Loss Functions for Deep Metric Learning Using Binary Supervision and Beyond

Suha Kwak

suha.kwak@postech.ac.kr

Graduate School of Artificial Intelligence

Dept. of Computer Science and Engineering

POSTECH

How much similar/dissimilar semantically?

D  ,  $<$ D  , 

Metric: Function that quantifies a distance

Metric Learning: Learning a metric from a set of data

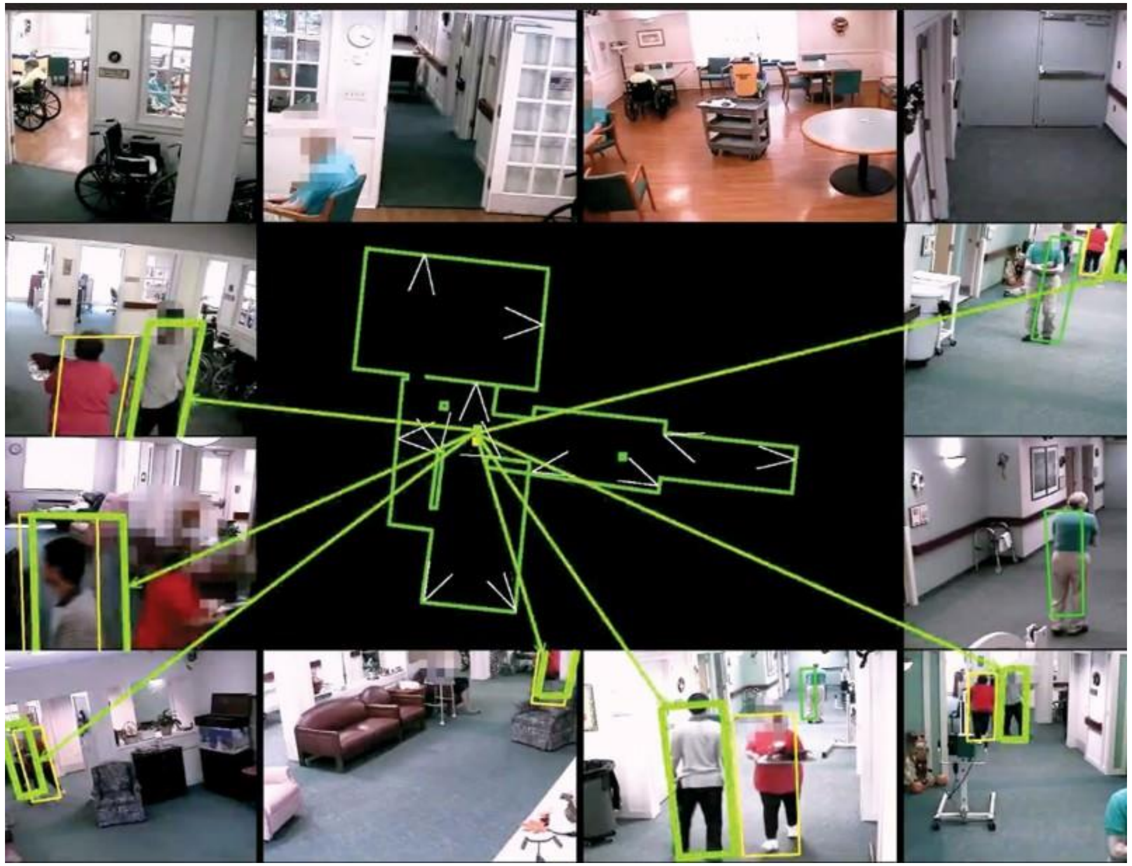
Applications

Content-based image retrieval

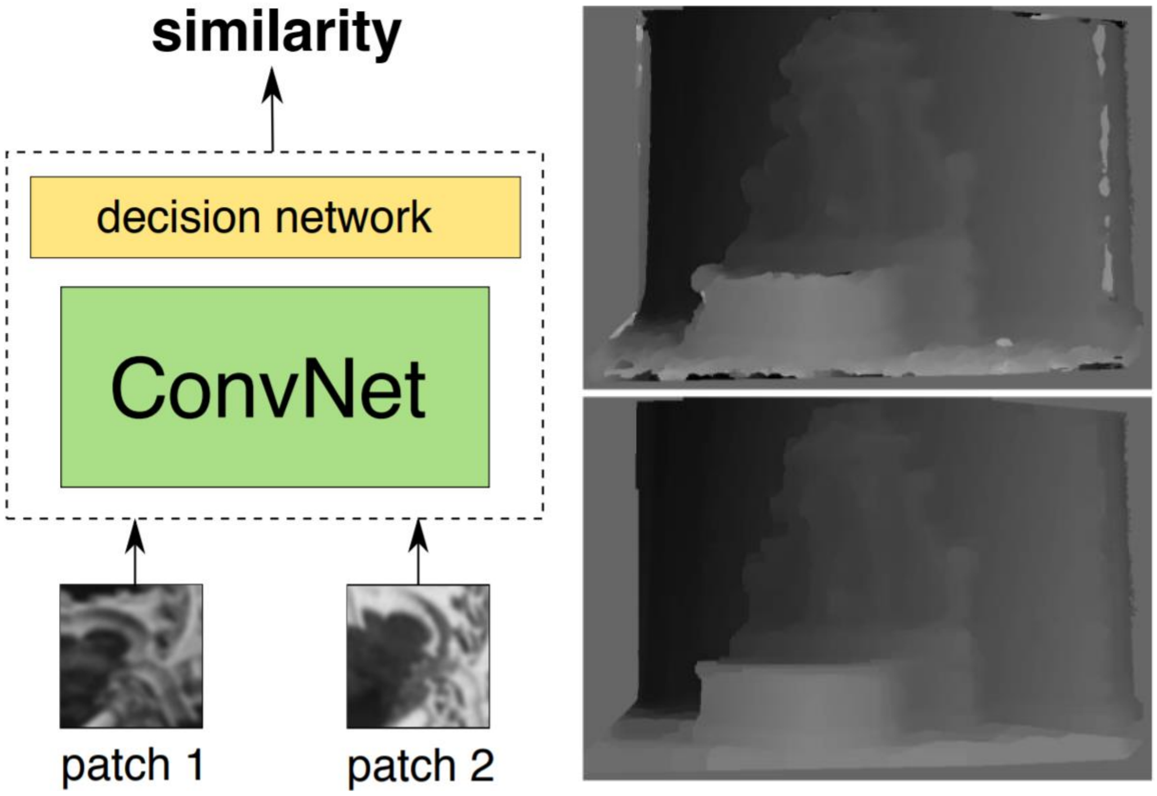
Face verification/identification[1]

[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015

Applications



Person re-identification[2]

