Deep Metric Learning using Similarities from Nonlinear Rank Approximations

Konstantin Schall, Kai Uwe Barthel, Nico Hezel, and Klaus Jung

Visual Computing Group - HTW Berlin

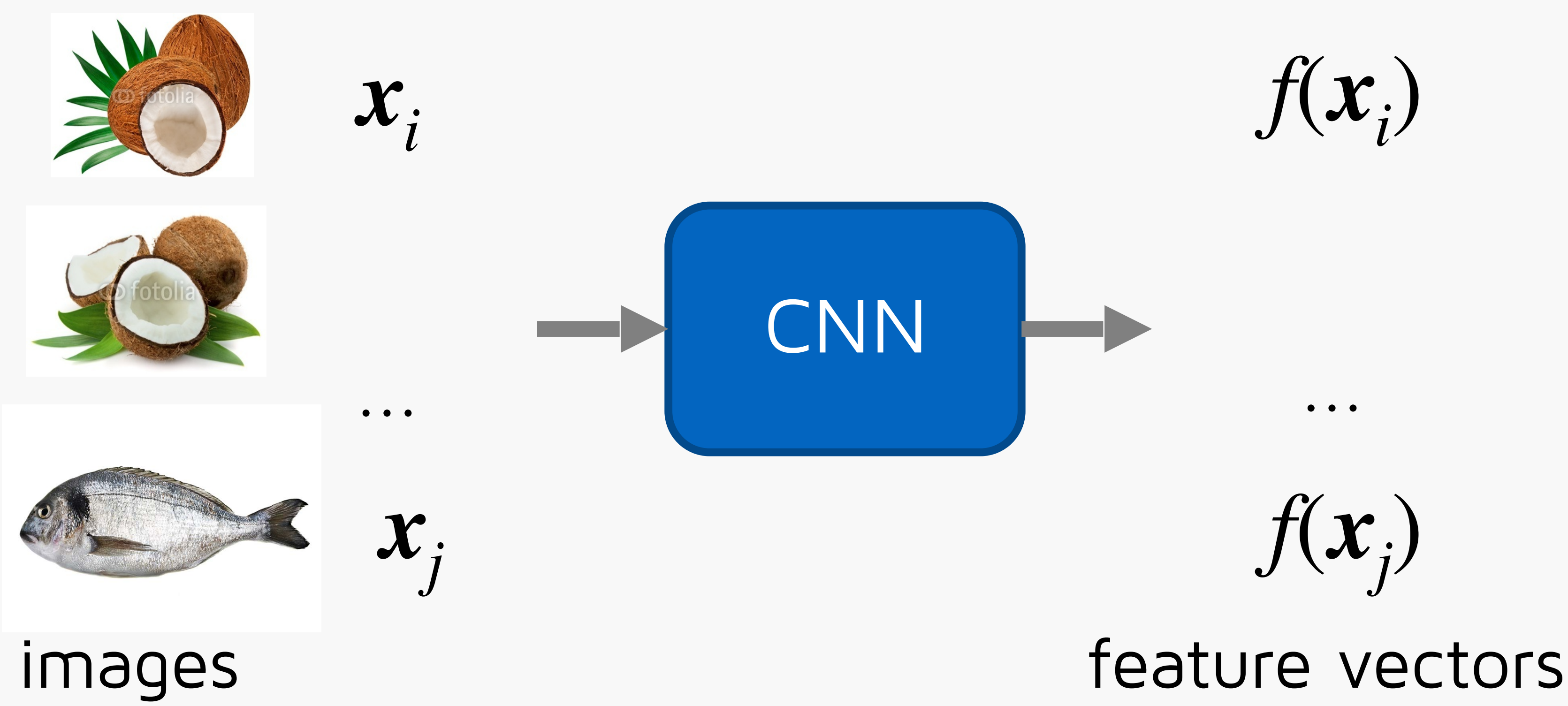
visual-computing.com

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

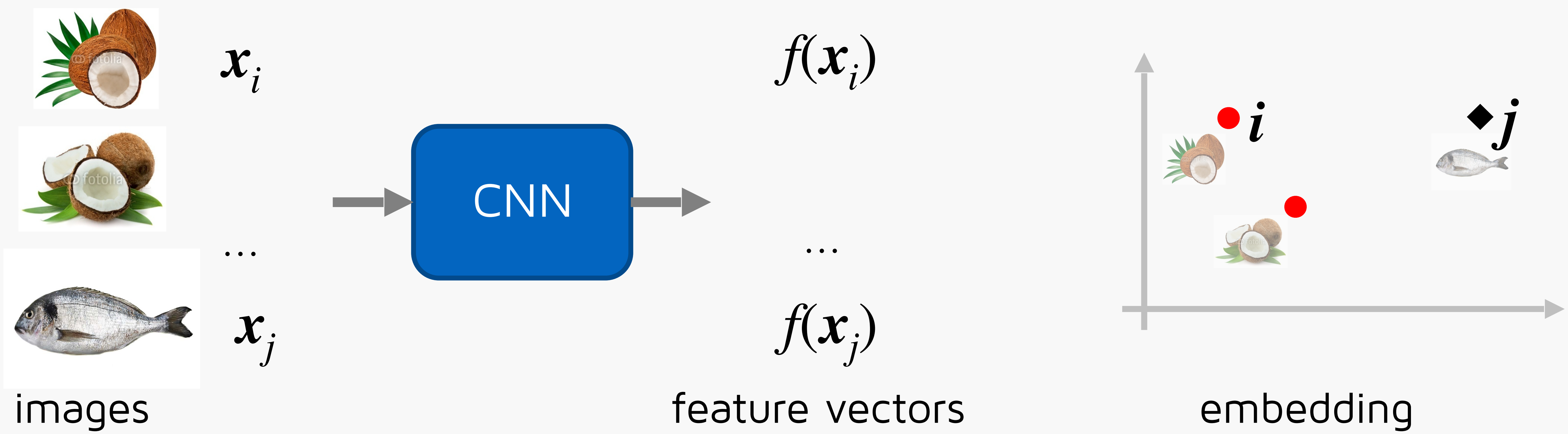
Feature embedding by CNNs

- Convolutional Neural Networks map images $\boldsymbol{x}$ to high dimensional feature vectors $f(\boldsymbol{x})$



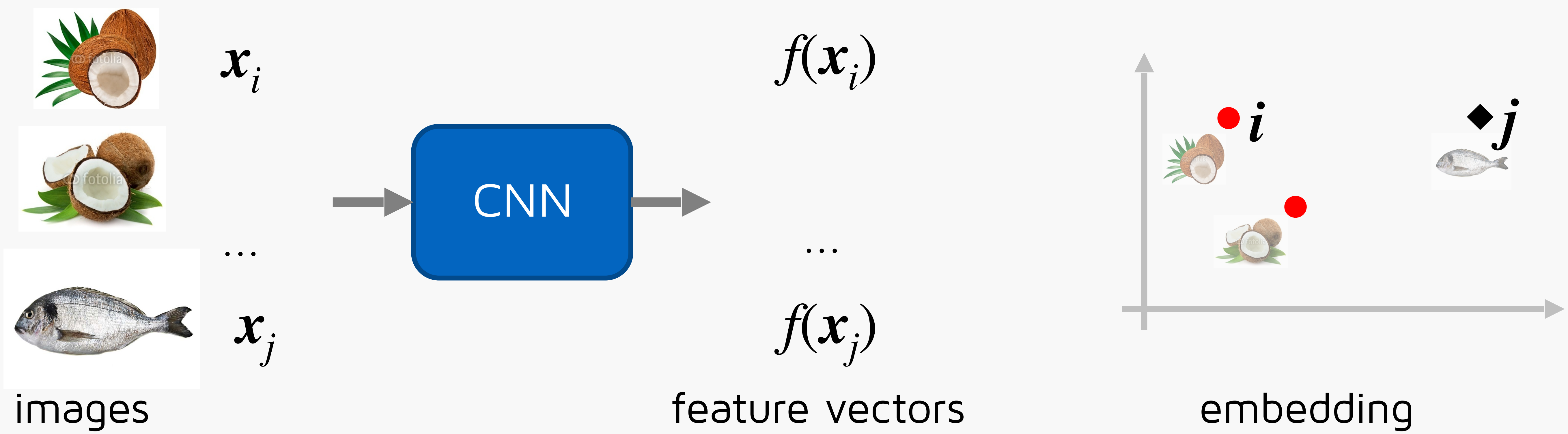
Feature embedding by CNNs

- Convolutional Neural Networks map images $\boldsymbol{x}$ to high dimensional feature vectors $f(\boldsymbol{x})$



Feature embedding by CNNs

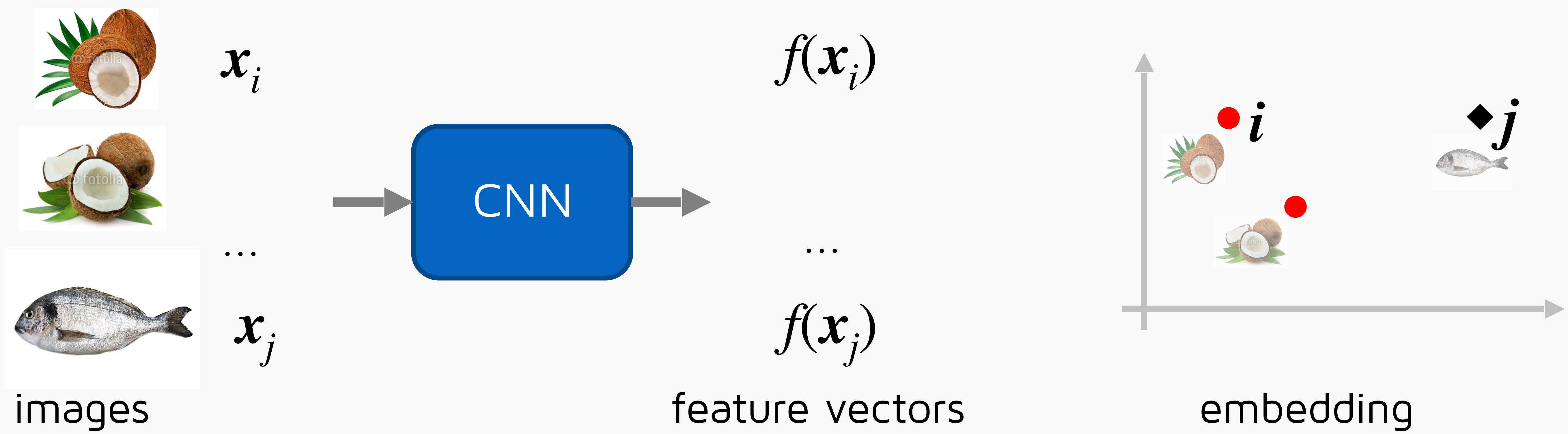
- Convolutional Neural Networks map images $\boldsymbol{x}$ to high dimensional feature vectors $f(\boldsymbol{x})$



