

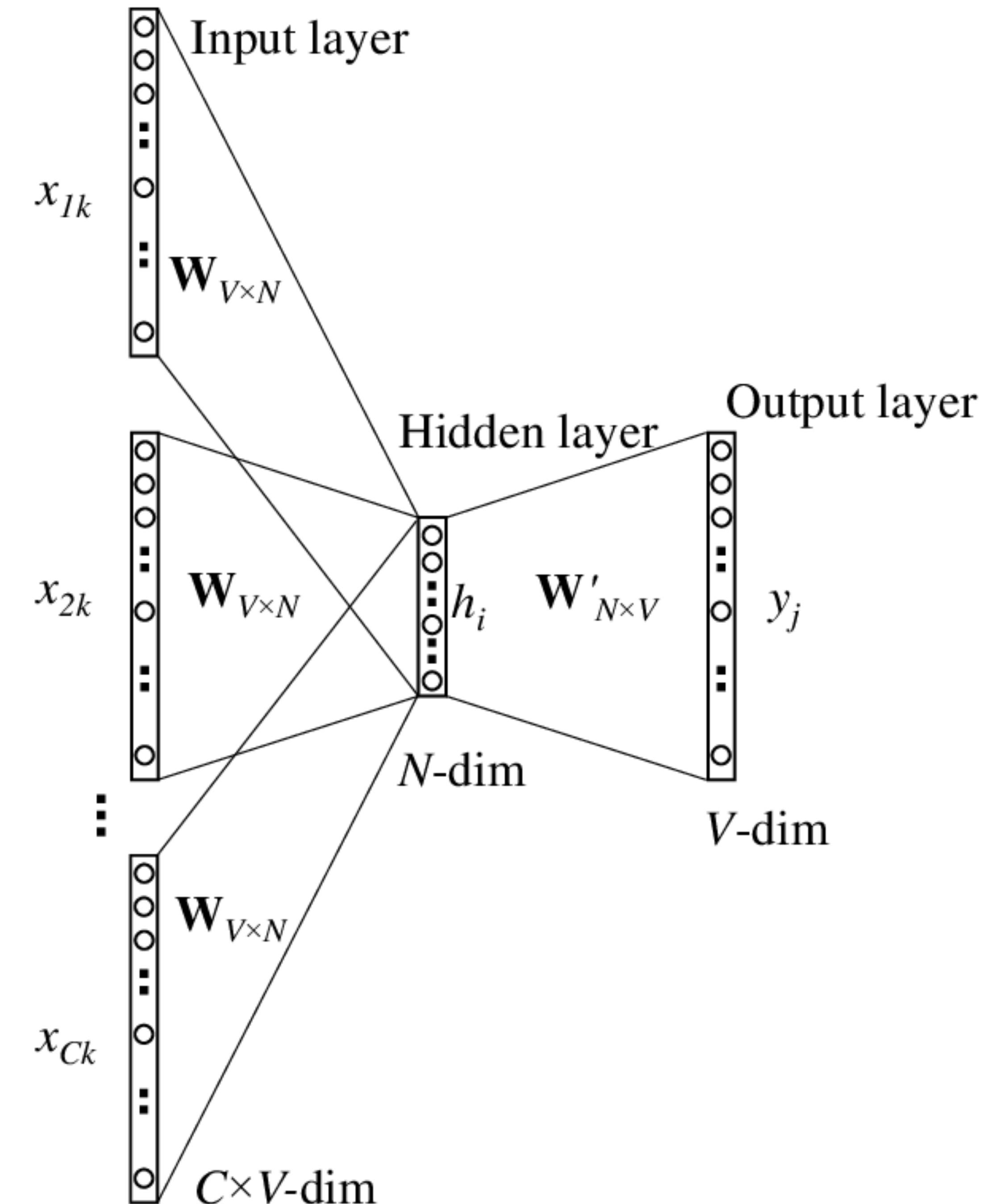
Metric Learning

Смирнов Тимофей

Word2vec

Краткий обзор

- Embedding слой на входе
- Линейный разжимающий слой + softmax
- Предсказываем пропущенный токен по контексту
- Self-supervised
- По сути задача классификации



Fasttext

Новый принцип обучения

- Идея брать n-граммы для получения эмбедингов
- Оптимизируем расстояние между эмбедингами
- Является базовым примером Metric Learning обучения

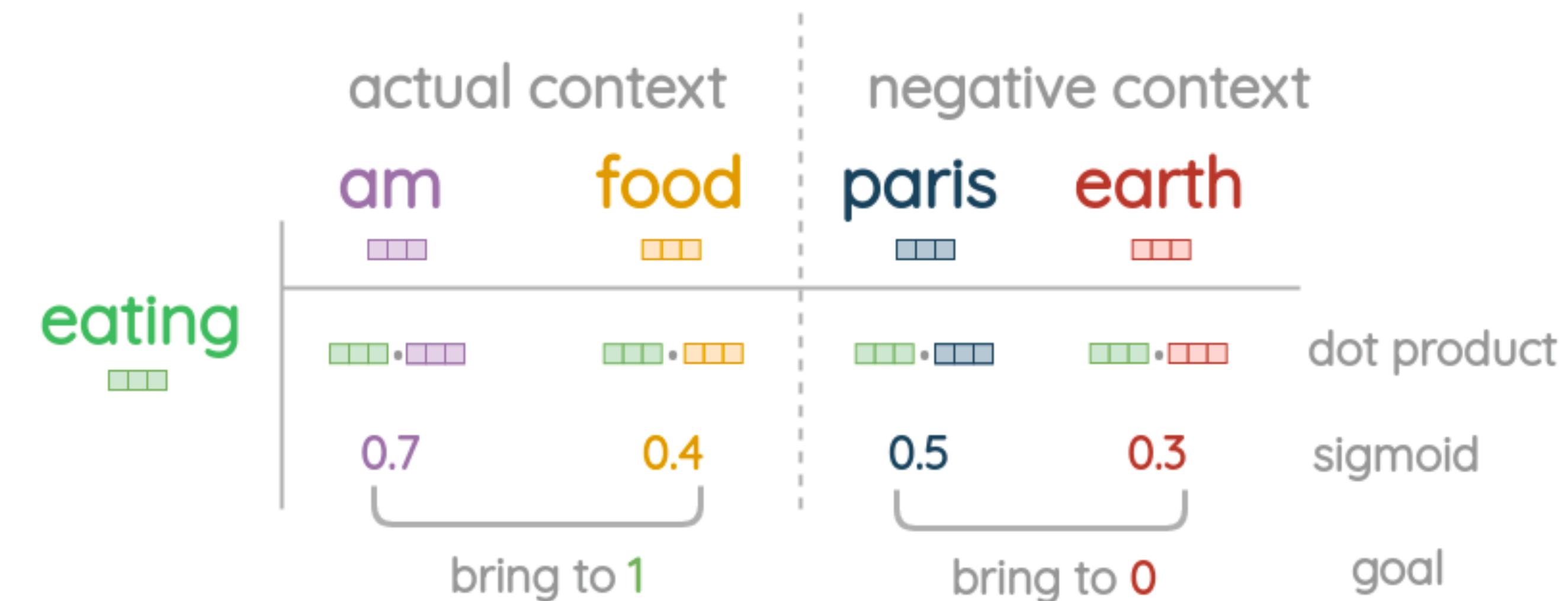
1. Выделяем контекст

I am eating food now

2. Составляем эмбединг слова как сумму эмбедингов n-грамм + эмбединг токена



3. Обучаем слова из контекста быть ближе к целевому, а остальные быть дальше



FaceNet

Face recognition task

- Задачу распознавания лиц нельзя решить обычной классификацией
- Векторные представления лиц одних людей ближе, а разных людей дальше

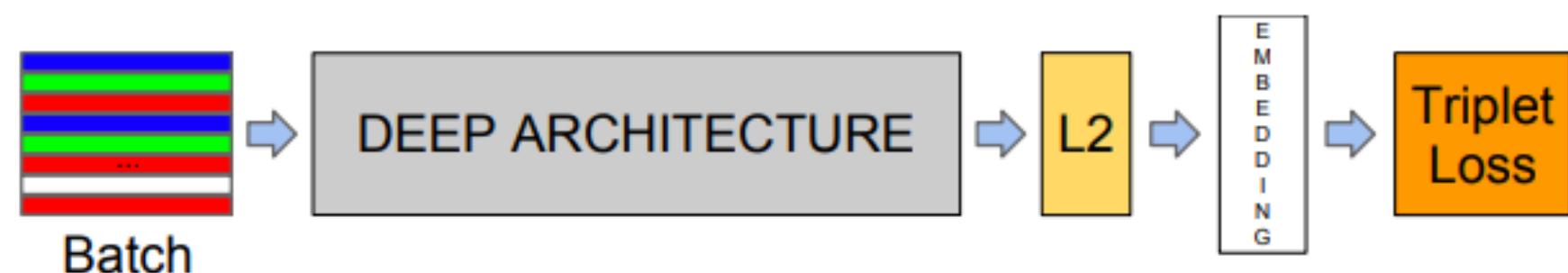
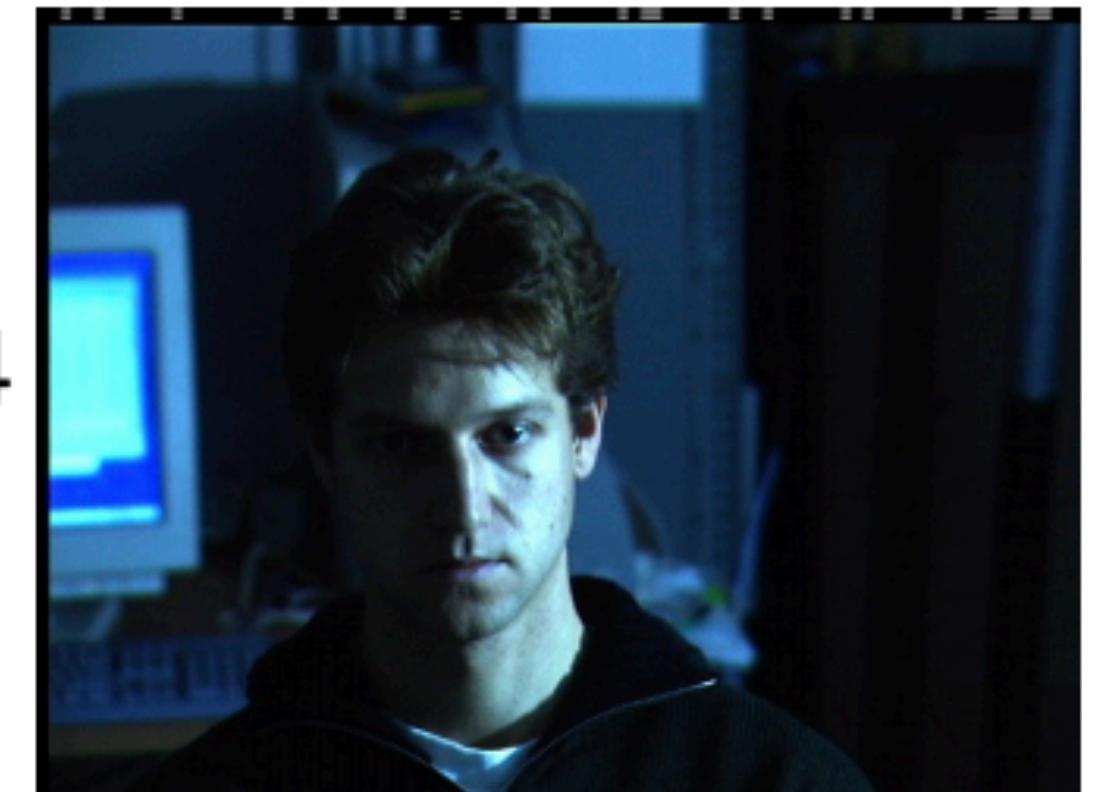