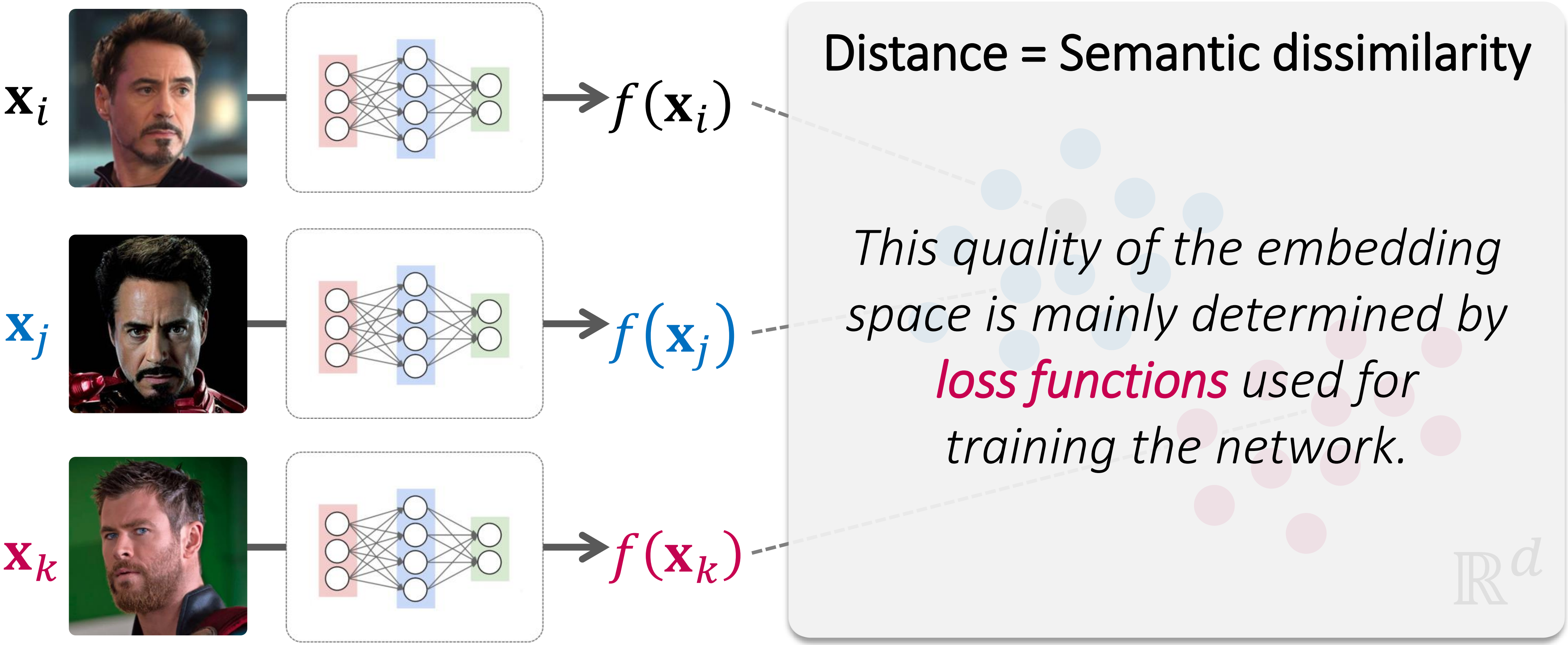


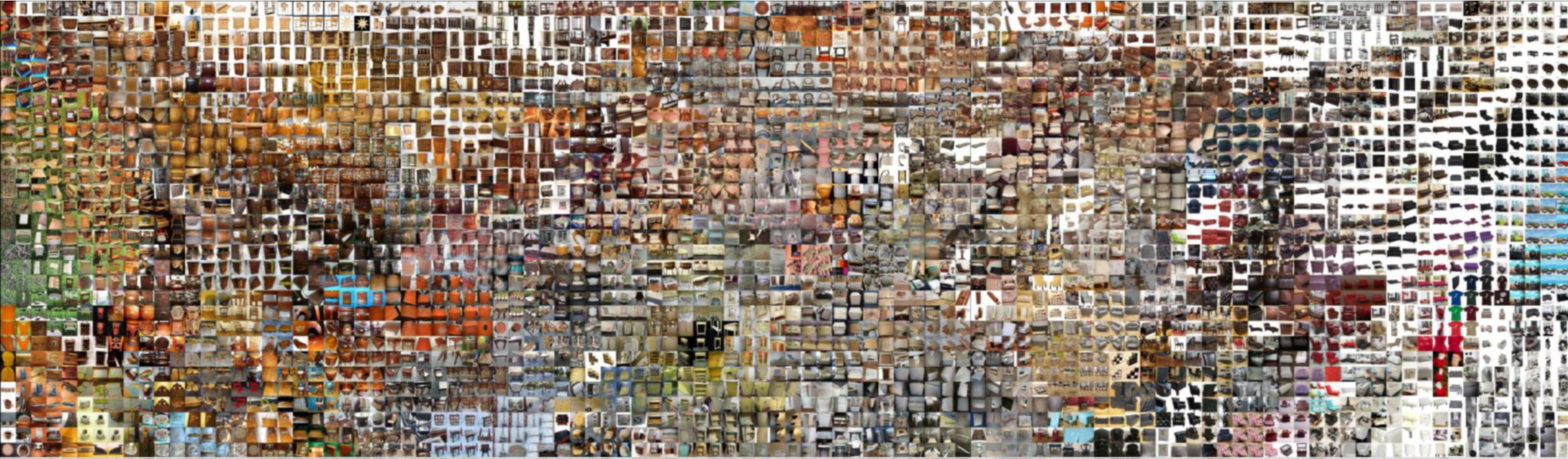
Patch matching/stereo imaging[3]

[2] Beyond triplet loss: a deep quadruplet network for person re-identification, CVPR 2017
[3] Learning to compare image patches via convolutional neural networks, CVPR 2015

Deep Metric Learning

Learning a deep embedding network f so that semantically similar images are closely grouped together