- Similar images have similar feature vectors (FVs)

Feature embedding by CNNs

- Convolutional Neural Networks map images $\boldsymbol{x}$ to high dimensional feature vectors $f(\boldsymbol{x})$

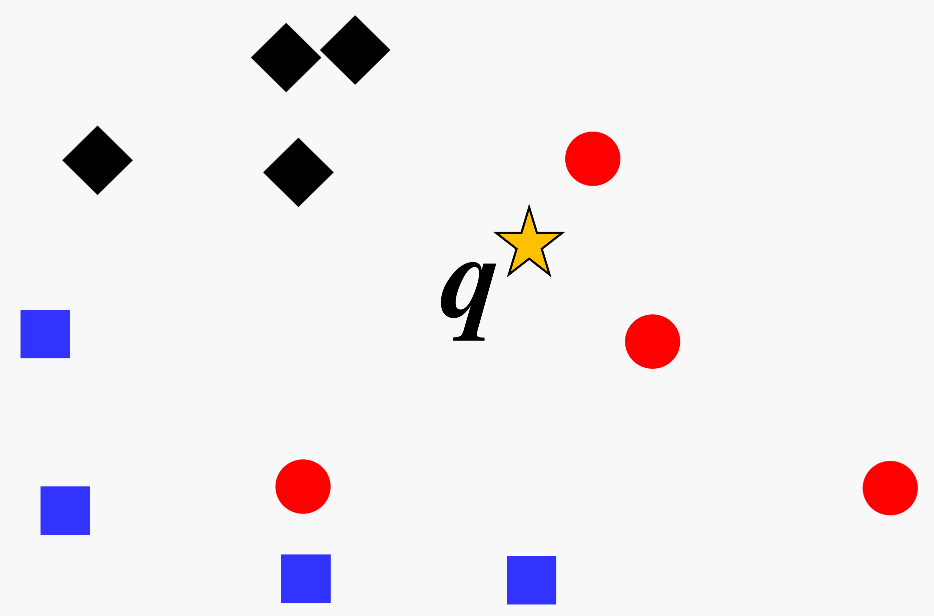


- Similar images have similar feature vectors (FVs)

  ➥ Embeddings can be used for similarity search

Deep Metric Learning for Image Retrieval

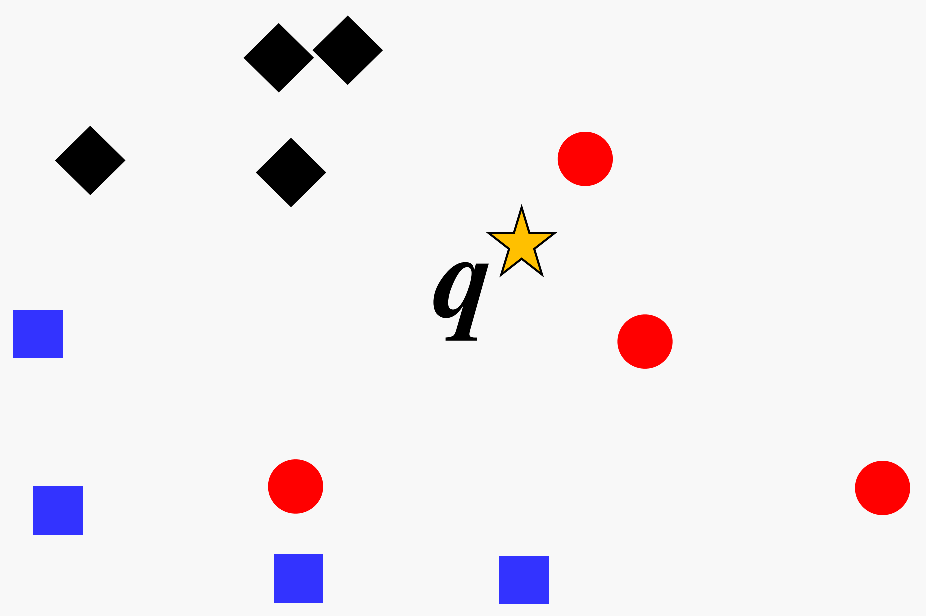
- **Image similarity search:** Retrieve feature vectors with small distances to a query vector $\boldsymbol{q}$.



Toy embedding of 12 FVs
(3 classes with 4 FVs)

Deep Metric Learning for Image Retrieval

- **Image similarity search:** Retrieve feature vectors with small distances to a query vector $\boldsymbol{q}$.

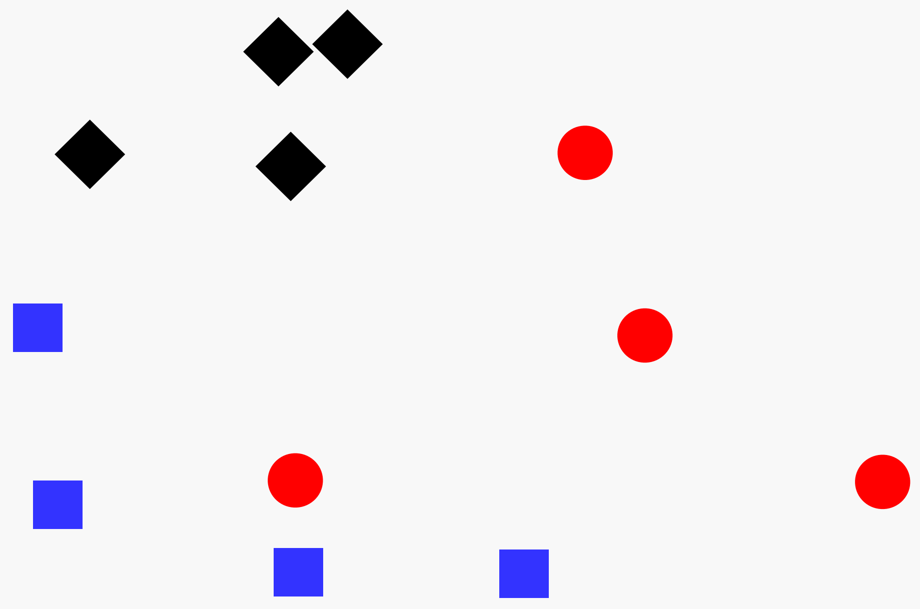


Toy embedding of 12 FVs
(3 classes with 4 FVs)

- **Deep metric learning based loss functions:** Try to optimize the embedding by enforcing distances between vectors of the same class to be smaller than to different classes.

Deep Metric Learning based Loss Functions J

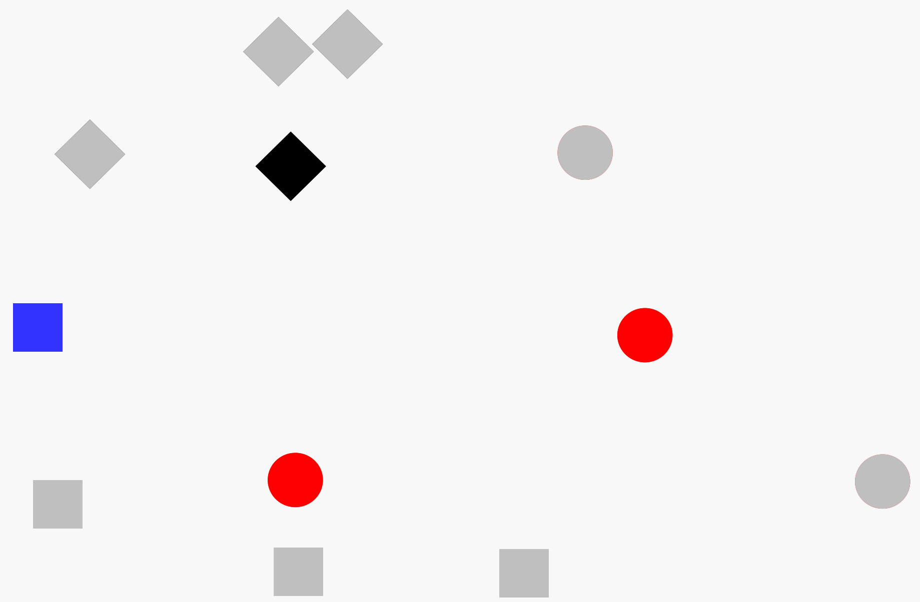
Properties & problems of previously proposed Loss Functions



- Contrastive Loss
- Triplet Loss
- Lifted Structured Loss
- N-Pair Loss
- ...
- **Nonlinear Rank Approximation Loss**

Deep Metric Learning based Loss Functions J

Properties & problems of previously proposed Loss Functions



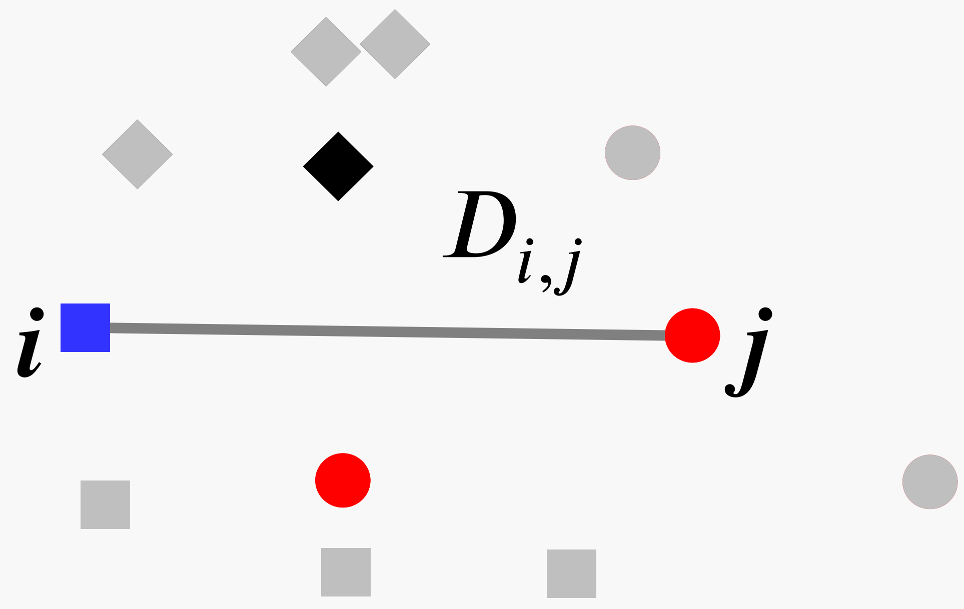
(non gray FVs belong to a particular batch)

- Contrastive Loss
- Triplet Loss
- Lifted Structured Loss
- N-Pair Loss
- ...
- **Nonlinear Rank Approximation Loss**