Схема работы модели



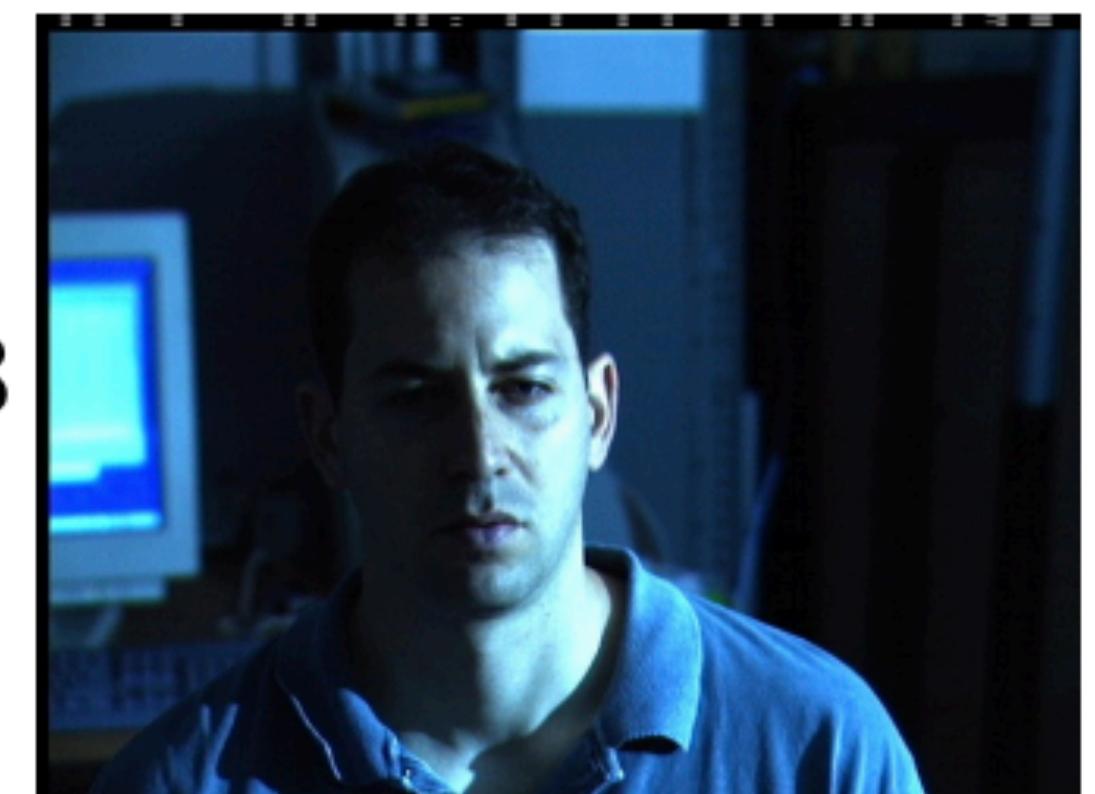
1.22



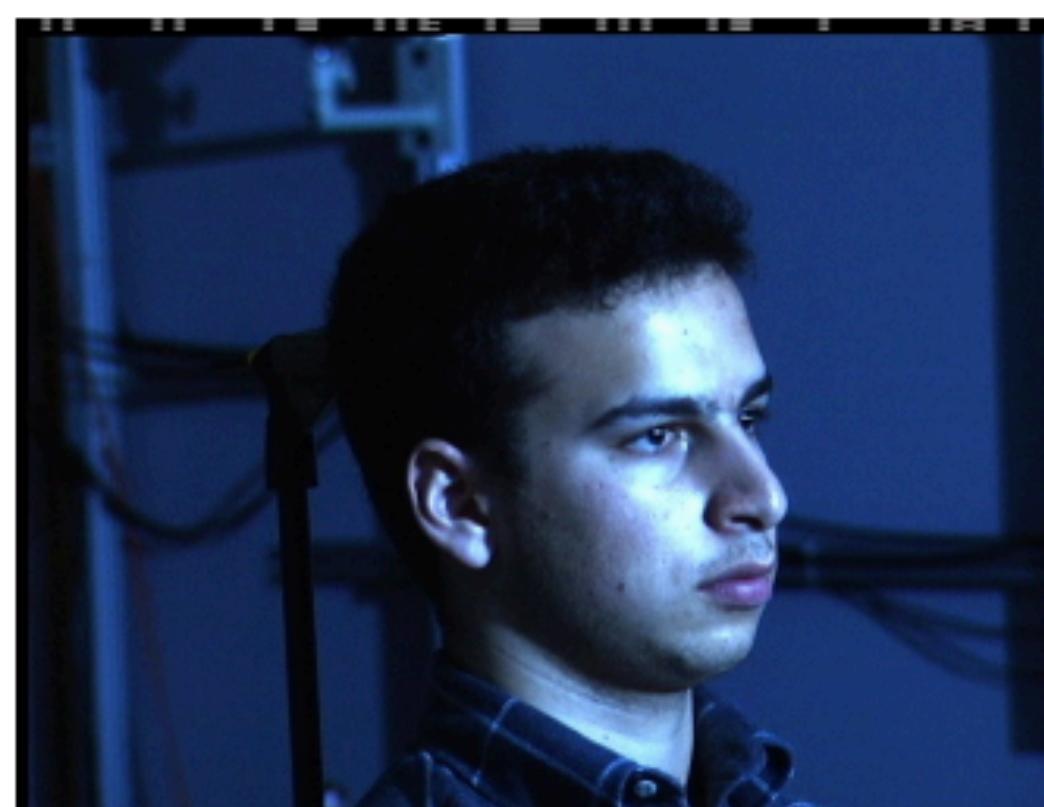
1.33



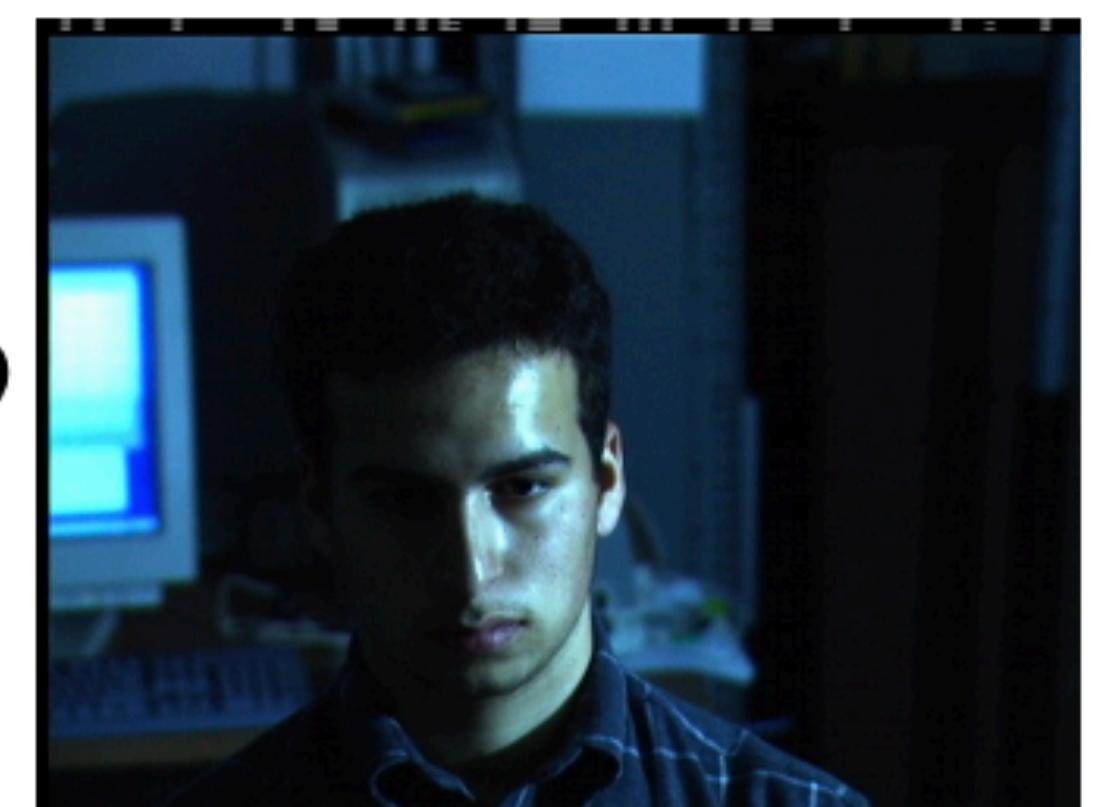
1.33



1.26



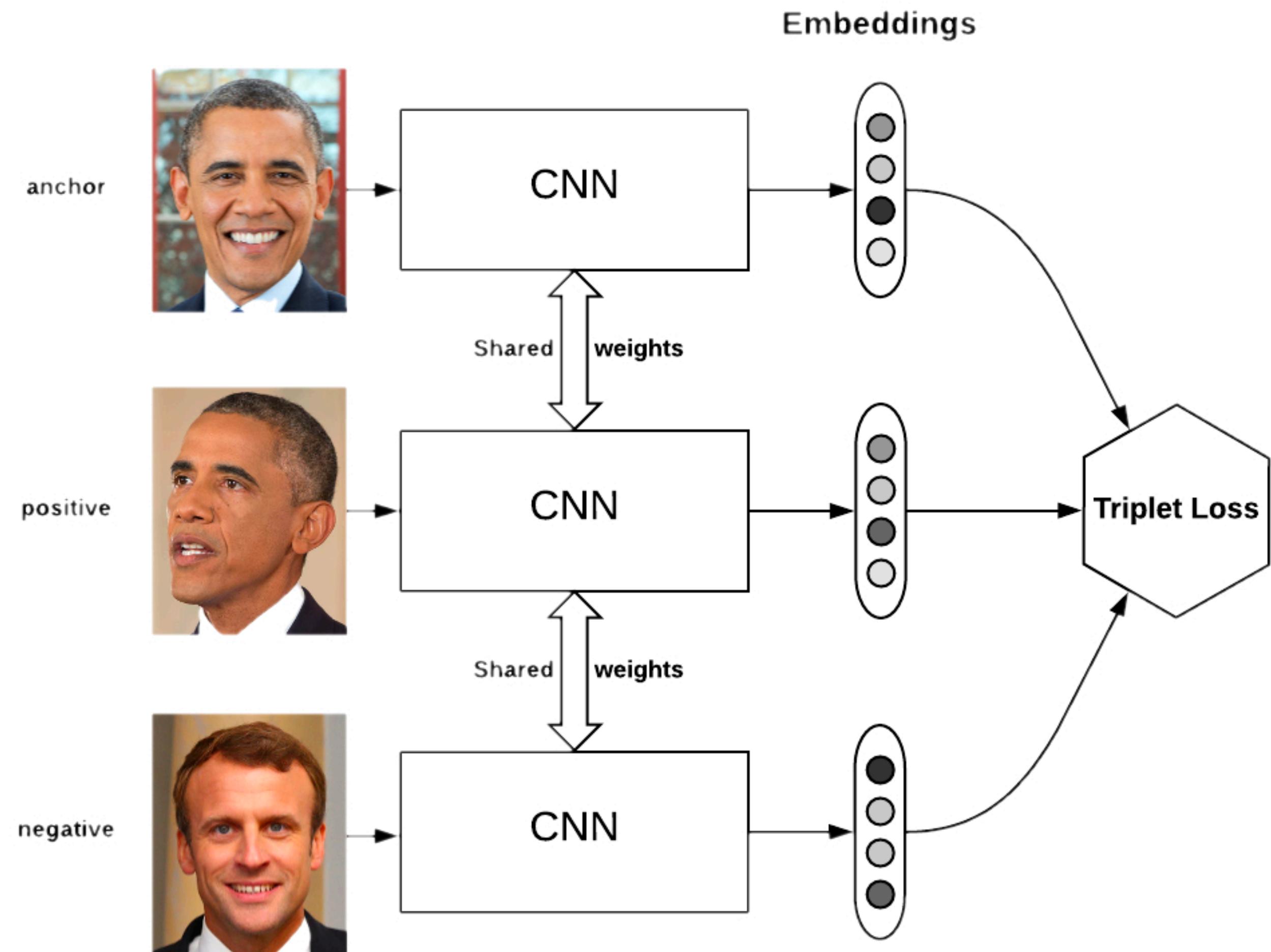
0.99



Triplet Loss

Что мы от него хотим

- Anchor (якорь) - объект класса X класса к которому мы хотим приблизить остальные объекты класса X
- Negative - объект класса Y который мы хотим отдалить от любого якоря класса не Y

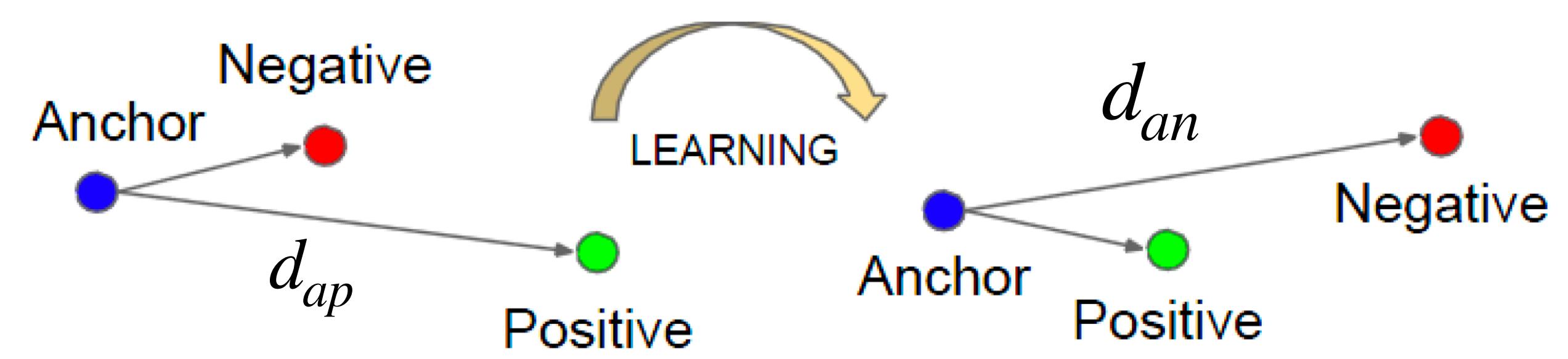


Triplet Loss

Формально

$$L_{triplet} = [d_{ap} - d_{an} + margin]_+$$

- margin - желаемое расстояние между позитивным и негативным примером