Proxy Anchor Loss for Deep Metric Learning

Sungyeon Kim Dongwon Kim Minsu Cho Suha Kwak

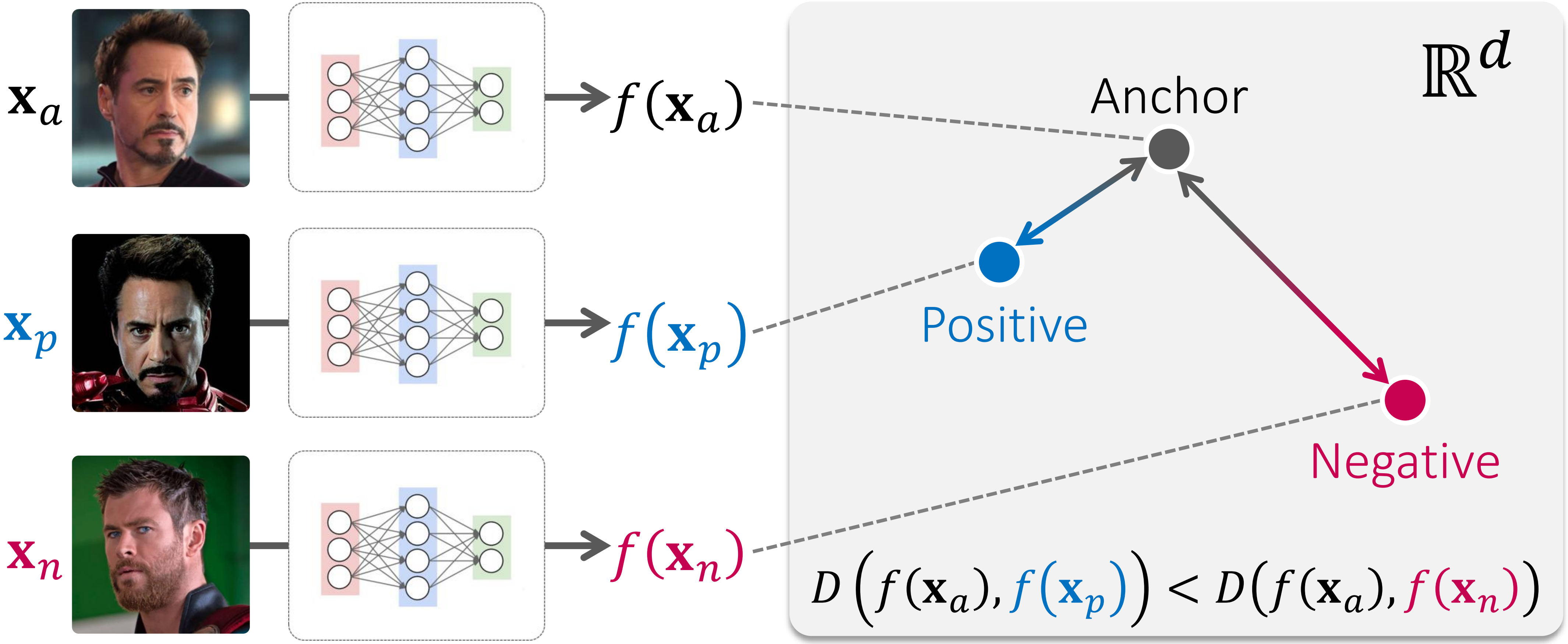
{tjddus9597, kdwon, mscho, suha.kwak}@postech.ac.kr

POSTECH

Well-known Examples of Metric Learning Losses

- Triplet rank loss[1]

$$\ell_{\text{tri}}(a,p,n) = \left[D\left(f_a, f_p\right) - D(f_a, f_n) + \delta\right]_+$$