Deep Metric Learning based Loss Functions J

Properties & problems of previously proposed Loss Functions

- Contrastive Loss
- Triplet Loss
- Lifted Structured Loss
- N-Pair Loss
- …
- **Nonlinear Rank Approximation Loss**

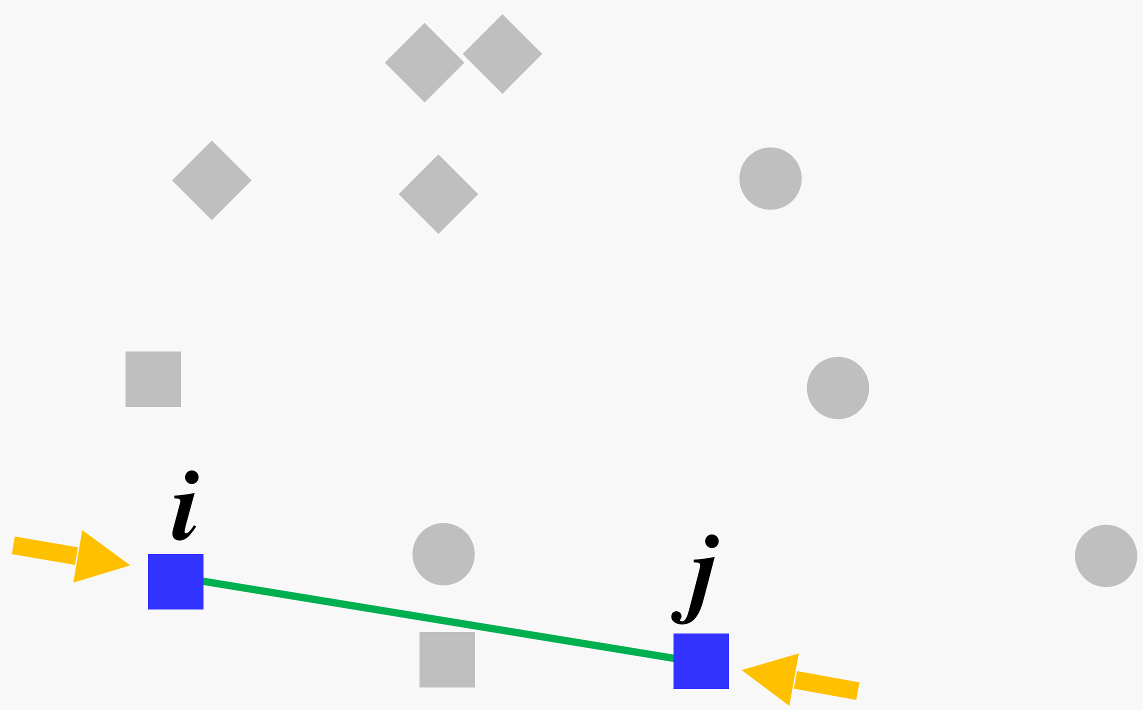


(non gray FVs belong to a particular batch)

Contrastive Loss

Only uses two samples (feature vectors)

$$J_{i,j} = \tilde{y}_{i,j} \underline{D_{i,j}^2} + (1 - \tilde{y}_{i,j}) \max\left(0, \alpha - D_{i,j}\right)^2$$

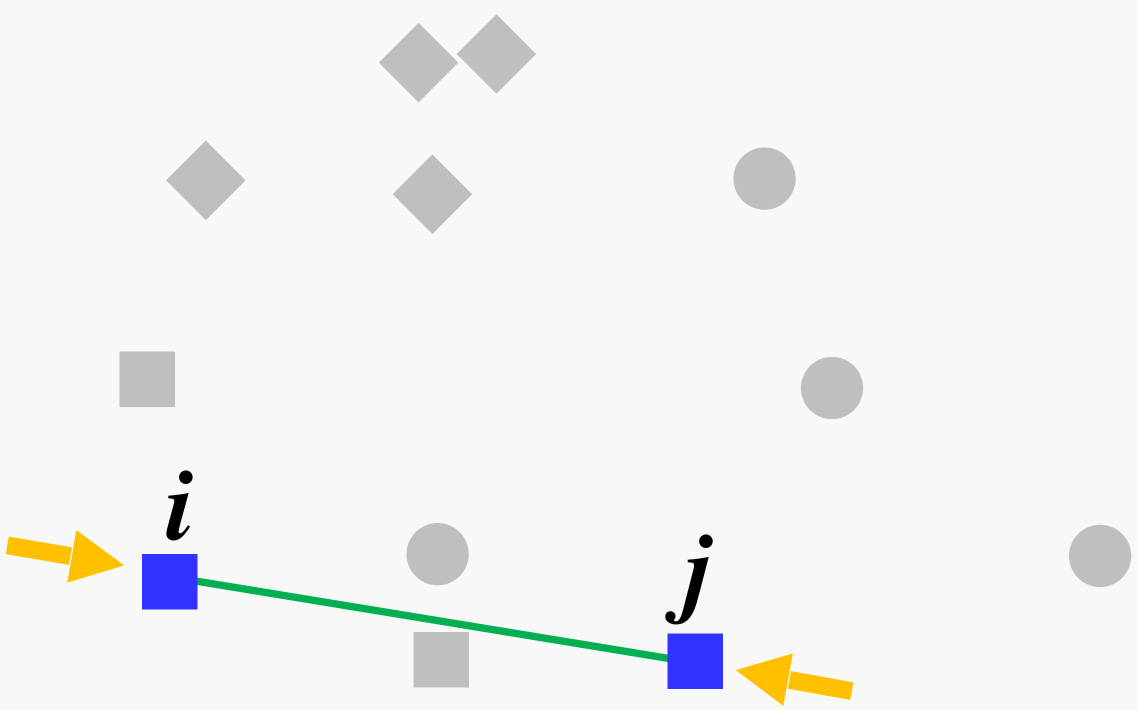


same class: $\tilde{y}_{i,j} = 1$

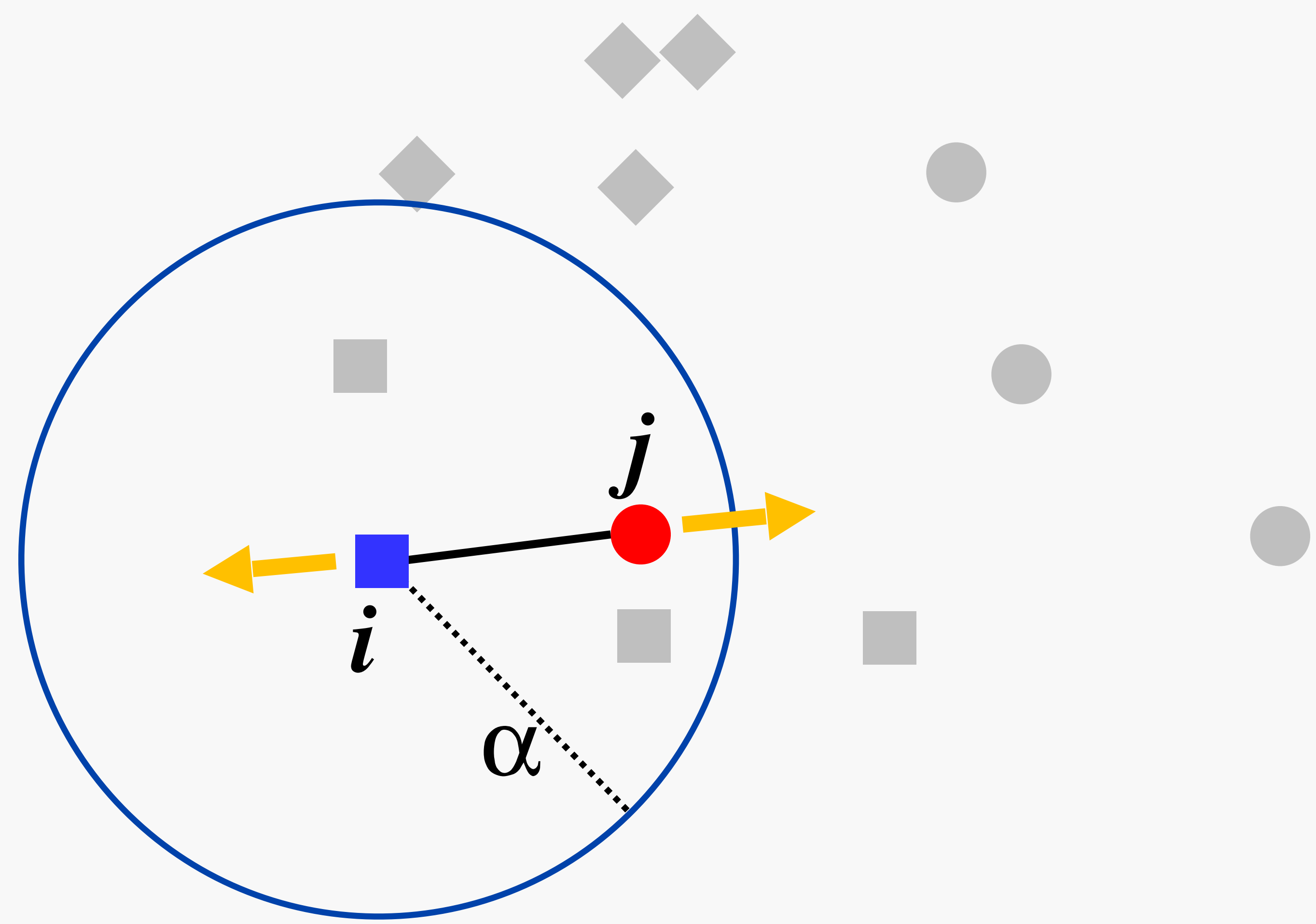
Contrastive Loss

Only uses two samples (feature vectors)

$$J_{i,j} = \tilde{y}_{i,j} \underline{D_{i,j}^2} + (1 - \tilde{y}_{i,j}) \max\left(0, \alpha - \underline{D_{i,j}}\right)^2$$



