$$\phi(\theta_j) = (\cos(m_1\theta_j + m_2) - m_3)$$

Margin

$$L_{margin} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s\phi(\theta_j)}}{e^{s\phi(\theta_j)} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

$$\phi(\theta_j) = (\cos(m_1\theta_j + m_2) - m_3)$$

$$m_2 = m_3 = 0 \quad \text{SphereFace (Multiplicative margin)}$$

Margin

$$L_{margin} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s\phi(\theta_j)}}{e^{s\phi(\theta_j)} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

$$\phi(\theta_j) = (\cos(m_1 \theta_j + m_2) - m_3)$$

$m_2 = m_3 = 0$ SphereFace (Multiplicative margin)

$m_1 = 1, m_2 = 0$ CosFace (Cosine margin, минимизация косинусного расстояния)

Margin

$$L_{margin} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s\phi(\theta_j)}}{e^{s\phi(\theta_j)} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

$$\phi(\theta_j) = (\cos(m_1\theta_j + m_2) - m_3)$$

$m_2 = m_3 = 0$ SphereFace (Multiplicative margin)

$m_1 = 1, m_2 = 0$ CosFace (Cosine margin, минимизация косинусного расстояния)

$m_1 = 1, m_3 = 0$ ArcFace (Angular margin, минимизация геодезического расстояния)

Margin

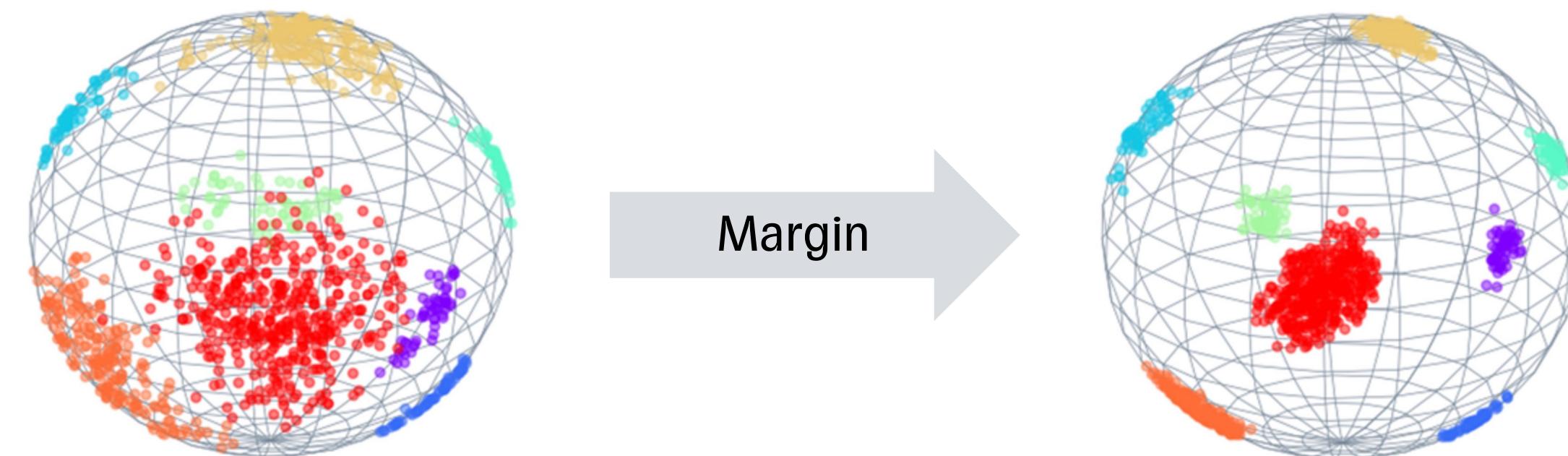
$$L_{margin} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s\phi(\theta_j)}}{e^{s\phi(\theta_j)} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

$$\phi(\theta_j) = (\cos(m_1 \theta_j + m_2) - m_3)$$

$m_2 = m_3 = 0$ SphereFace (Multiplicative margin)

$m_1 = 1, m_2 = 0$ CosFace (Cosine margin, минимизация косинусного расстояния)

$m_1 = 1, m_3 = 0$ ArcFace (Angular margin, минимизация геодезического расстояния)