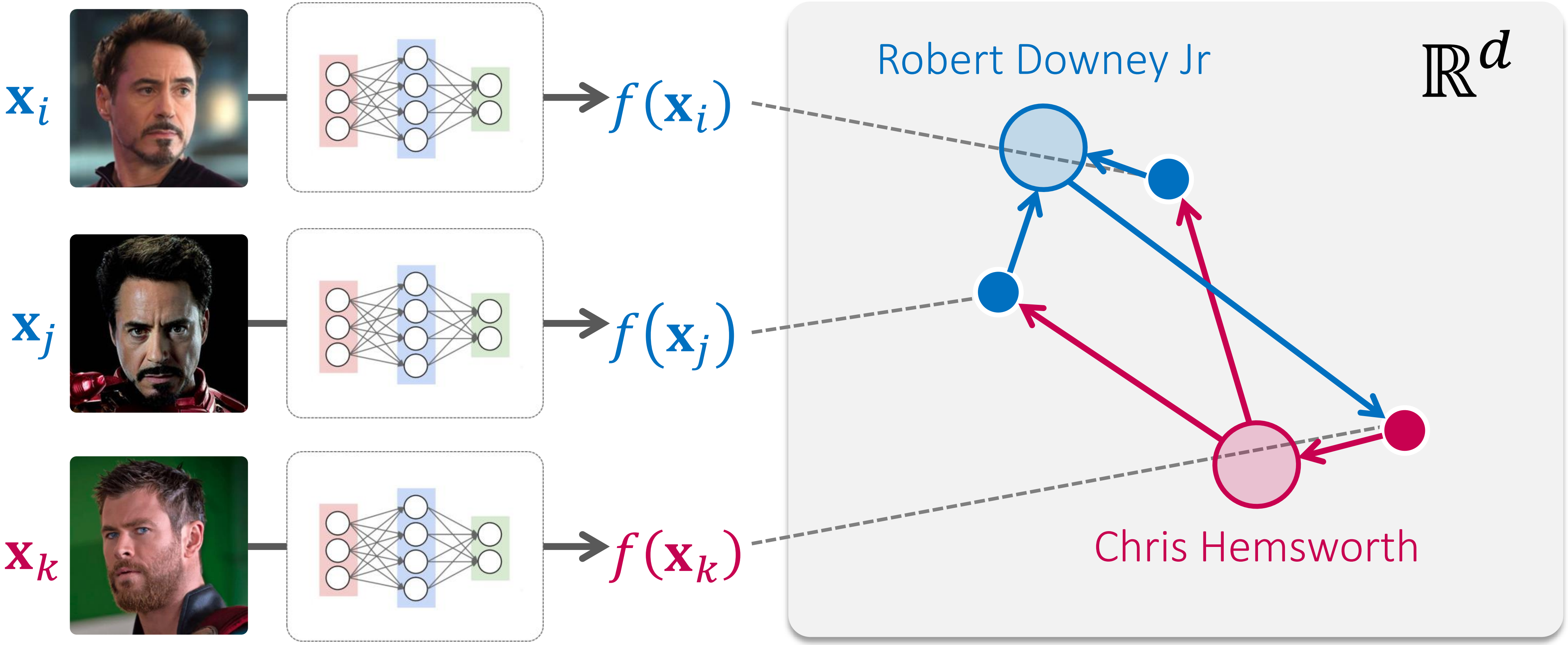


[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015

Well-known Examples of Metric Learning Losses

- Proxy NCA loss[6]

$$\ell_{\text{proxyNCA}}(B) = \sum_{i \in B}\left\{ D(f_i, p^+) - \log \sum_{p^- \in P^-} \exp\left(-D(f_i, p^-)\right)\right\}$$



[6] No fuss distance metric learning using proxies, ICCV 2017

Two Categories of Existing Metric Learning Losses

- Pair-based losses
  - (+) Exploiting *data-to-data relations*, fine-grained relations between data
  - (–) Prohibitively high training complexity

  - Examples
    - Contrastive loss[4]

$$\ell_{\text{ctr}}(i,j) = y_{ij} D\left(f_i, f_j\right)^2 + \left(1 - y_{ij}\right)\left[\delta - D(f_i, f_j)\right]_+^2$$

    - Triplet rank loss[1]

$$\ell_{\text{tri}}(a,p,n) = \left[D\left(f_a, f_p\right) - D(f_a, f_n) + \delta\right]_+$$

    - N-pair loss[5]

$$\ell_{\text{NP}}(a,p,n_1,\ldots,n_{N-1}) = \log\left(1 + \sum_{i=1}^{N-1} \exp\left(D\left(f_a, f_p\right) - D\left(f_a, f_{n_i}\right)\right)\right)$$

[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015
[4] Learning a similarity metric discriminatively with application to face verification, CVPR 2005
[5] Improved deep metric learning with multi-class N-pair loss objective, NeurIPS 2016