same class: $\tilde{y}_{i,j} = 1$



different class: $\tilde{y}_{i,j} = 0$

Triplet Loss

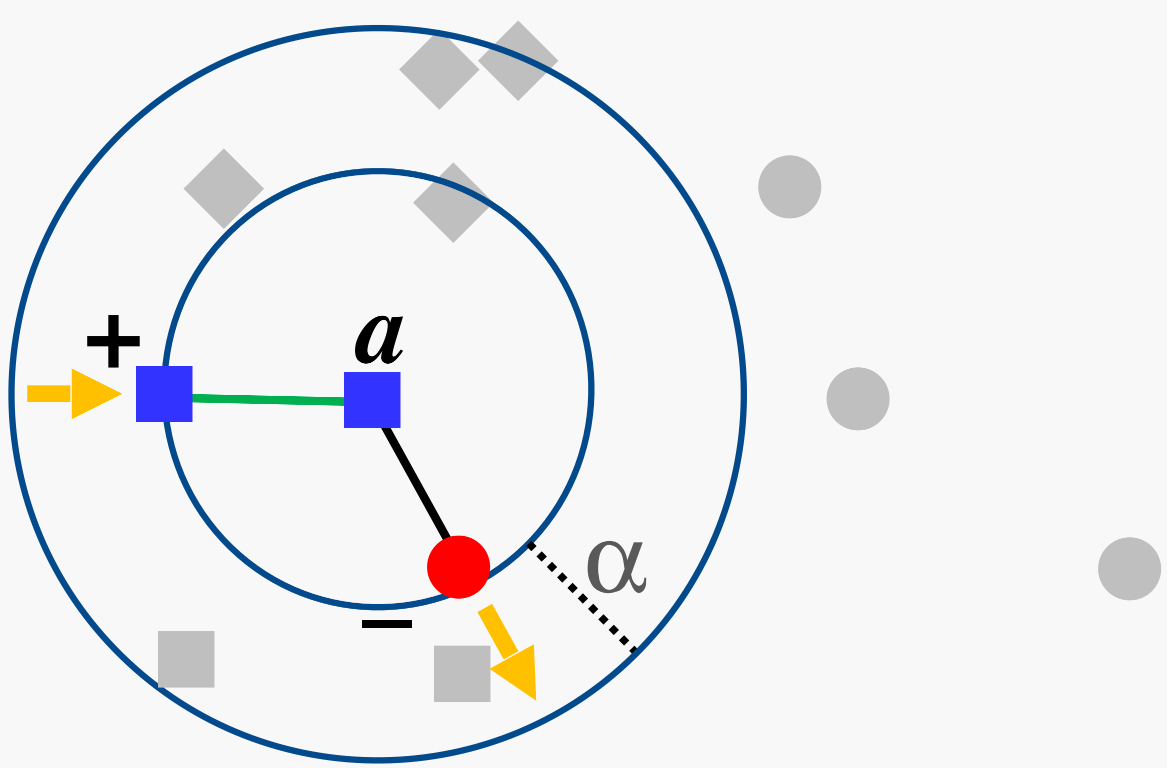
Uses three samples:

an anchor: $\boldsymbol{a}$

a positive sample of the same class: **+**

a negative sample of a different class: **–**

$$J_{a,+,-} = \max\left(0, \underline{D^2_{a,+}} - \underline{D^2_{a,-}} + \alpha\right)$$

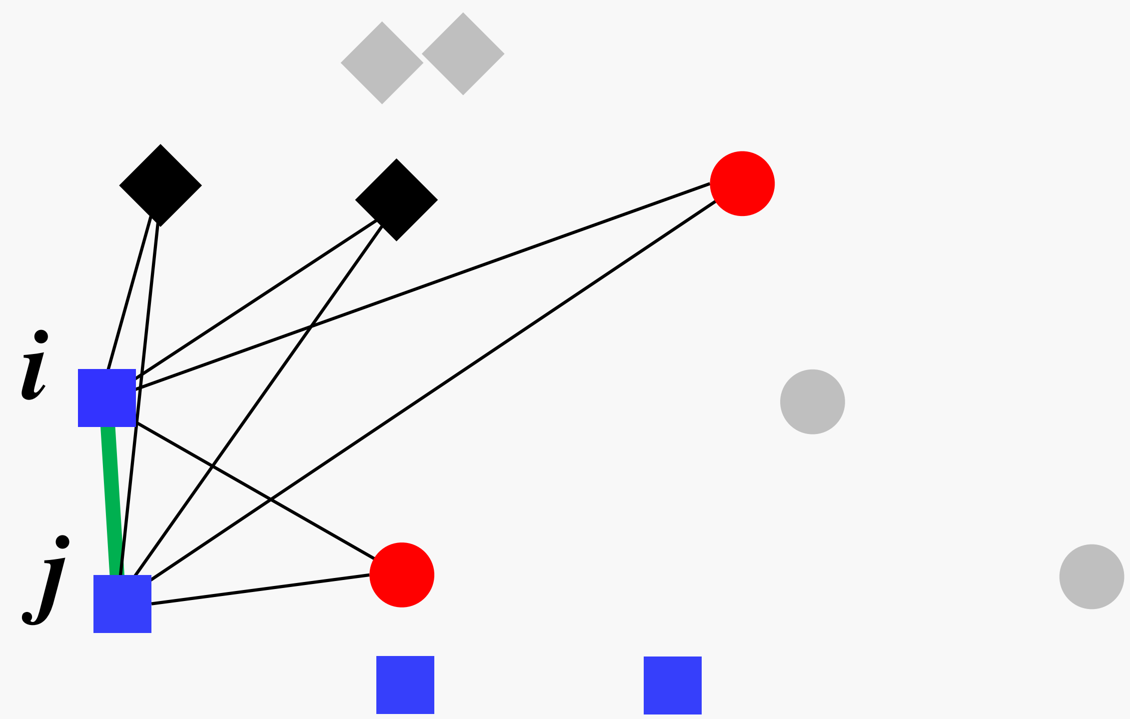


Lifted Structured Loss

Uses more samples from one class,
however focus on one class only

$$\tilde{J}_{i,j} = \log\left(\sum_{(i,k)\in\mathcal{N}} \exp(\alpha - \underline{D_{i,k}}) + \sum_{(j,l)\in\mathcal{N}} \exp(\alpha - \underline{D_{j,l}}) \right) + \underline{D_{i,j}}$$

$$J = \frac{1}{2|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \max(0, \tilde{J}_{i,j})^2$$



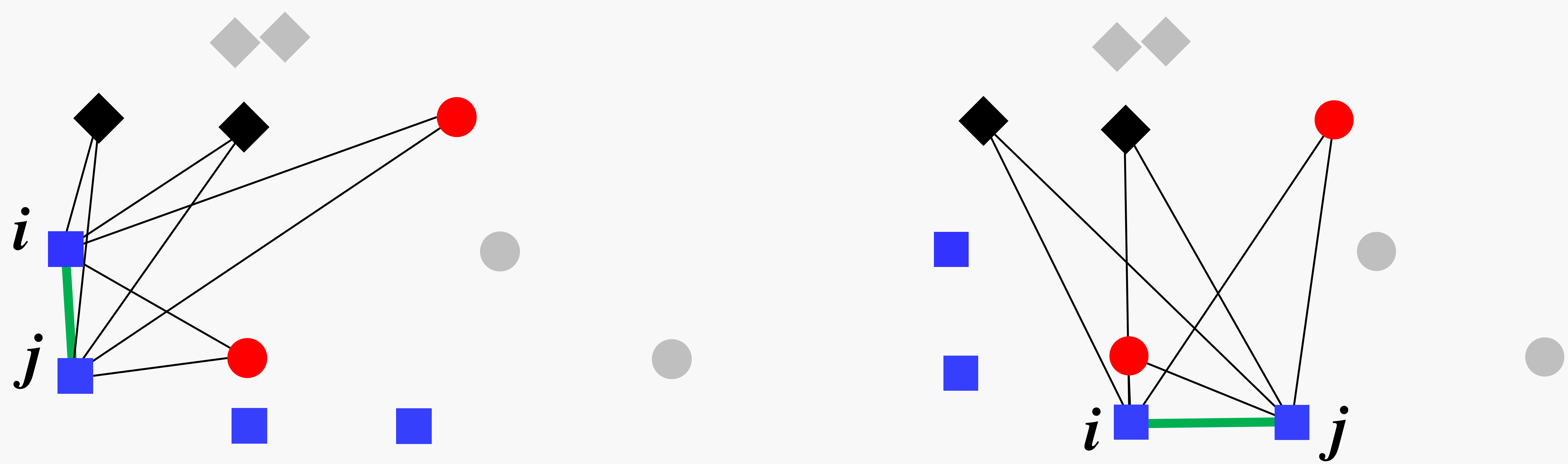
$\mathcal{P}$ **:** all pairs (i,j) of the same class, $\mathcal{N}$ **:** all samples of a different class

Lifted Structured Loss

Uses more samples from one class,
however focus on one class only

$$\tilde{J}_{i,j} = \log\Big(\sum_{(i,k)\in\mathcal{N}} \exp(\alpha - \underline{D_{i,k}}) + \sum_{(j,l)\in\mathcal{N}} \exp(\alpha - \underline{D_{j,l}}) \Big) + \underline{D_{i,j}}$$

$$J = \frac{1}{2|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \max(0, \tilde{J}_{i,j})^2$$



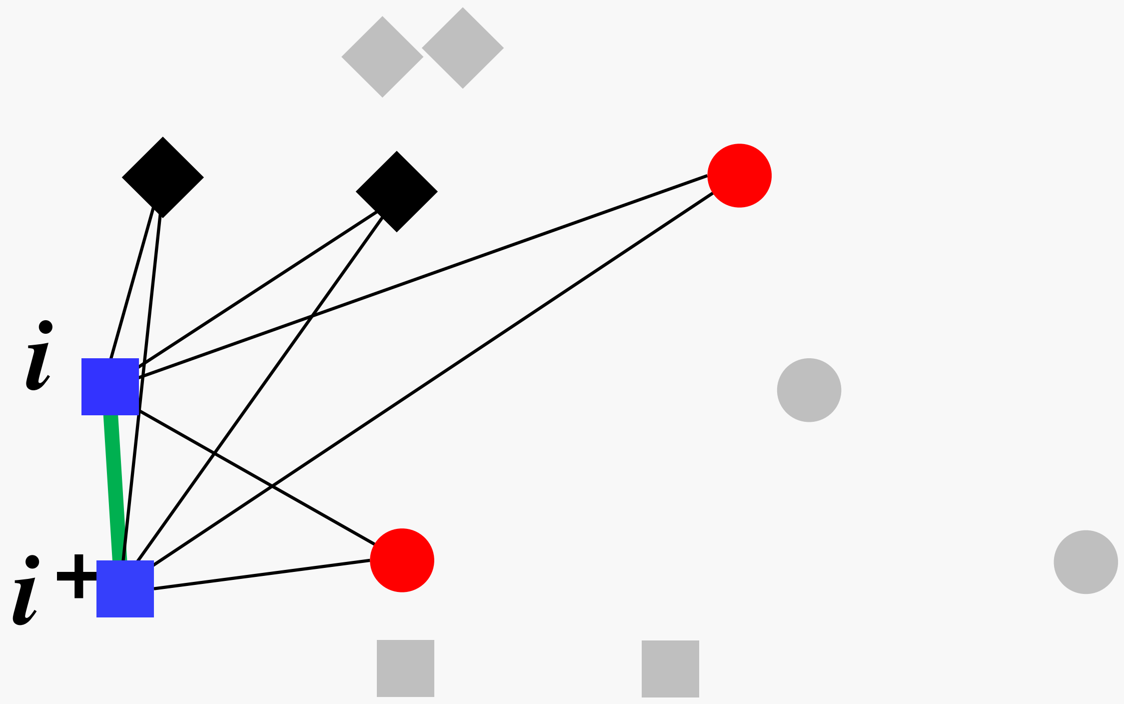
$\mathcal{P}$ **:** all pairs (i,j) of the same class, $\mathcal{N}$ **:** all samples of a different class