Other MeL(Metric Learning) Losses

- Contrastive Loss

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

- Cos Face Loss(Large Margin Cosine Loss)

$$L_{lmc} = \frac{1}{N} \sum_i^N -\log \frac{e^{s \cdot (\cos(\phi_{y_i, i}) - m)}}{e^{s \cdot (\cos(\phi_{y_i, i}) - m)} + \sum_{j \neq y_i} e^{s \cdot \cos(\phi_{y_i, j})}}$$

- Multi Similarity Loss

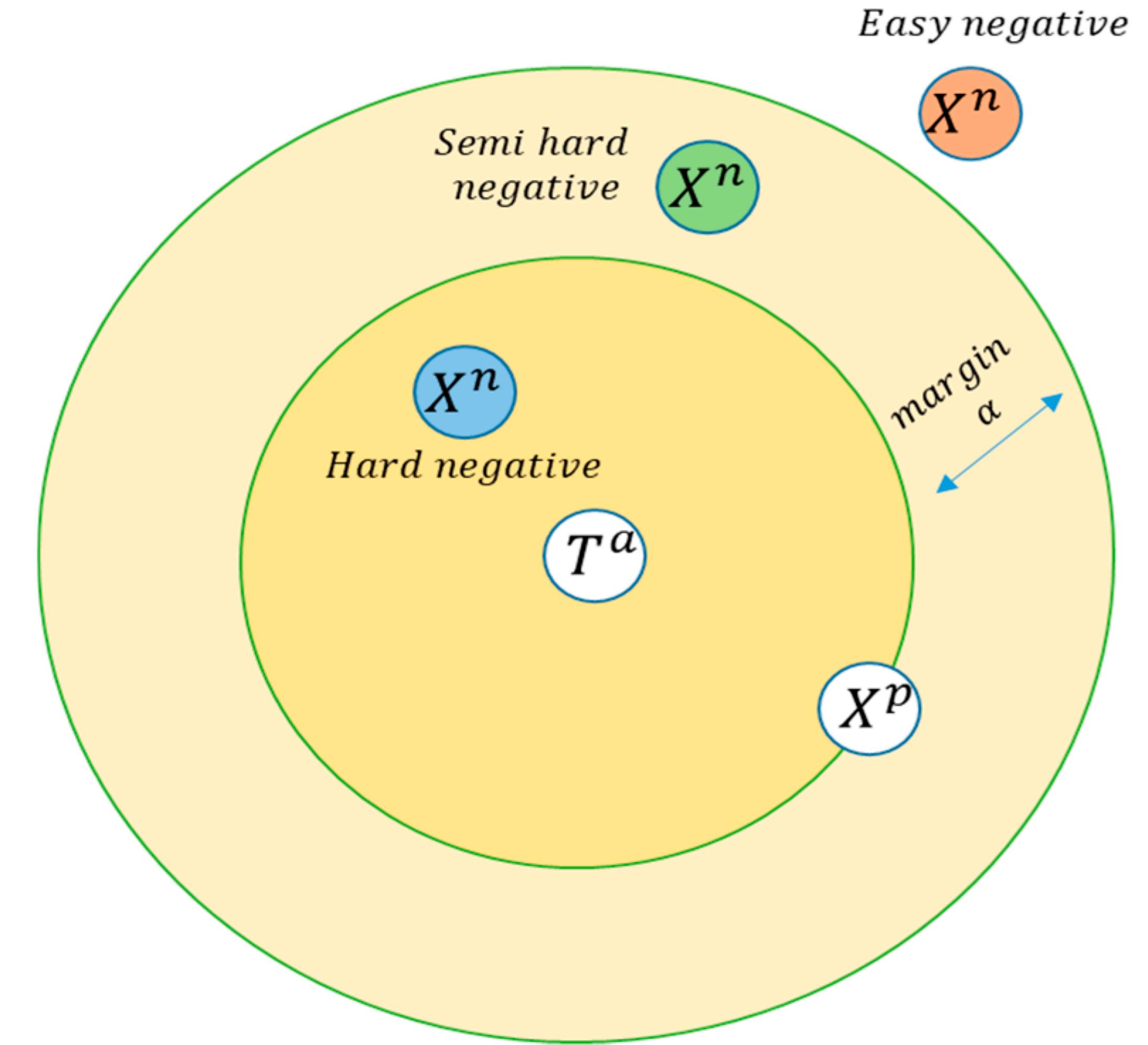
$$L_{MS} = \frac{1}{m} \sum_{i=1}^m \frac{1}{a} \log(1 + \sum_{k \in P_i} e^{-a(S_{ik} - \lambda)}) + \frac{1}{b} \log(1 + \sum_{k \in N_i} e^{b \sum ik - \lambda})$$