Two Categories of Existing Metric Learning Losses

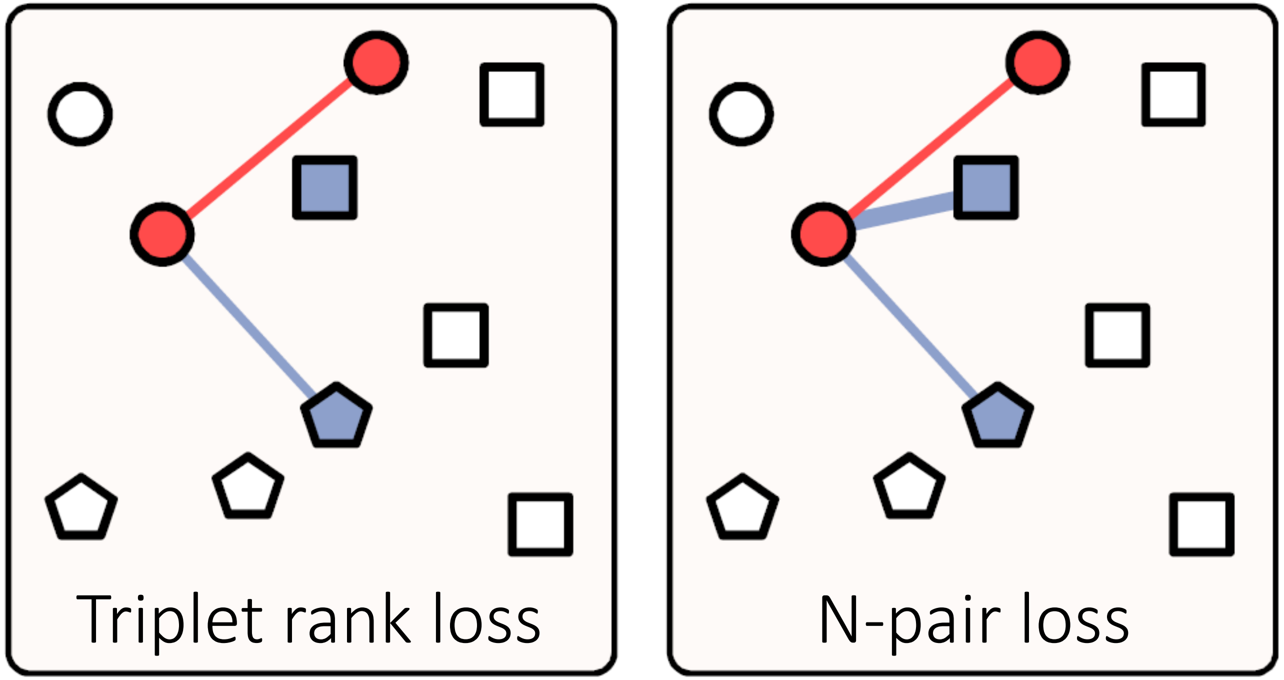
- Proxy-based losses
  - Proxy
    - Representative of a subset of training data
    - Learned as a part of the network parameters
  - Taking each data point as an anchor and associating it with proxies
  - (+) Lower training complexity, faster convergence in general
  - (+) More robust against label noises and outliers
  - (–) Leveraging impoverished data-to-proxy relations only
  - Example: Proxy-NCA loss[6]

$$\ell_{\text{proxyNCA}}(B) = -\sum_{i \in B} \log \frac{\exp\left(-D(f_i, p^+)\right)}{\sum_{p^- \in P^-} \exp\left(-D(f_i, p^-)\right)}$$

[6] No fuss distance metric learning using proxies, ICCV 2017

Two Categories of Existing Metric Learning Losses

Pair-based losses

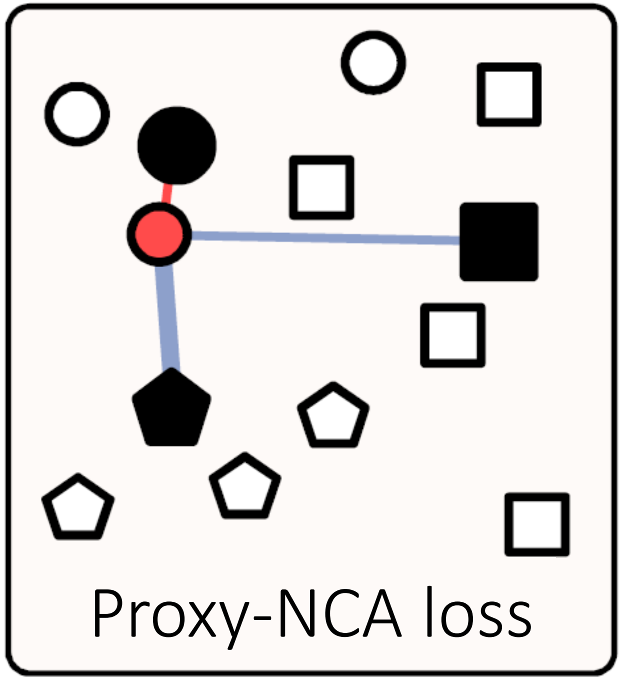


"Data-to-data relations"