N-Pair Loss

Uses multiple pairs of same class samples, however only one pair per class

$$J = \frac{1}{N}\sum_{i=1}^{N}\log\Big(1+\sum_{j\neq i}\exp(\underline{f(x_i)^T f(x_j)} - \underline{f(x_i)^T f(x_i^+)})\Big)$$

Example of N=3 classes → 3 pairs $\boldsymbol{i}$ and $\boldsymbol{i}^+$:



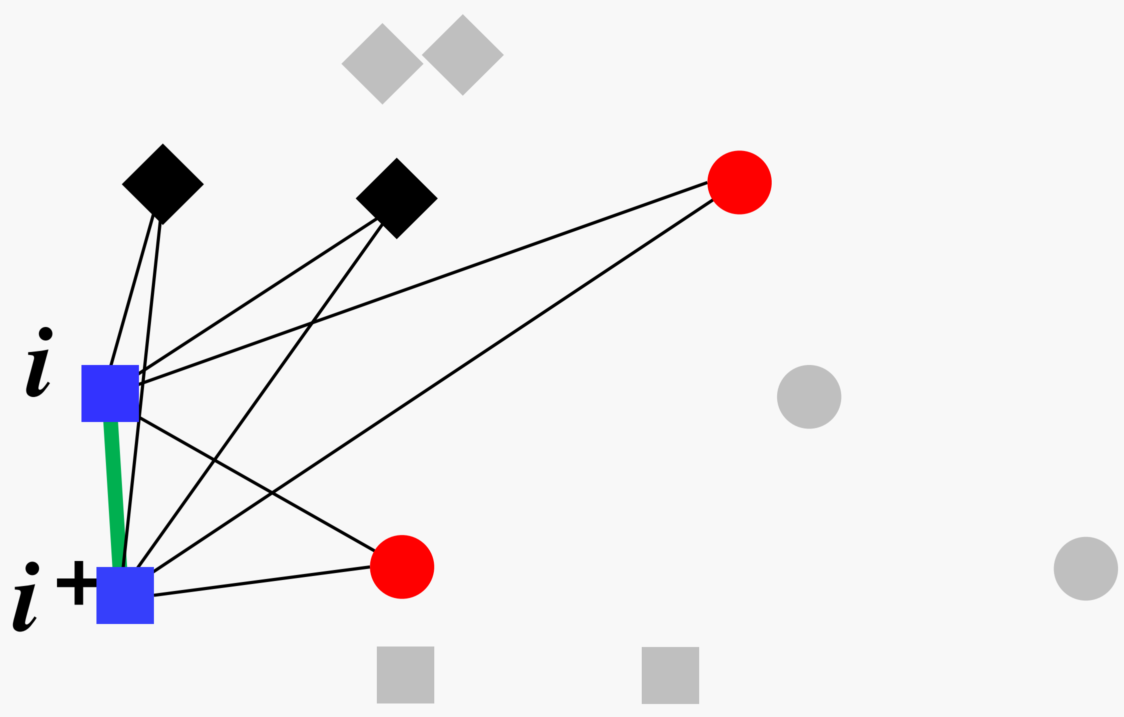
with regard to the blue class

N-Pair Loss

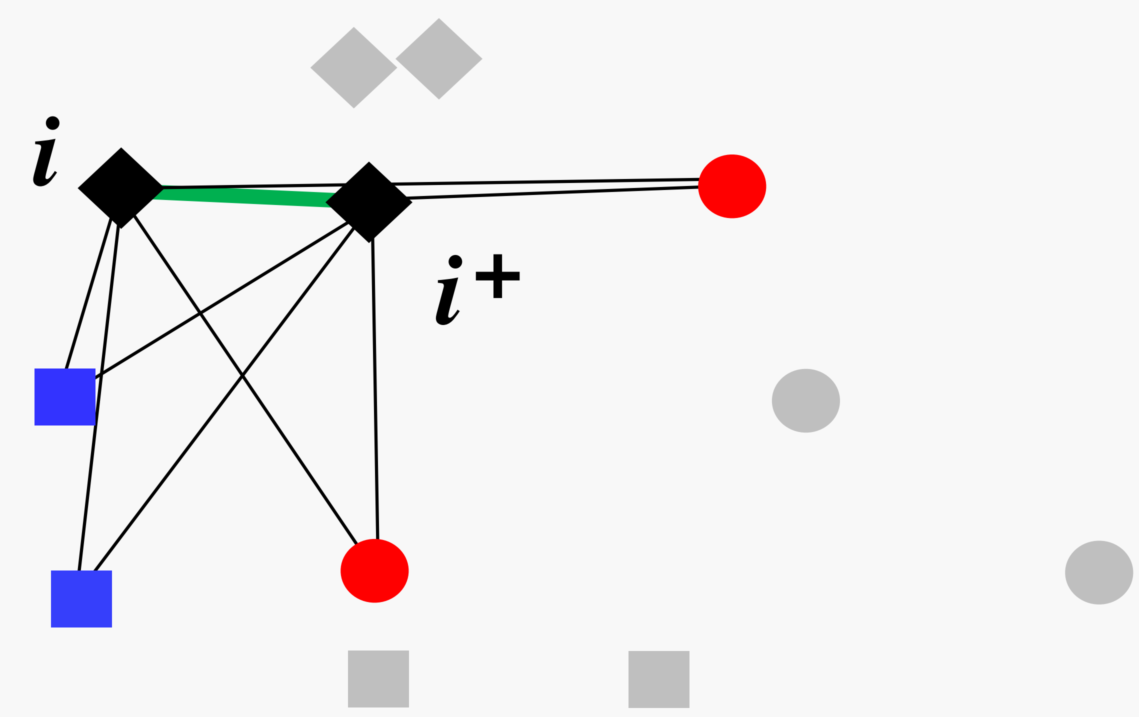
Uses multiple pairs of same class samples, however only one pair per class

$$J = \frac{1}{N} \sum_{i=1}^{N} \log\left(1 + \sum_{j \neq i} \exp(\underline{f(x_i)^T f(x_j)} - \underline{f(x_i)^T f(x_i^+)})\right)$$

Example of N=3 classes → 3 pairs i and i^+:



with regard to the blue class



with regard to the black class

...

Our proposal:

- Use **multiple classes** and
- **Multiple samples per class**
- As retrieval quality does not depend on the actual distances, but rather on the ranking order, use **normalized approximated ranks** instead of distances
- **Focus on** those batch elements that **hurt image retrieval quality most**
- Use a **nonlinear rank transformation function** to **boost the impact of ranking errors**