- И еще 20+ разных лоссов

Samples Mining

Или просто Miner

- Нет смысла учиться на примерах которые модель итак уже хорошо различает
- 4 стратегии выбора - **hard**, **semihard**, **all**, **easy**
- Использование miner делает loss неинтерпретируемым, тк изменяет выборку



T_a - anchor

X_p - самый удаленный от якоря позитивный элемент

X_n - негативный элемент

Результаты Face Net

- 95.12% accuracy на YouTube Faces DB против 93.2 у предидущего SOTA подхода

#dims	VAL
64	86.8% \pm 1.7
128	87.9% \pm 1.9
256	87.7% \pm 1.9
512	85.6% \pm 2.0

Зависимость качества от размера эмбединга

#training images	VAL
2,600,000	76.3%
26,000,000	85.1%
52,000,000	85.1%
260,000,000	86.2%

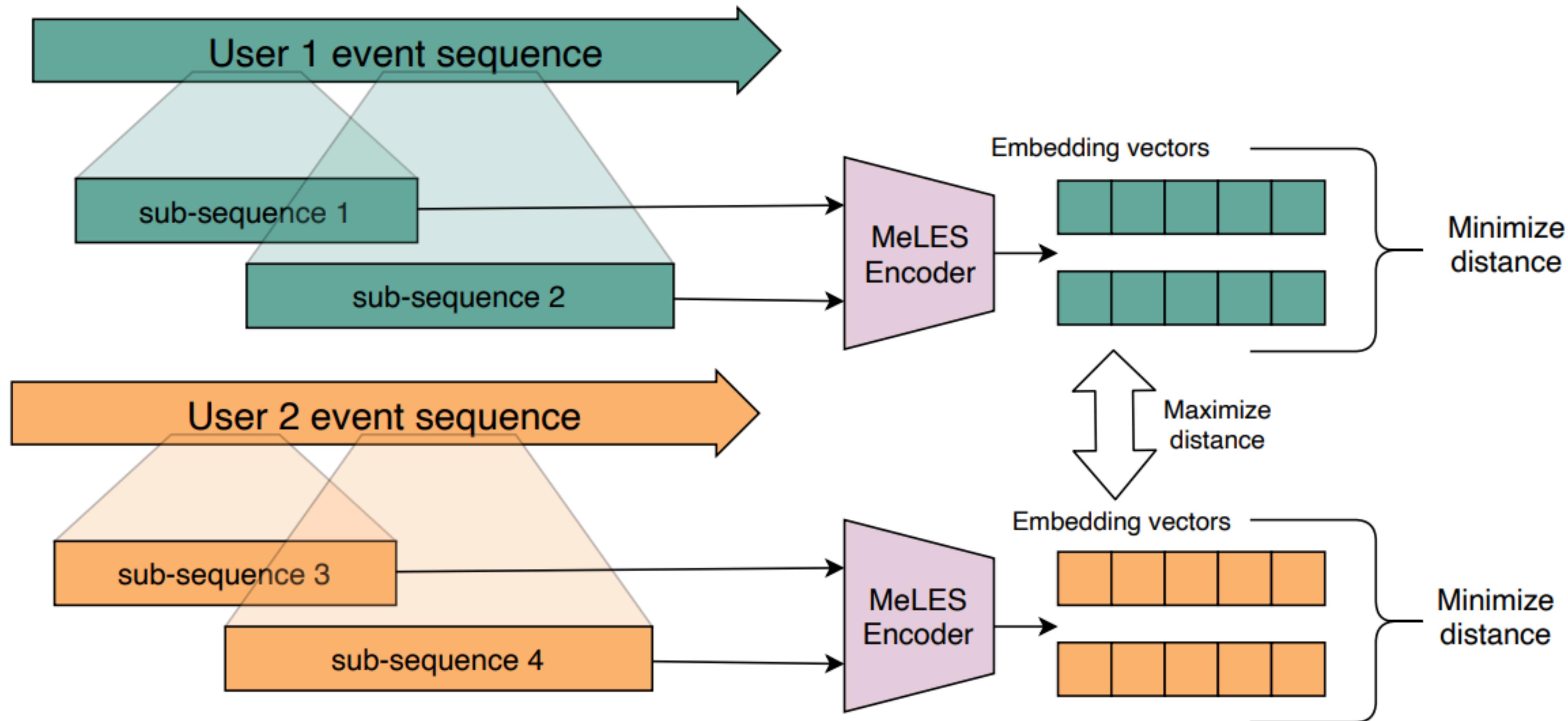
Зависимость качества от количества данных

architecture	VAL
NN1 (Zeiler&Fergus 220×220)	87.9% \pm 1.9
NN2 (Inception 224×224)	89.4% \pm 1.6
NN3 (Inception 160×160)	88.3% \pm 1.7
NN4 (Inception 96×96)	82.0% \pm 2.3
NNS1 (mini Inception 165×165)	82.4% \pm 2.4
NNS2 (tiny Inception 140×116)	51.9% \pm 2.9

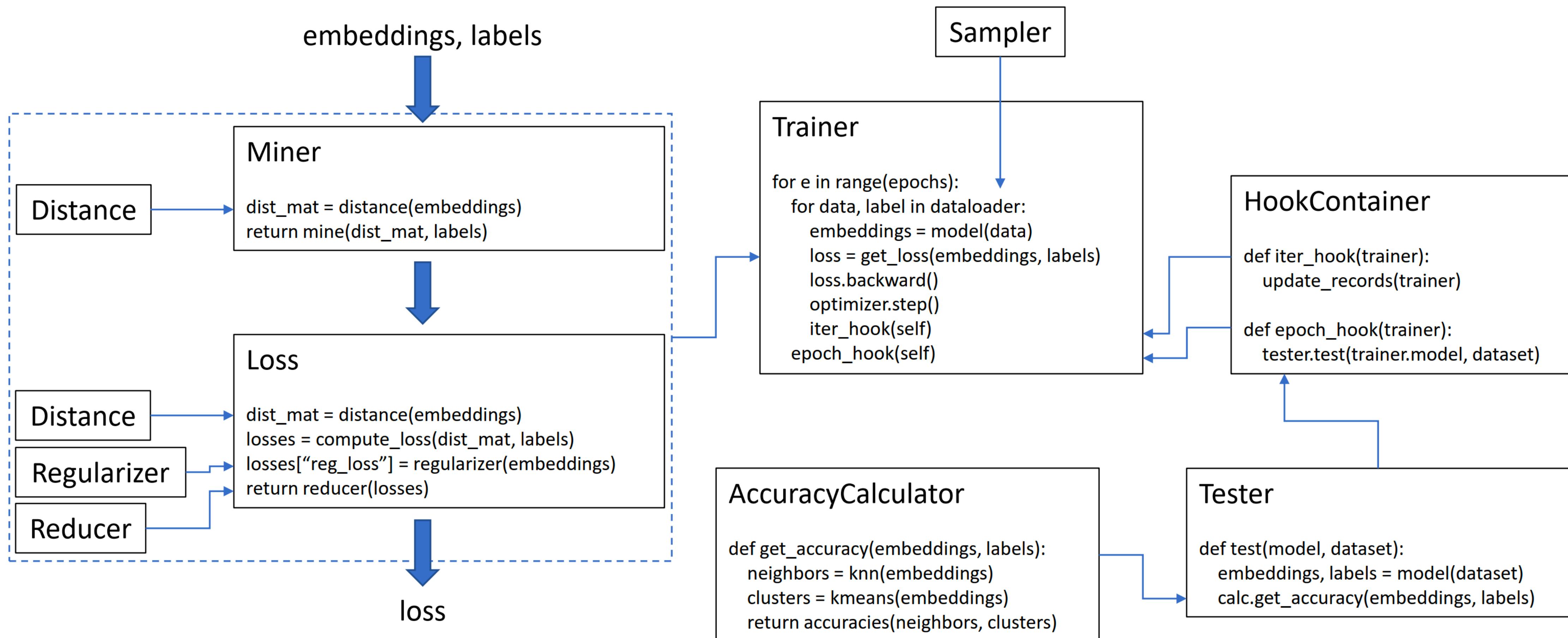
Зависимость качества от архитектуры

MeLES(Metric Learning for Event Sequences)