Rich and fine-grained

Demanding high training complexity

Proxy-based losses



"Data-to-proxy relations"

Reducing training complexity

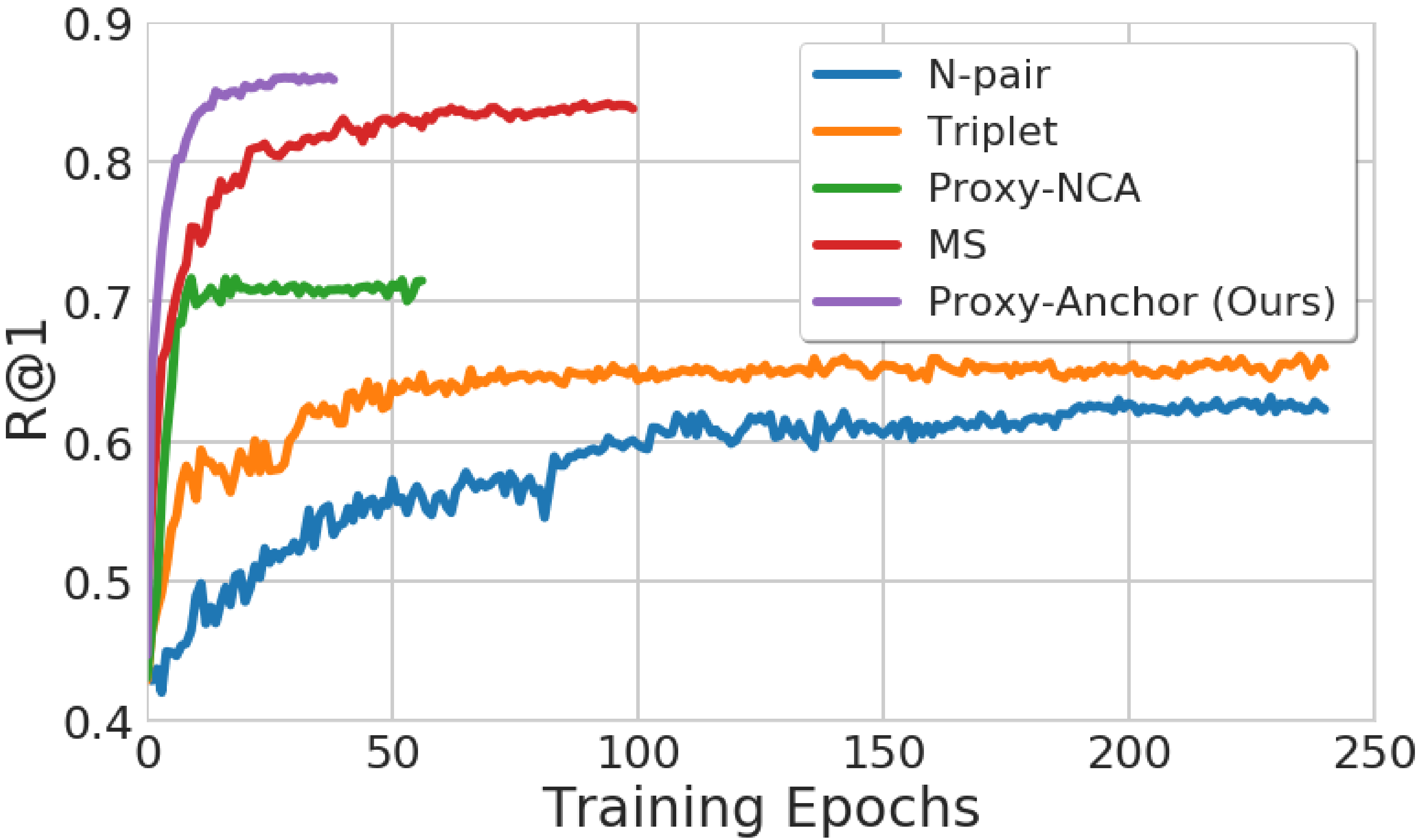
Impoverished information

Our Method

- A new proxy-based loss called *proxy anchor loss*
  - Taking only advantages of both categories
  - Overcoming their limitations
- How it works
  - Using a proxy as an anchor, and associating it with all data in a batch
  - Fast convergence thanks to the use of proxies
  - Taking data-to-data relations into account by allowing data points to interact with each other during training
- Results
  - State-of-the-art performance
  - Fastest convergence (on the Cars-196 dataset)