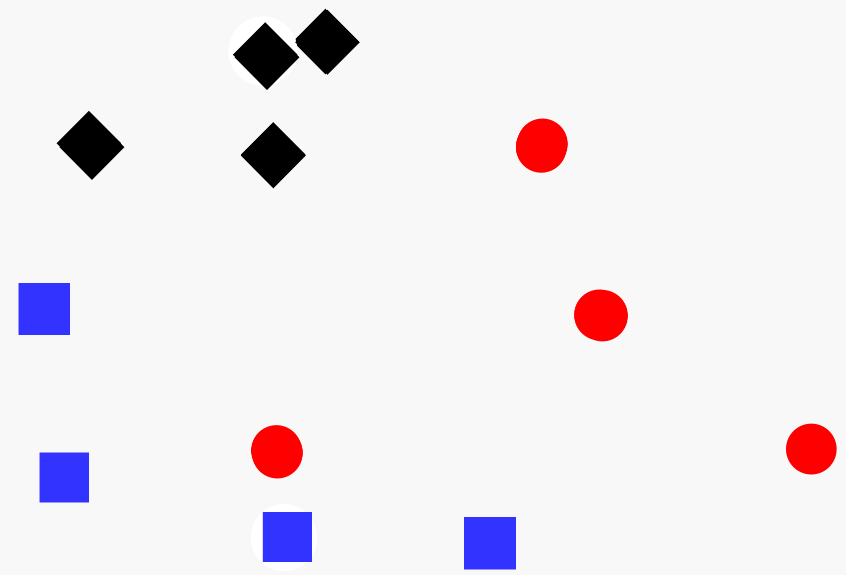
Nonlinear Rank Approximation (NRA) Loss

- Batches of size $m = k \cdot n$

  k classes

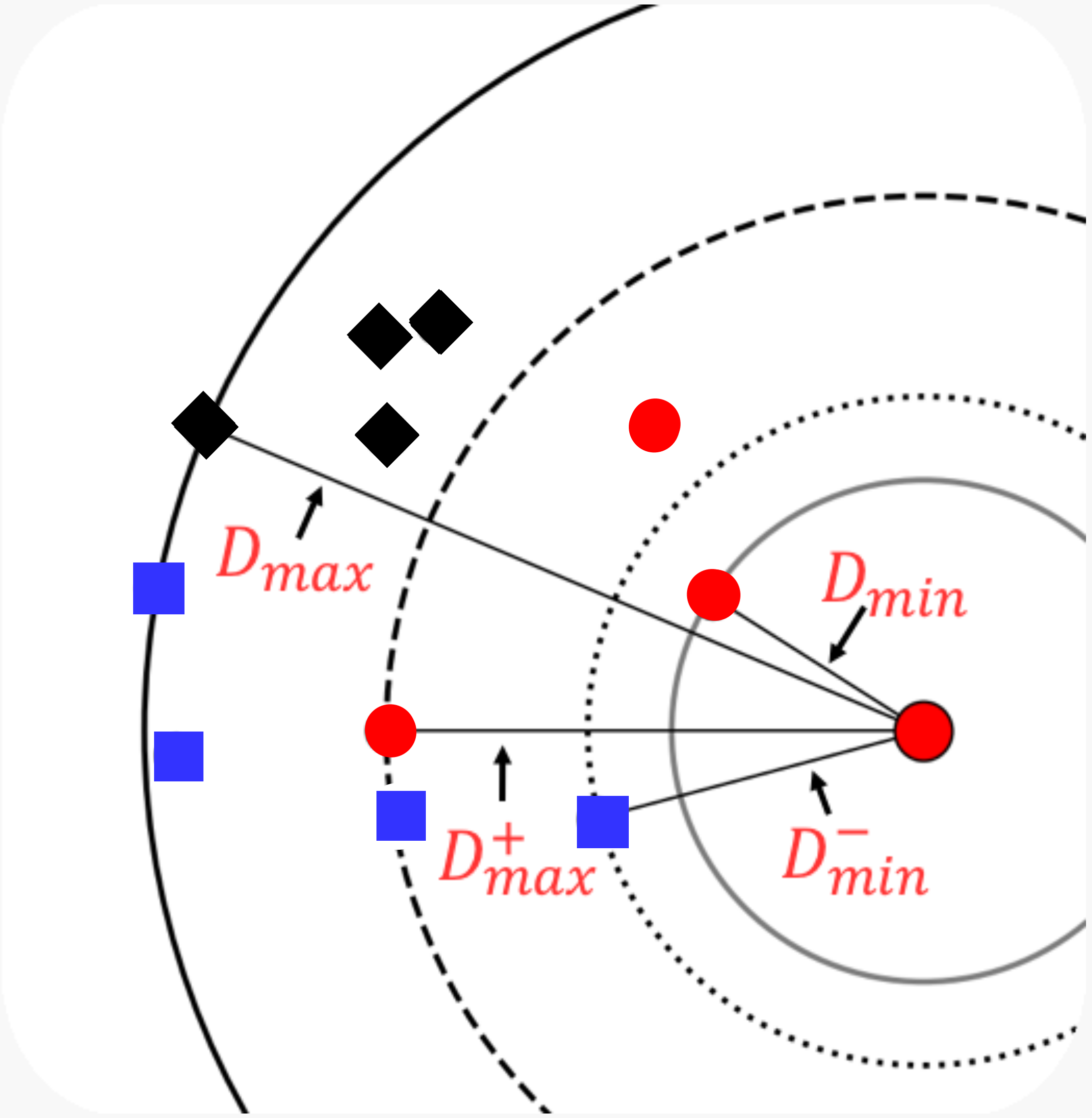
  n samples per class

- All FVs of the batch are treated as anchors



Nonlinear Rank Approximation (NRA) Loss

- Batches of size $m = k \cdot n$
  k classes
  n samples per class
- All FVs of the batch
  are treated as anchors
- Instead of evaluating all
  other samples of the batch,
  for each anchor focus on
  the **most distant sample of the same class** D^+_{max}
  the **closest sample of a different class** D^-_{min}