Модификация подхода для объектов заданных последовательностью событий



Весь пайплайн еще раз



Результаты MeLES

На задачах Age и Gender Prediction

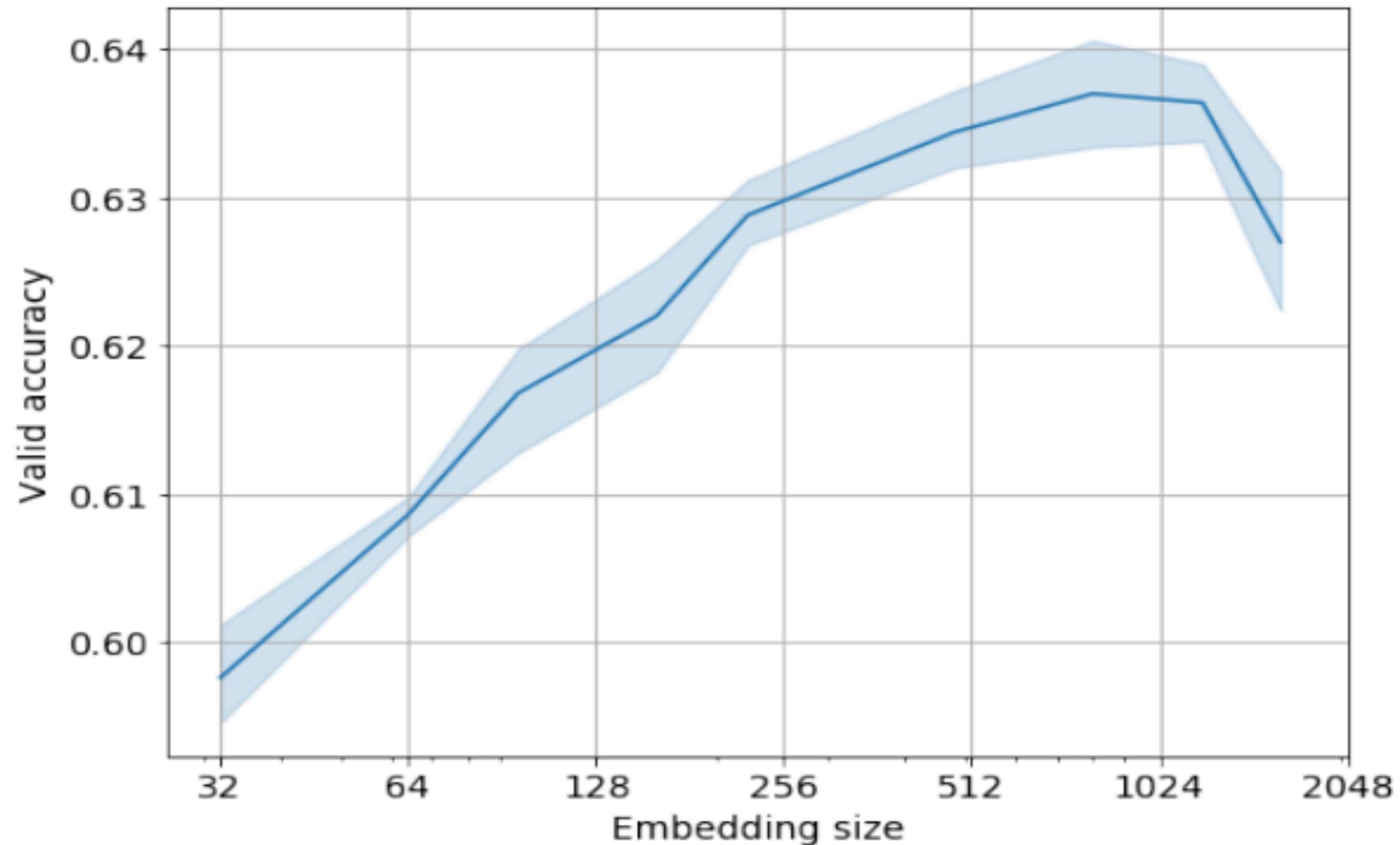


Рисунок 1. Зависимость качества от размера эмбединга

Pair generation method	Age, Accuracy ±95%	Gender, AUROC ±95%
Random samples	0.628 ± 0.003	0.851 ± 0.004
Random disjoint samples	0.608 ± 0.004	0.836 ± 0.008
Random slices	0.639 ± 0.006	0.872 ± 0.005

Таблица 1. Сравнение метода сэмплинга последовательности

Loss type	Age, Accuracy ±95%	Gender, AUROC ±95%
Contrastive loss	0.639 ± 0.006	0.871 ± 0.003
Binomial deviance loss	0.535 ± 0.005	0.853 ± 0.005
Histogram loss	0.642 ± 0.002	0.851 ± 0.004
Margin loss	0.631 ± 0.003	0.871 ± 0.004
Triplet loss	0.610 ± 0.006	0.855 ± 0.003

Таблица 2. Сравнение функций потерь

Negative sampling strategy	Age, Accuracy ±95%	Gender, AUROC ±95%
Hard negative mining	0.637 ± 0.005	0.872 ± 0.004
Random negative sampling	0.615 ± 0.005	0.826 ± 0.004
Distance weighted sampling	0.620 ± 0.003	0.867 ± 0.003

Таблица 3. Сравнение метода отбора триплетов для обучения

Результаты MeLES

Сравнение архитектур энкодера и других моделей

Method	Age, Accuracy $\pm 95\%$	Gender, AUROC $\pm 95\%$
LightGBM on hand-crafted features	0.626 ± 0.004	0.875 ± 0.004
LightGBM on MeLES embeddings	0.639 ± 0.006	0.872 ± 0.005
LightGBM on both hand-crafted features and MeLES embeddings	0.643 ± 0.009	0.882 ± 0.003
Supervised learning	0.631 ± 0.010	0.871 ± 0.007
MeLES fine-tuning	0.643 ± 0.007	0.888 ± 0.002
LightGBM on CPC embeddings	0.595 ± 0.004	0.848 ± 0.004
Fine-tuned Contrastive Predictive Coding	0.621 ± 0.007	0.873 ± 0.007

Таблица 1. Сравнение моделей

Econder type	Age, Accuracy $\pm 95\%$	Gender, AUROC $\pm 95\%$
LSTM	0.620 ± 0.003	0.870 ± 0.005
GRU	0.639 ± 0.006	0.871 ± 0.004
Transformer	0.621 ± 0.001	0.848 ± 0.002

Таблица 2. Сравнение энкодеров

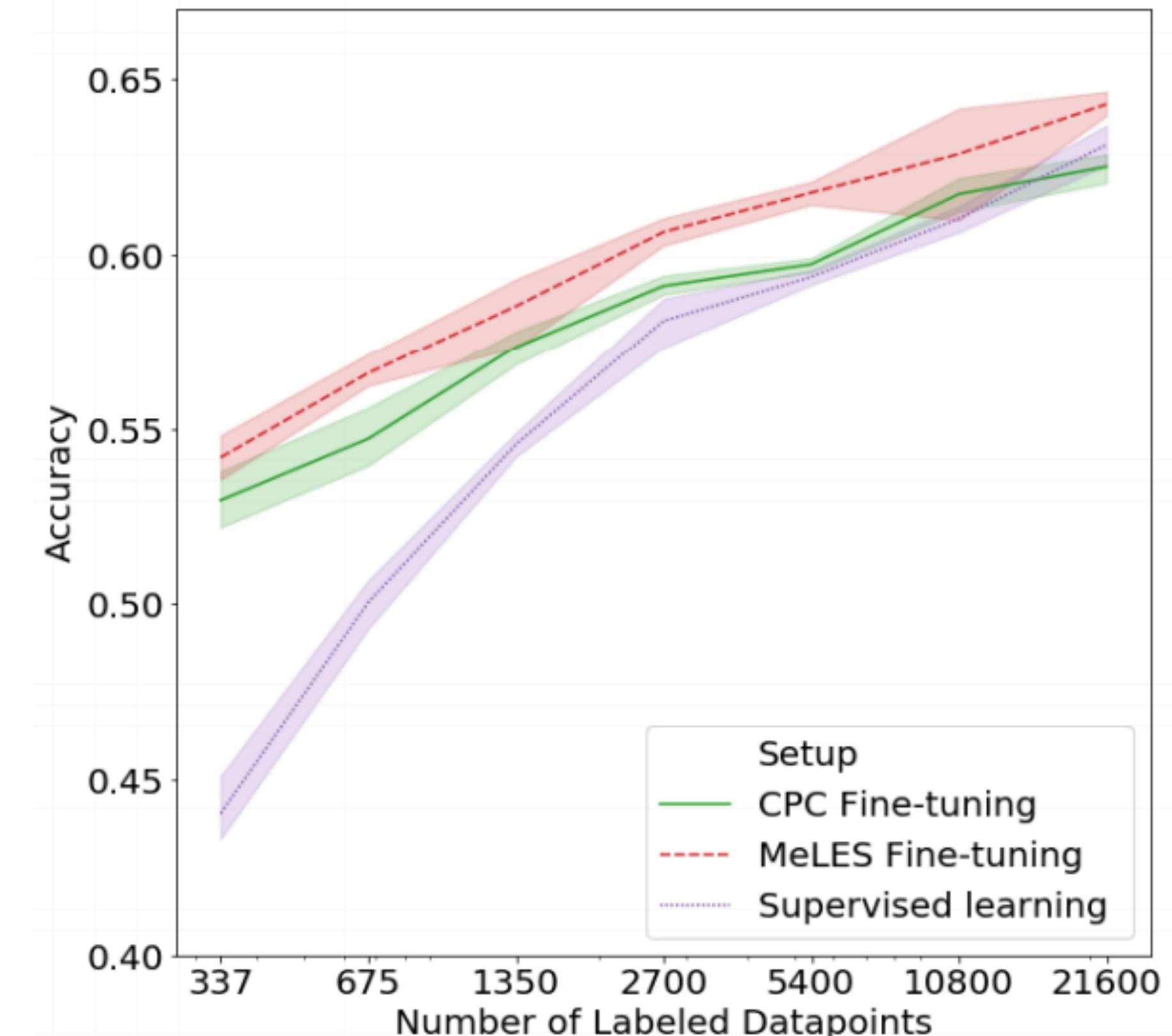
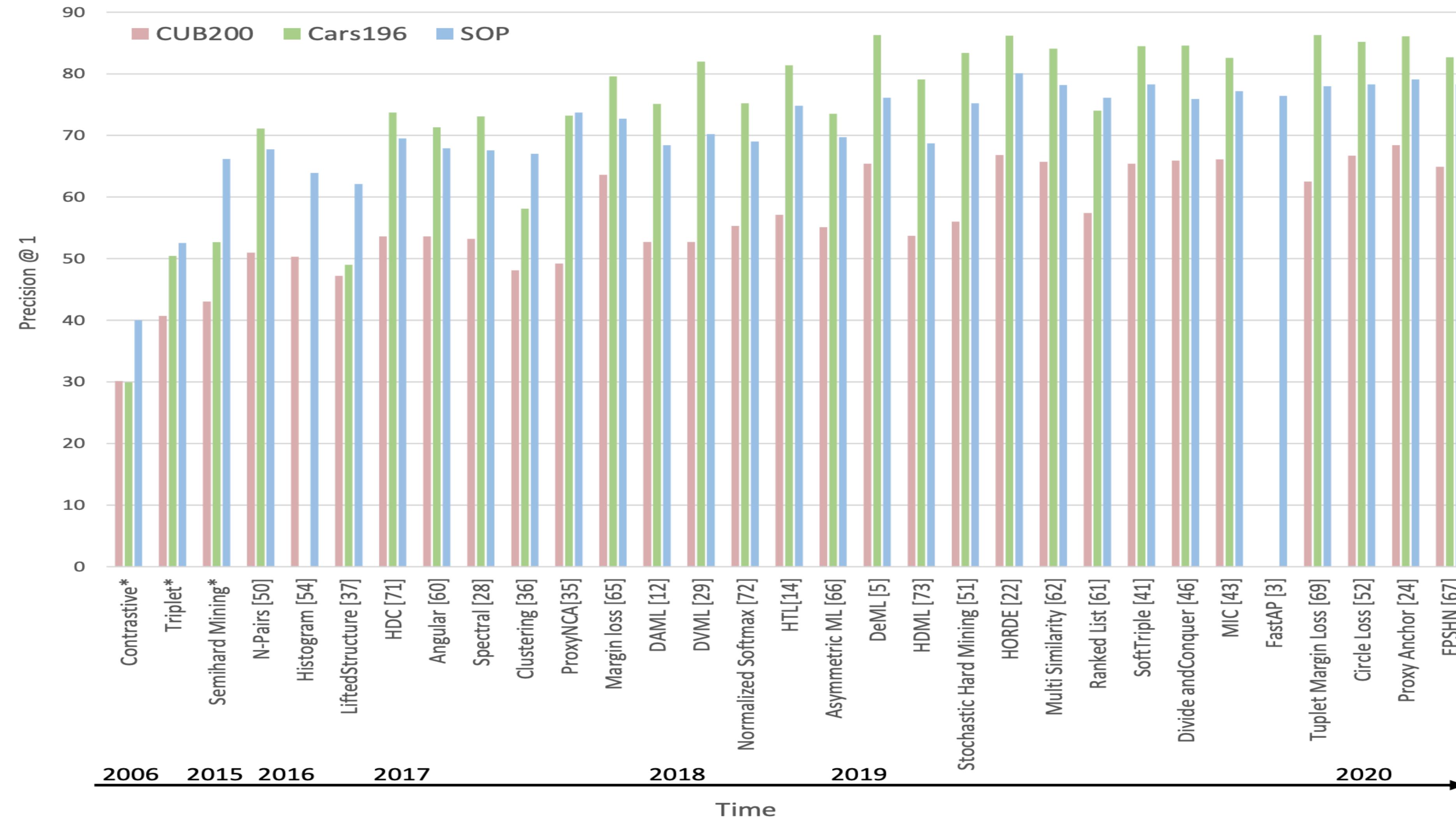