Margin

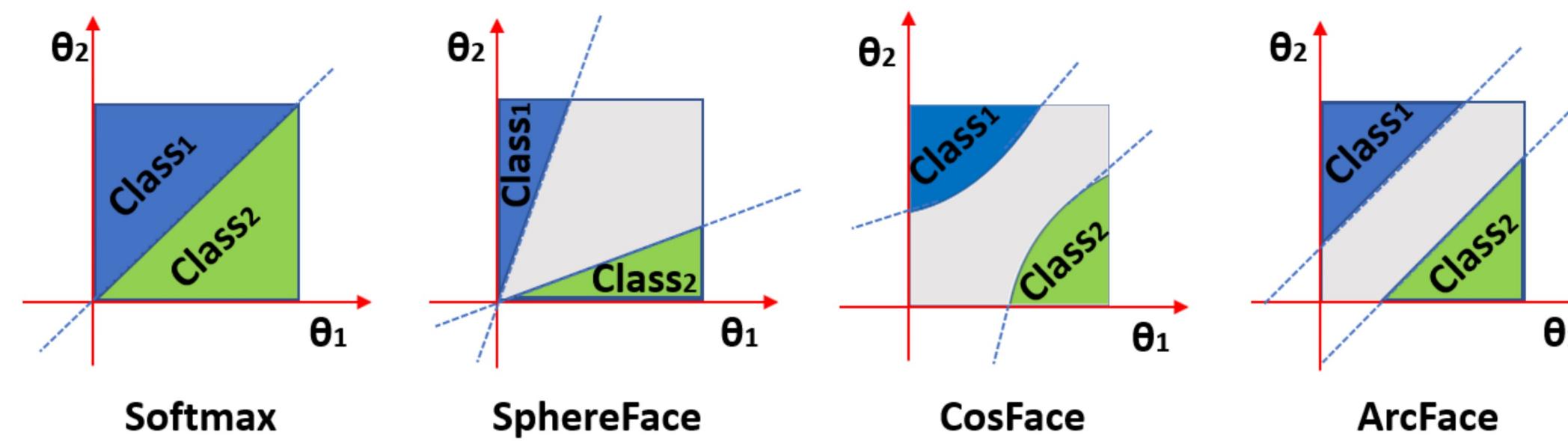
$$L_{margin} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s\phi(\theta_j)}}{e^{s\phi(\theta_j)} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

$$\phi(\theta_j) = (\cos(m_1 \theta_j + m_2) - m_3)$$

$m_2 = m_3 = 0$ SphereFace (Multiplicative margin)

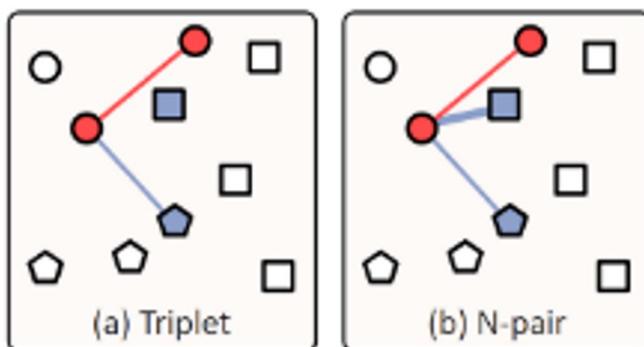
$m_1 = 1, m_2 = 0$ CosFace (Cosine margin, минимизация косинусного расстояния)

$m_1 = 1, m_3 = 0$ ArcFace (Angular margin, минимизация геодезического расстояния)



Metric learning supervised losses

Contrastive-based	Classification-based
<ul style="list-style-type: none">▪ Direct distance optimization in embedding space	<ul style="list-style-type: none">▪ Build embedding space via solving classification problem
<ul style="list-style-type: none">▪ Contrastive (classic)▪ Triplet (classic)▪ Proxy-based▪ Distance learning▪ SNR loss (2019)▪ Proxy Anchor loss (2020)▪ Ranked List loss (2019)▪ Etc.	<ul style="list-style-type: none">▪ Softmax loss (classic)▪ Normalized softmax▪ Margin-based▪ Angular margin loss



Pair-based VS Classification-based

Contrastive-based

- Большая сложность обучения (введение proxy частично решает проблему)
- Нужно заморачиваться с семплированием (как выбирать пары, тройки, proxy, ...)
- Напрямую решает задачу Metric Learning
- Удобно по памяти в GPU: нужно хранить только Embedder

Classification-based

- Сложность обучения равна размеру исходного датасета, учить быстрее
- Семплирование не так роляет
- Решение Metric Learning через прокси-задачу (классификацию)
- В GPU хранится матрица размером $C \times D$ (C - число классов, D - размерность embeddings). Иногда кратно увеличивает требуемую память



Summary

- ▶ Обсудили несколько задач, для которых полезен Metric Learning
- ▶ Познакомились с метриками качества в Metric Learning
- ▶ Разбрали pair-based и classification-based подходы, их плюсы и минусы