Nonlinear Rank Approximation (NRA) Loss

- Batches of size $m = k \cdot n$

  k classes

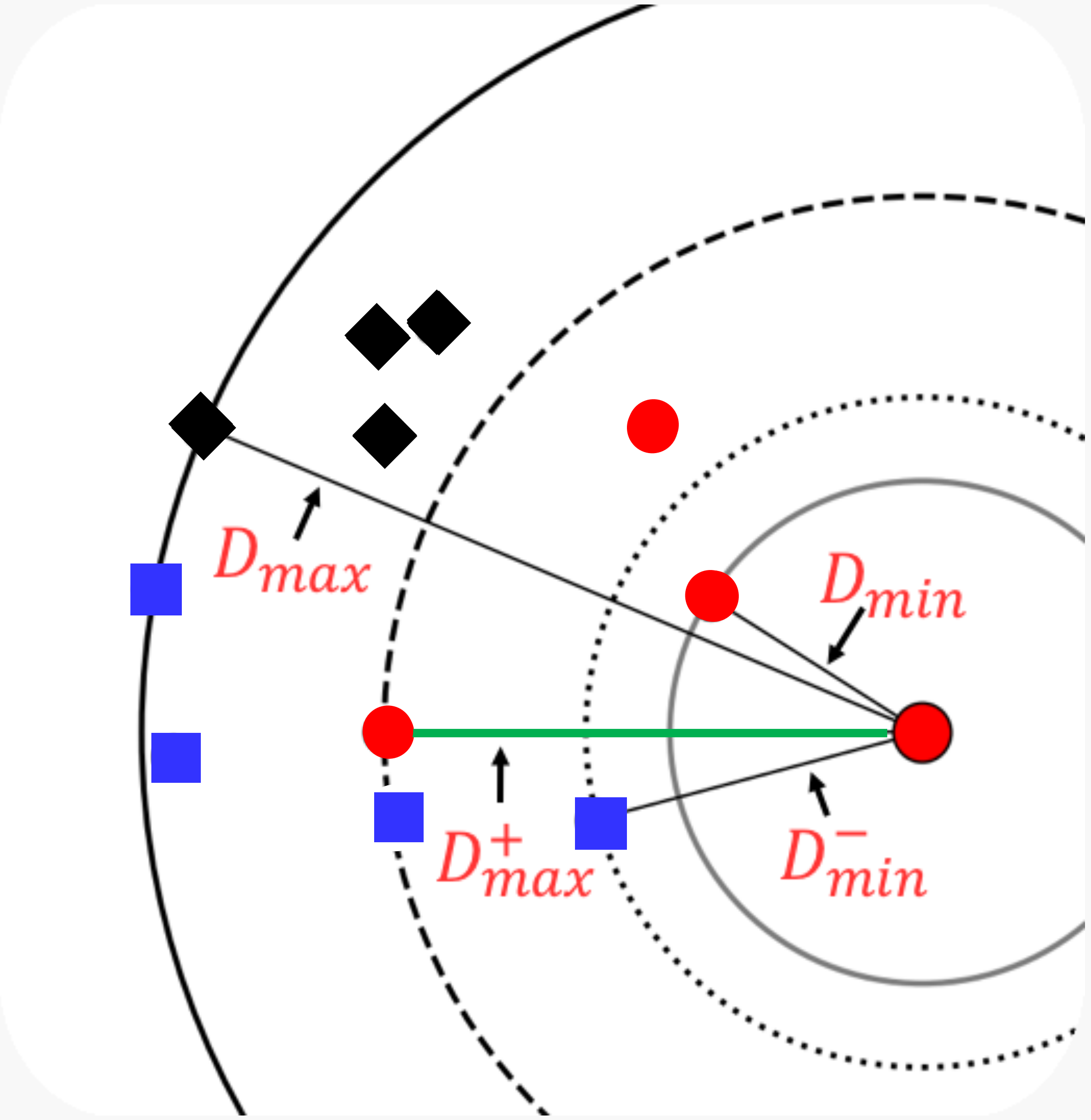
  n samples per class

- All FVs of the batch are treated as anchors

- Instead of evaluating all other samples of the batch, for each anchor focus on

  the **most distant sample of the same class** D^{+}_{max}

  the **closest sample of a different class** D^{-}_{min}



Nonlinear Rank Approximation (NRA) Loss

- Batches of size $m = k \cdot n$

  k classes

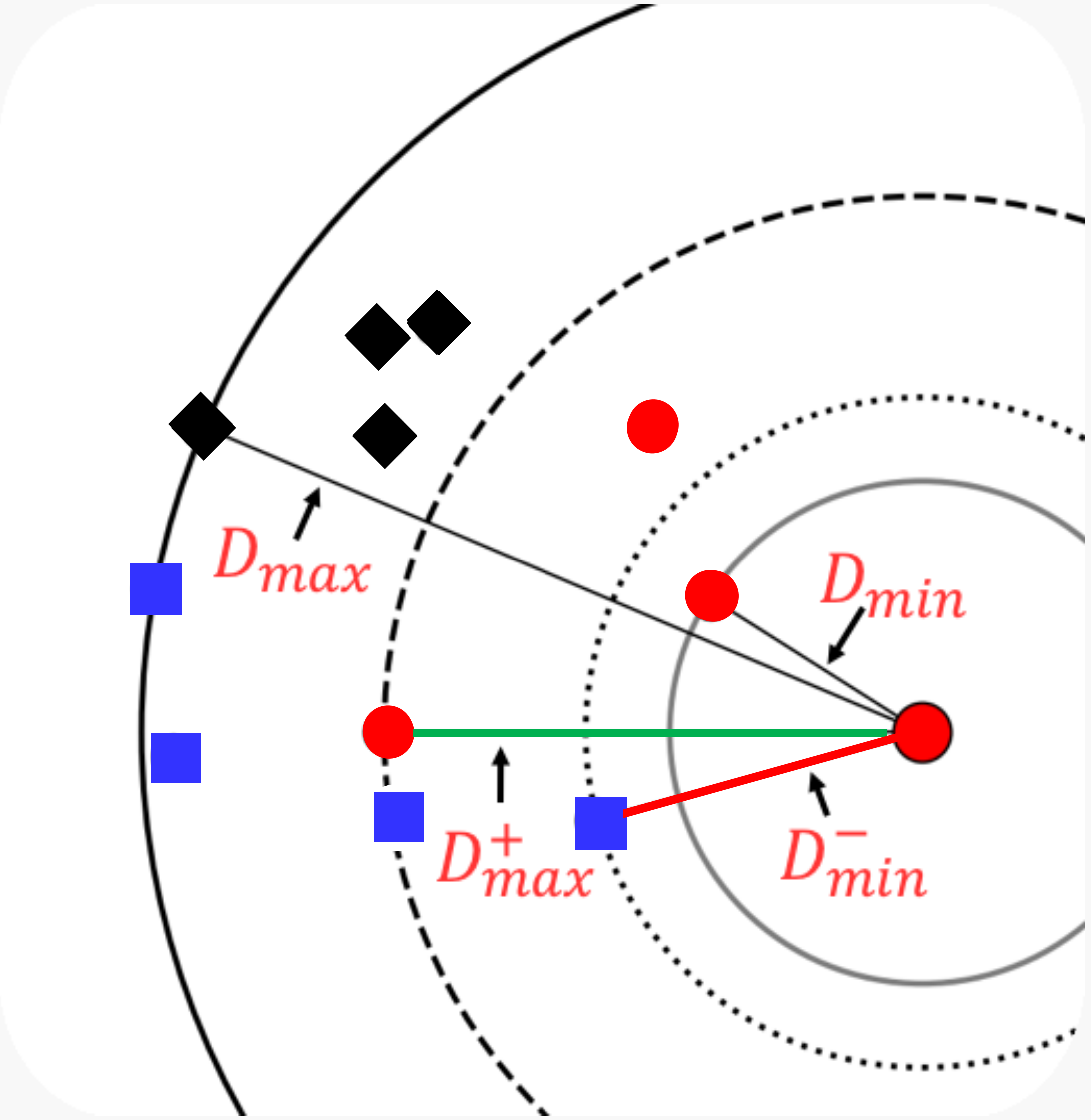
  n samples per class

- All FVs of the batch are treated as anchors

- Instead of evaluating all other samples of the batch, for each anchor focus on

  the **most distant sample of the same class** D^{+}_{max}

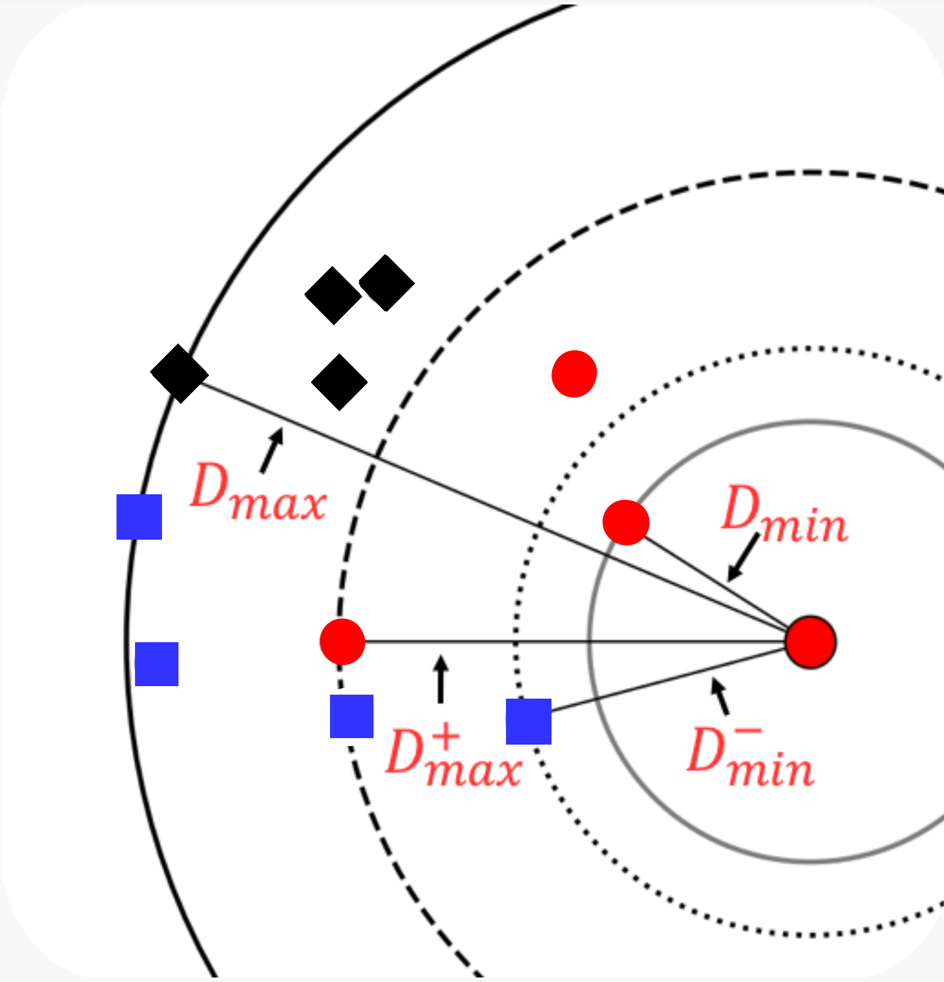
  the **closest sample of a different class** D^{-}_{min}



Nonlinear Rank Approximation

- For each anchor i the distances are converted to approximated **normalized ranks**:

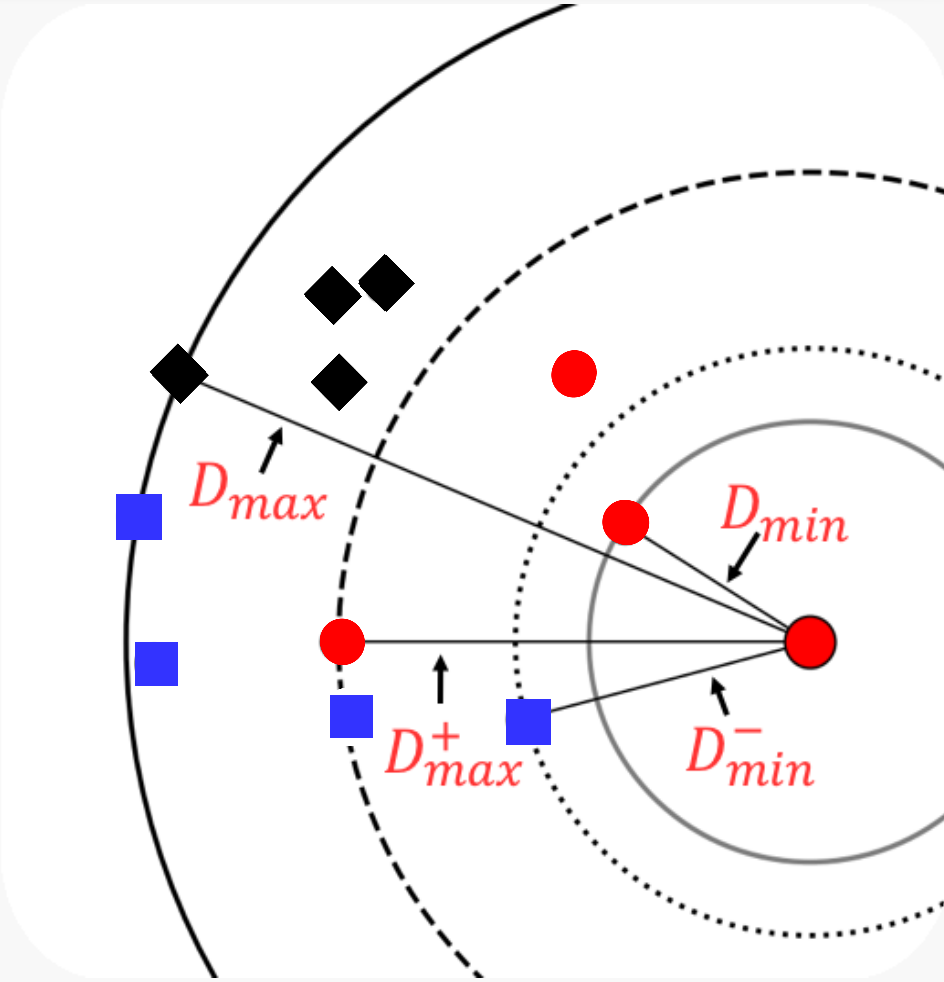
$$r_{i,j} = \frac{D_{i,j} - D_{i,\min}}{D_{i,\max} - D_{i,\min}} \in [0, 1]$$



Nonlinear Rank Approximation

- For each anchor i the distances are converted to approximated **normalized ranks**:

$$r_{i,j} = \frac{D_{i,j} - D_{i,\min}}{D_{i,\max} - D_{i,\min}} \in [0, 1]$$



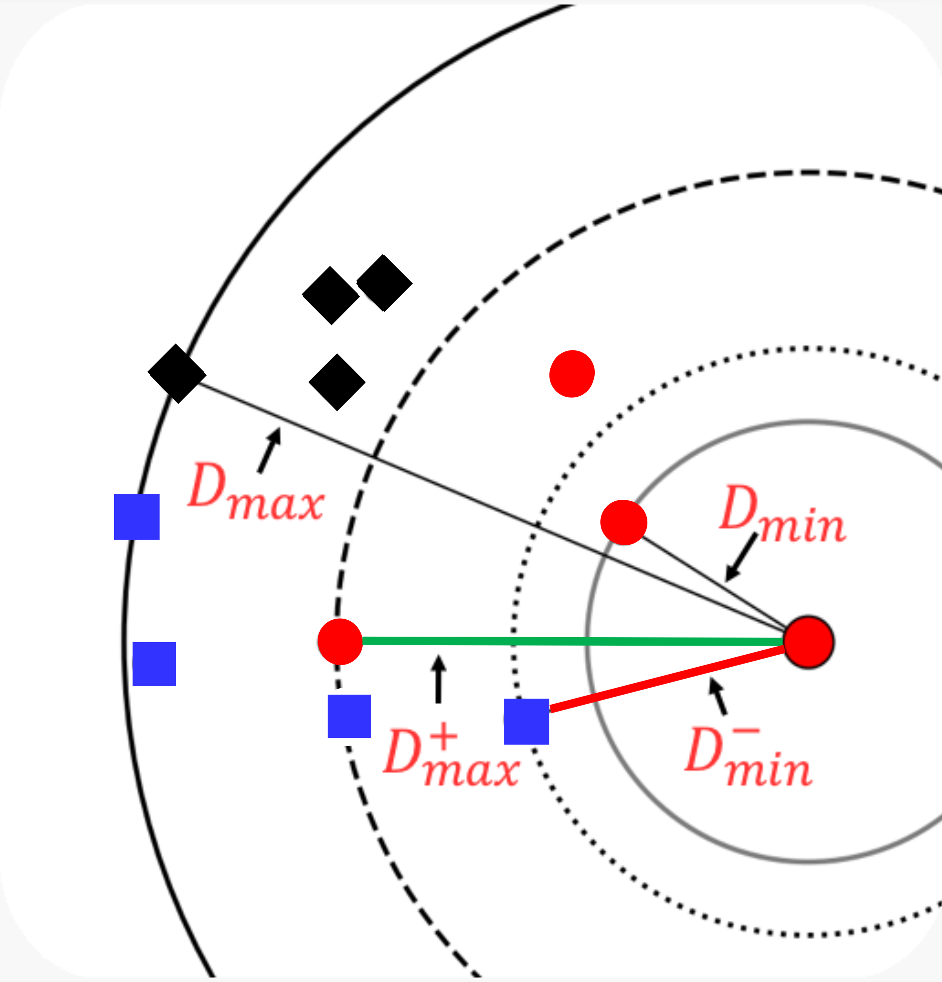
- These ranks are nonlinearly transformed to **similarities**:

$$s_{i,j} = 1 - w(r_{i,j})$$

Nonlinear Rank Approximation

- For each anchor i the distances are converted to approximated **normalized ranks**:

$$r_{i,j} = \frac{D_{i,j} - D_{i,\min}}{D_{i,\max} - D_{i,\min}} \in [0, 1]$$



- These ranks are nonlinearly transformed to **similarities**:

$$s_{i,j} = 1 - w(r_{i,j})$$

Most distant sample of the same class:

$$\underline{D^{+}_{i,max} \rightarrow r^{+}_{i,max} \rightarrow s^{+}_{i,max}}$$