Рисунок 1. Зависимость качества от количества
размеченных данных

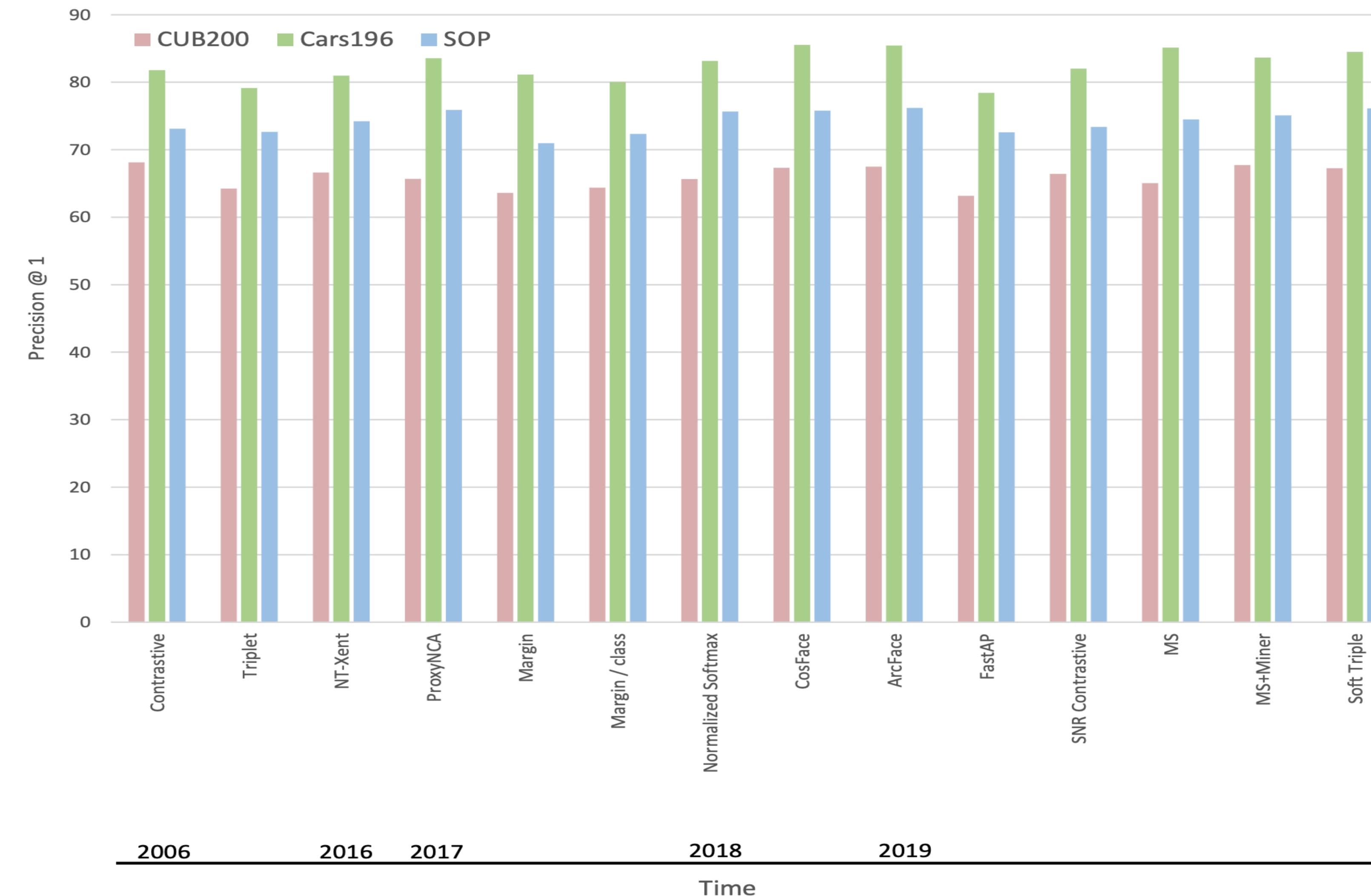
Какой metric learning loss использовать ?

По статьям



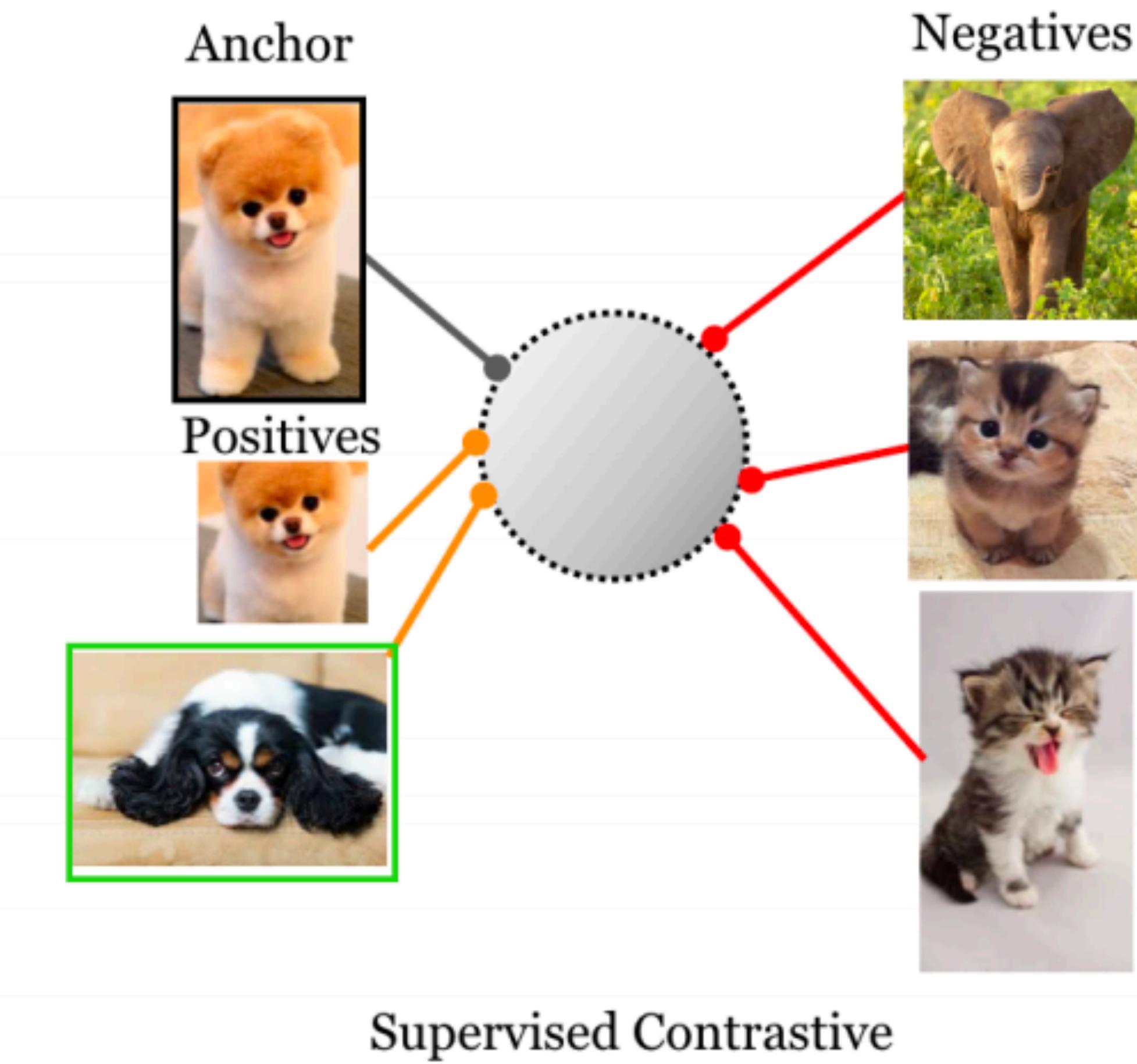
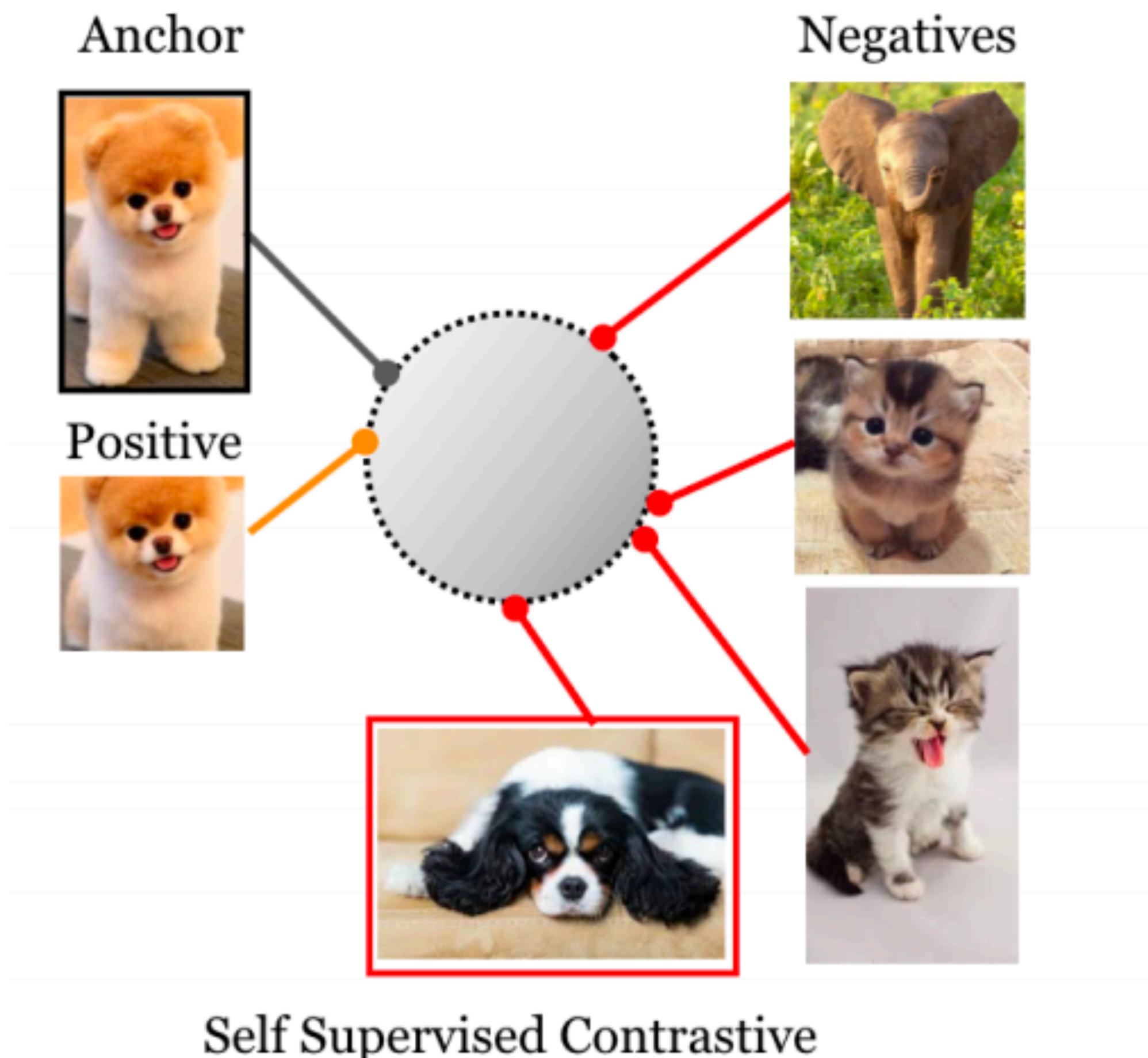
Какой metric learning loss использовать ?