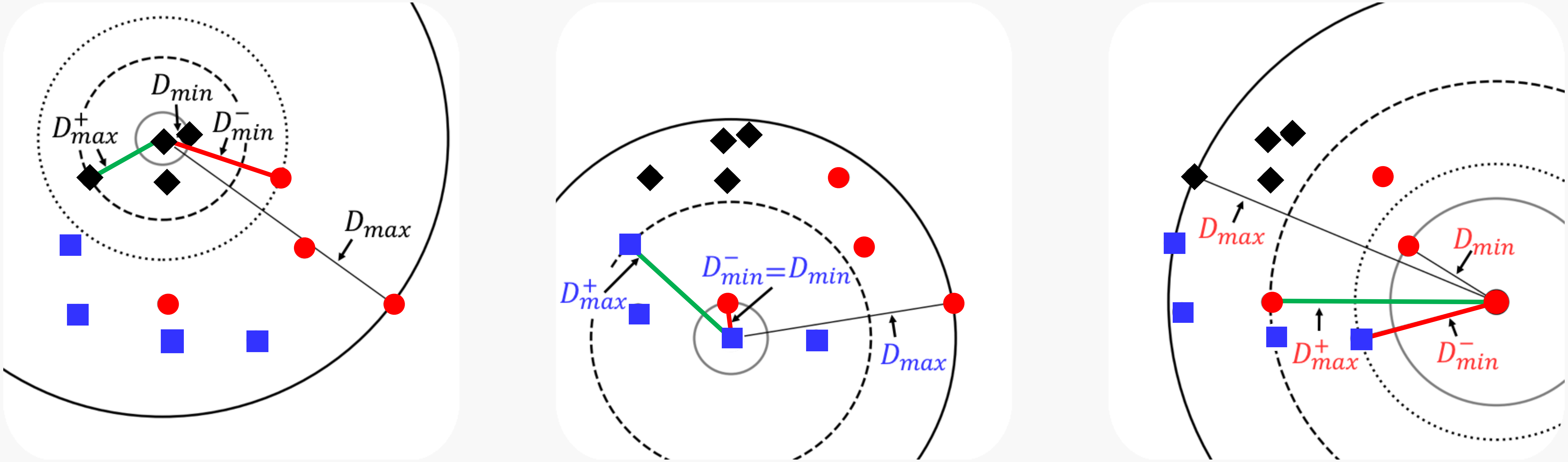
Closest sample of a different class:

$$\underline{D^{-}_{i,min} \rightarrow r^{-}_{i,min} \rightarrow s^{-}_{i,min}}$$

NRA Loss Function

$$J = -\frac{1}{m}\sum_{i=1}^{m}\left(\log(\underline{s^{+}_{i,\max}}) + \log(1 - \underline{s^{-}_{i,\min}})\right)$$

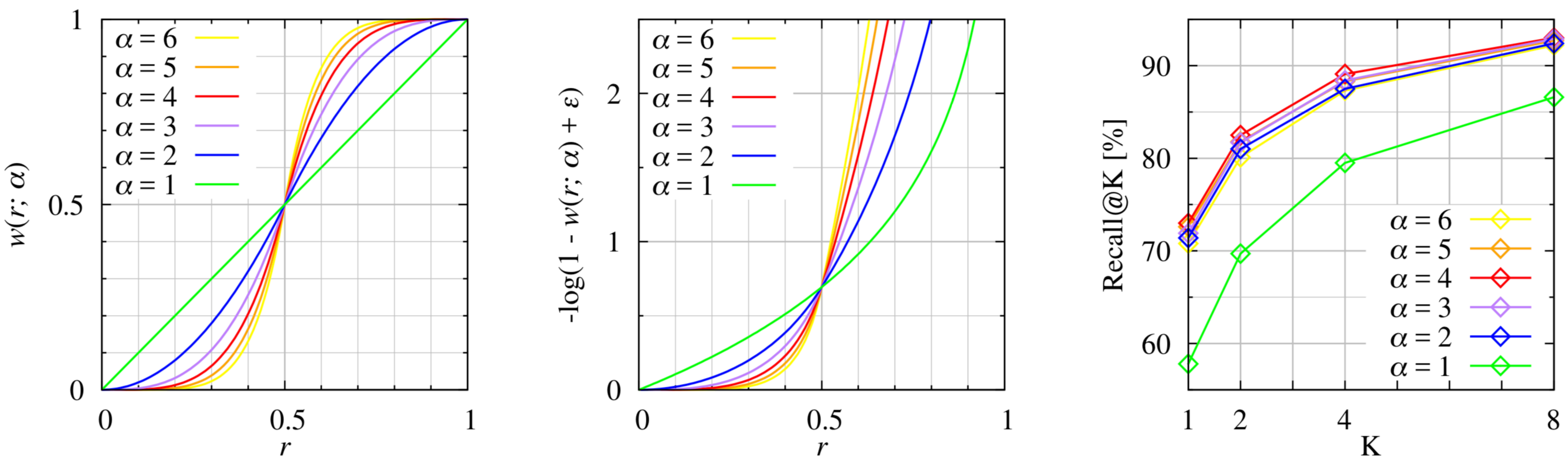
Three of the 12 specific configurations of the previous example:



Evaluation of Nonlinear Transfer Function w

- The approximated ranks $r_{i,j}$ are nonlinearly transformed to similarities by $s_{i,j} = 1 - w(r_{i,j})$

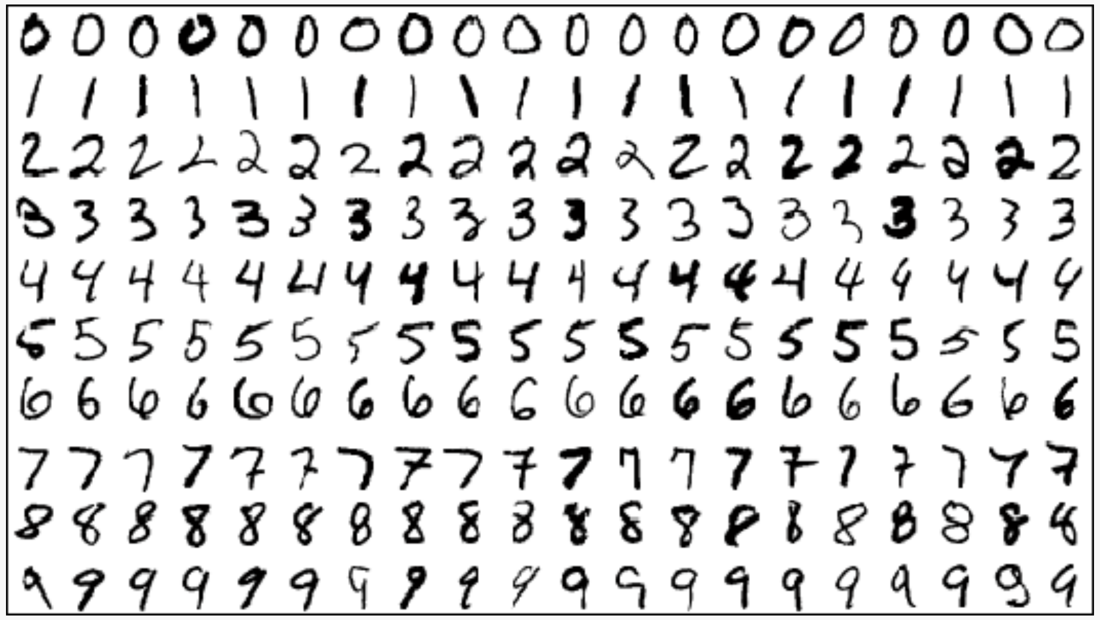
$$w(r;\alpha) = \begin{cases} \frac{1}{2}(2r)^{\alpha} & r \in [0, \frac{1}{2}) \\ 1 - \frac{1}{2}(2(1-r))^{\alpha} & r \in [\frac{1}{2}, 1] \end{cases}$$

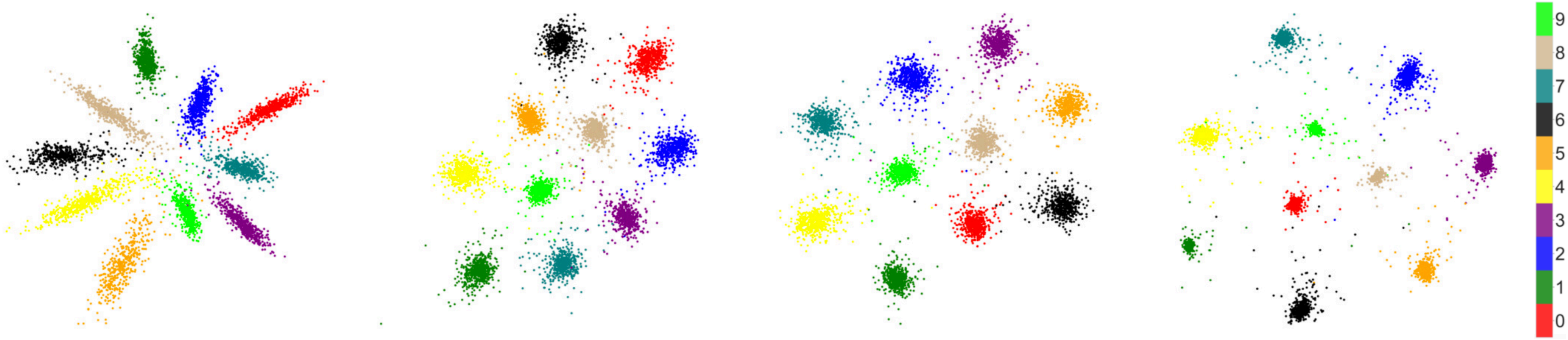


Transfer function (left), the corresponding loss component (center), and Recall@K results on the Cars196 data set (right)

Visualization of the 2D Embedding Space

... for different loss functions using the MNIST data set





(a) Softmax Cross Entropy
mAP = 93.3

(b) Triplet Loss
mAP = 97.8

(c) Lifted Structured Loss
mAP = 98.4

(d) Nonlinear Rank Approx. Loss (ours)
mAP = 98.8

Fine Tuning for Unseen Object Retrieval

- Protocol introduced by Song et al. 2016 is strictly followed for comparability
- 3 Datasets CUB200-2011, Cars196, Stanford Online Products (SOP) are split such that the images from the first half of categories are used for training and images from the second half are used for testing
- GoogleNet is used as the CNN
- Feature vectors in 64 and 512 dimensions

Fine Tuning for Unseen Object Retrieval

Recall@1 values for 64/512 dimensional FVs

(Reproduced results)

Method	CUB	Cars196	SOP
Triplet	46.3 / 51.6	56.5 / 58.4	57.2 / 59.8
Lifted	45.7 / 55.7	48.8 / 50.7	61.6 / 63.8
N-Pair	51.8 / 56.4	63.3 / 68.3	63.6 / 65.4
NRA (ours)	**57.6 / 64.3**	**73.0 / 81.9**	**71.9 / 75.6**

Comparison to the State of the Art

Recall@1 values from different methods

Method	Network	Dim.	CUB	Cars196	SOP
Margin	ResNet50 v2	128	63.6	79.6	72.7
NRA (ours)	ResNet50 v2	128	**64.5**	**79.9**	**75.3**
Angular	GoogLeNet	512	54.7	71.4	70.9
A-BIER	GoogLeNet	512	57.5	82.0	74.2
ABE-8	GoogLeNet (x8)	512	60.6	**85.2**	**76.3**
NRA (ours)	GoogLeNet	512	**64.3**	82.1	75.6

Thank you very much!

More Information at

www.visual-computing.com