По факту



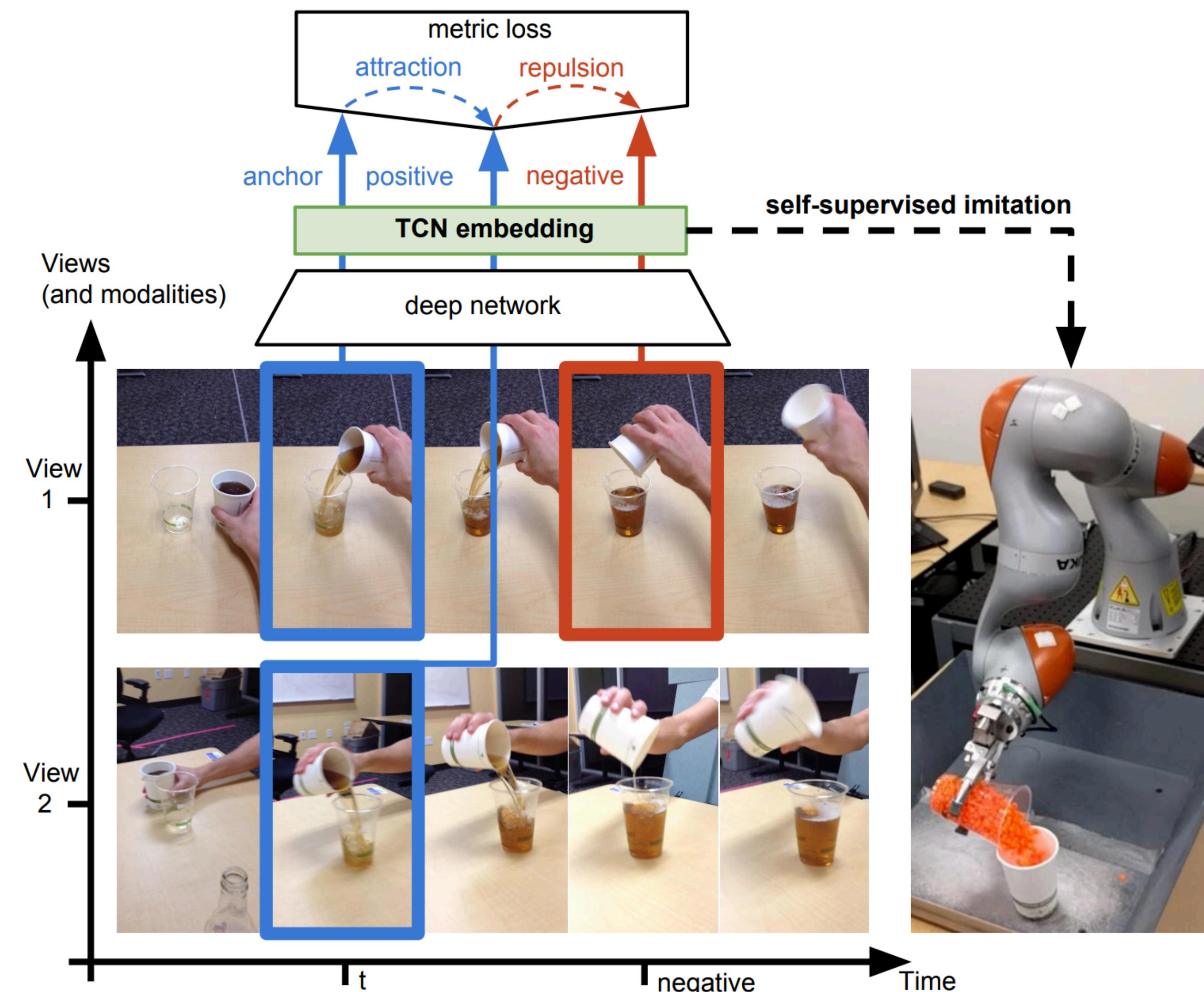
Где еще можно использовать Metric Learning

Image Classification



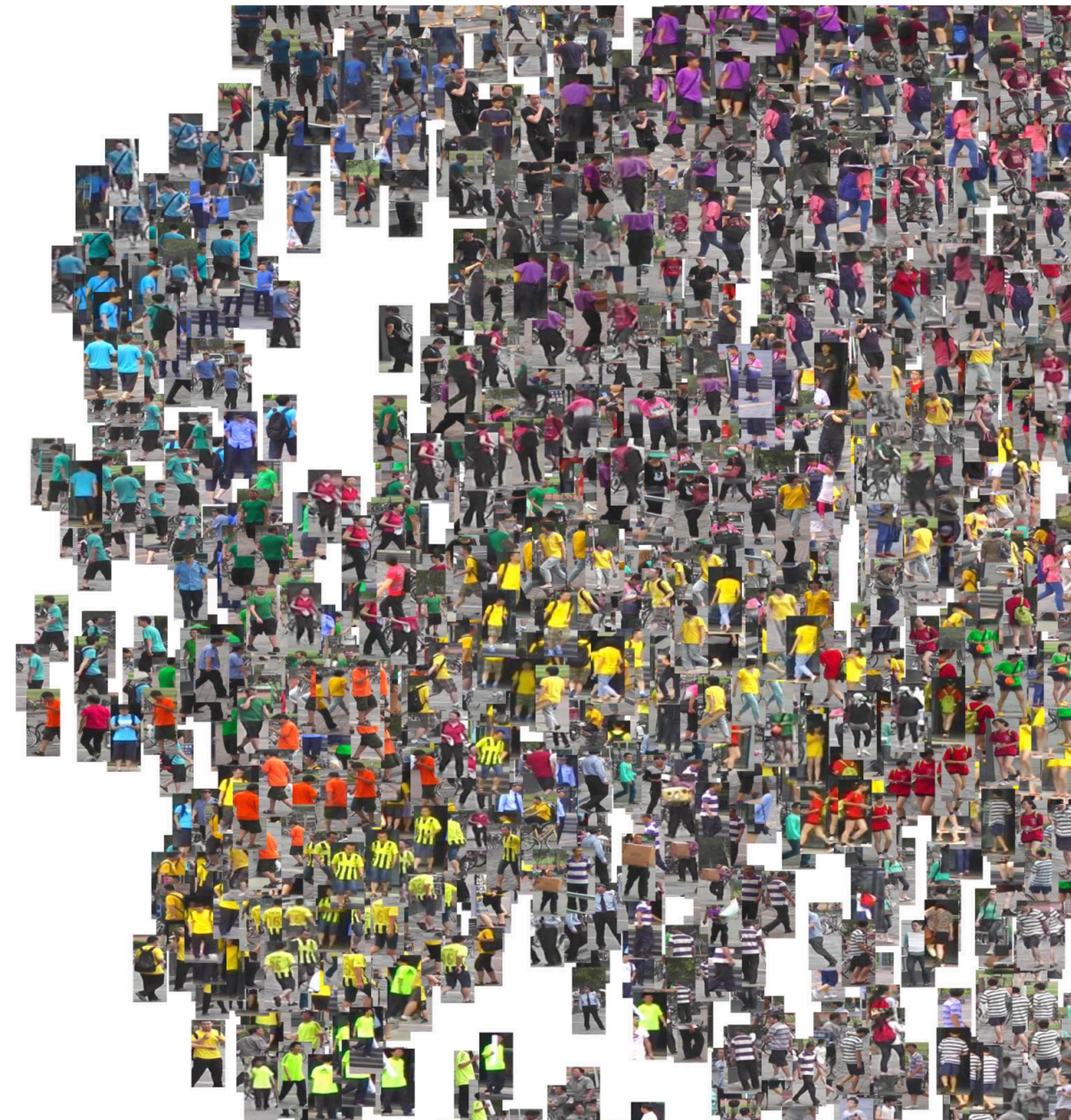
Где еще можно использовать Metric Learning

Imitation Learning



Где еще можно использовать Metric Learning

Person Re-Identification



Где еще можно использовать Metric Learning

Dimensionality reduction

Original (AUC, GS)



(a)

t-SNE (**0.18**, 0.18)



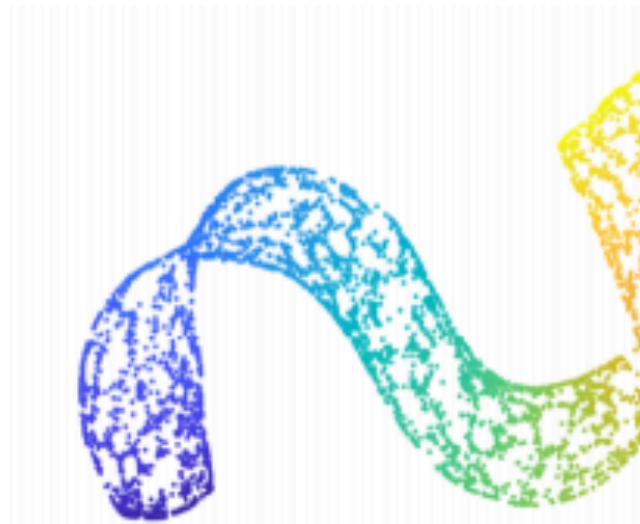
(b)

UMAP (0.16, 0.13)



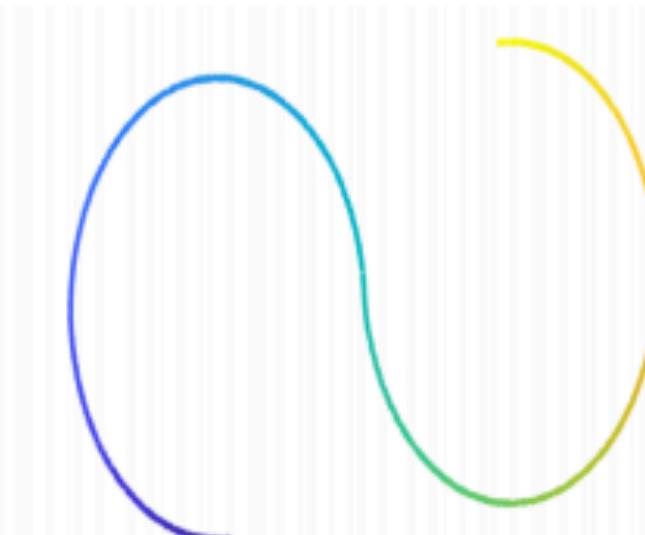
(c)

TriMap (0.15, **0.80**)



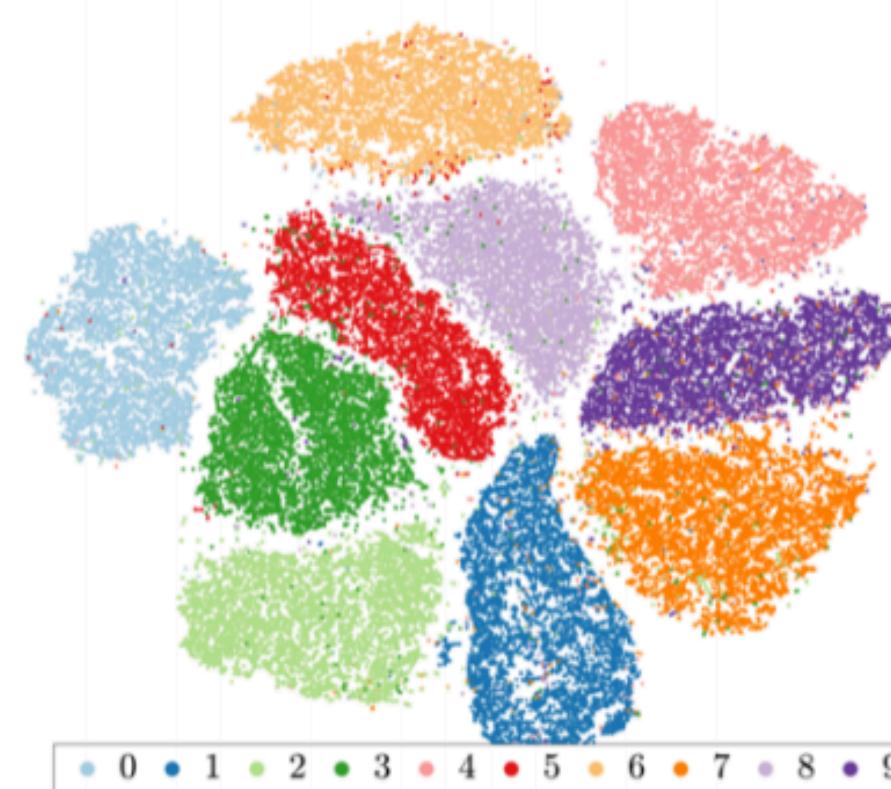
(d)

PCA (0.03, 1.00)



(e)

MNIST - t-SNE (0.97, 0.90)

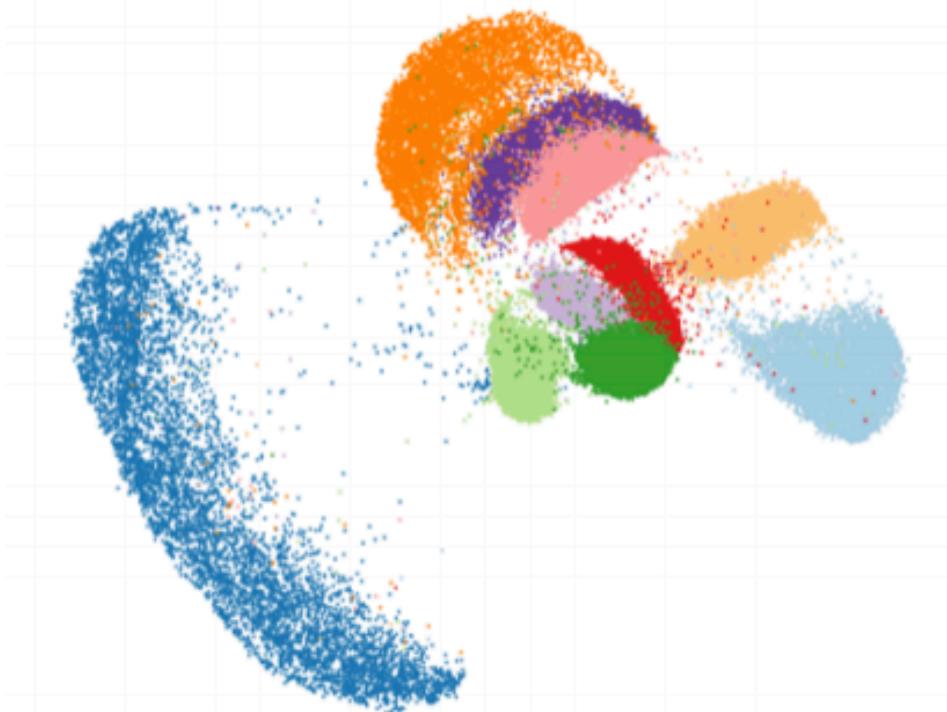


• 0 • 1 • 2 • 3 • 4 • 5 • 6 • 7 • 8 • 9

MNIST - UMAP (0.95, 0.91)



MNIST - TriMap (0.94, 0.92)

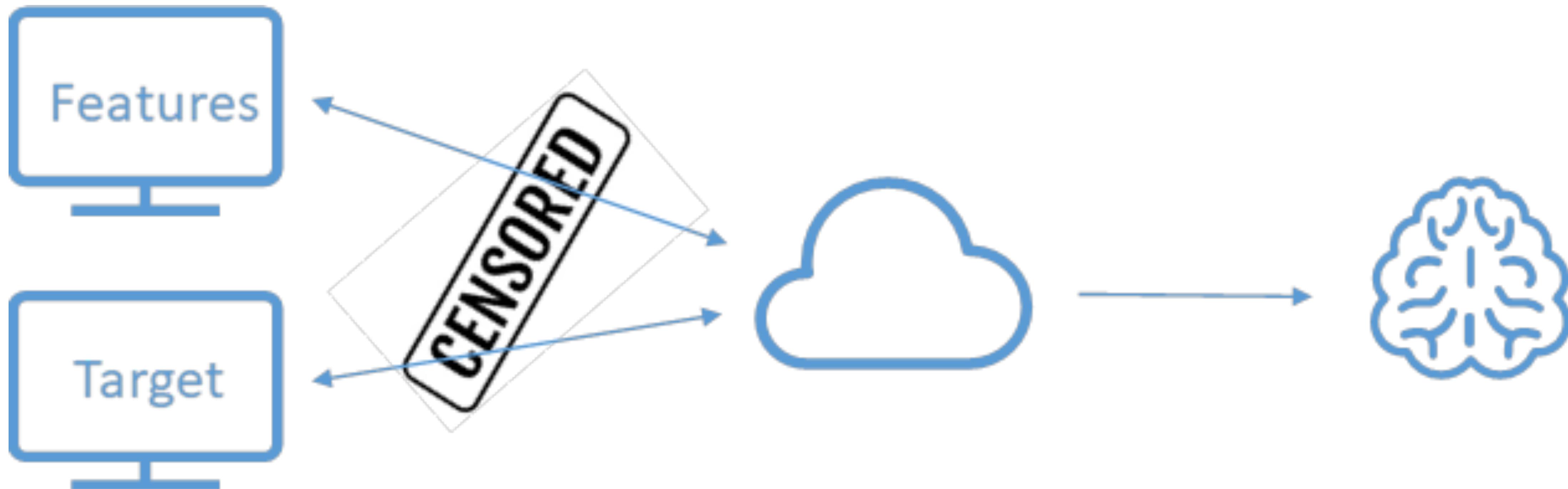


MNIST - PCA (0.38, 1.00)



Где еще можно использовать Metric Learning

Multi-Partial Computations



Итоги

- Metric Learning - self-supervised подход позволяющие получить качественные осмысленные эмбединги наших данных
- Особенно эффективно можно использовать в задачах в которых неприменима энтропийная функция потерь(FaceNet и подобные)
- Лучше всего для начала использовать Contrastive или Triplet Margin Loss
- Лучше всего использовать библиотеку pyTorch Metric Learning для своих экспериментов
- Вопрос эффективного и честного подбора гиперпараметров MeL не решен и исследования в этой области ведутся