Our Method

Recall@1 vs. training epochs on the Cars-196 dataset



Details of Proxy Anchor Loss

- Mathematical form and its interpretation

$$\ell(B) = \frac{1}{|P^+|}\sum_{p\in P^+}\log\left(1+\sum_{i\in B_p^+}\exp[-\alpha(S(f_i,p)-\delta)]\right)$$
$$+\frac{1}{|P|}\sum_{p\in P}\log\left(1+\sum_{j\in B_p^-}\exp[\alpha(S(f_j,p)+\delta)]\right)$$

$$= \frac{1}{|P^+|}\sum_{p\in P^+}\left[\mathrm{SoftPlus}\left(\underset{i\in B_p^+}{\mathrm{LSE}}\,-\alpha(S(f_i,p)-\delta)\right)\right]$$
$$+\frac{1}{|P|}\sum_{p\in P}\left[\mathrm{SoftPlus}\left(\underset{j\in B_p^-}{\mathrm{LSE}}\,\alpha(S(f_j,p)+\delta)\right)\right]$$

$S(\cdot,\cdot)$
Cosine similarity

SoftPlus
A smooth approx. of ReLU

LSE
A smooth approx. of MAX

Details of Proxy Anchor Loss

- Mathematical form and its interpretation

$$\ell(B) = \frac{1}{|P^+|} \sum_{p \in P^+} \left[\mathrm{SoftPlus}\left(\operatorname*{LSE}_{i \in B_p^+} -\alpha(S(f_i, p) - \delta) \right) \right] + \frac{1}{|P|} \sum_{p \in P} \left[\mathrm{SoftPlus}\left(\operatorname*{LSE}_{i \in B_p^-} \alpha\left(S\left(f_j, p\right) + \delta\right) \right) \right]$$

Regarding LSE as MAX: pull p and its hardest positive example together,
push p and its hardest negative example apart.

In practice pull/push all embedding vectors in the batch, but with different degrees of strength determined by their relative hardness.

Details of Proxy Anchor Loss

- Analysis on its gradients

$$\frac{\partial \ell(B)}{\partial S(f_i, p)} = \begin{cases} \frac{1}{|P^+|} \frac{-\alpha\, h_p^+(f_i)}{1 + \sum_{j \in B_p^+} h_p^+(f_j)}, & \forall i \in B_p^+, \\ \frac{1}{|P|} \frac{\alpha\, h_p^-(f_i)}{1 + \sum_{k \in B_p^-} h_p^-(f_k)}, & \forall i \in B_p^-, \end{cases}$$

where

$$h_p^+(f) = \exp[-\alpha(S(f, p) - \delta)]$$: Positive hardness metric

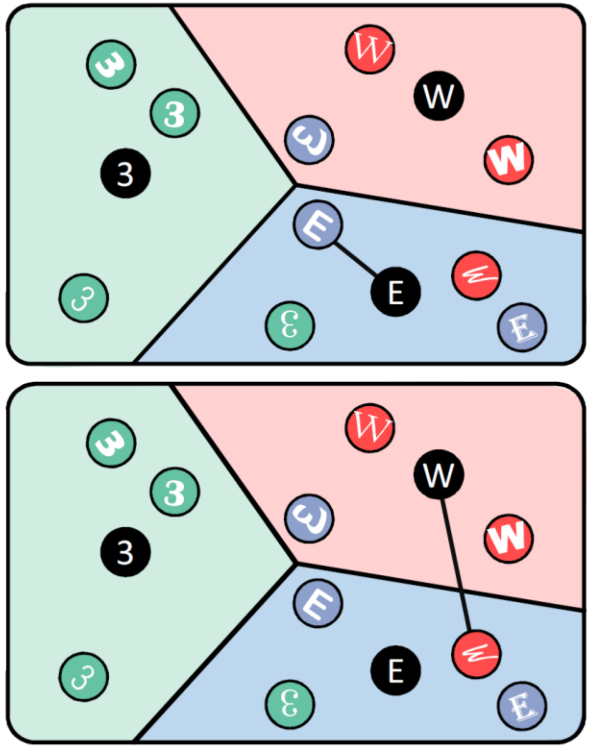
$$h_p^-(f) = \exp[\alpha(S(f, p) + \delta)]$$: Negative hardness metric

The gradient w.r.t. f_i is affected by other examples in the batch.
(The gradient becomes larger when f_i is harder than others.)

Comparison to Proxy NCA

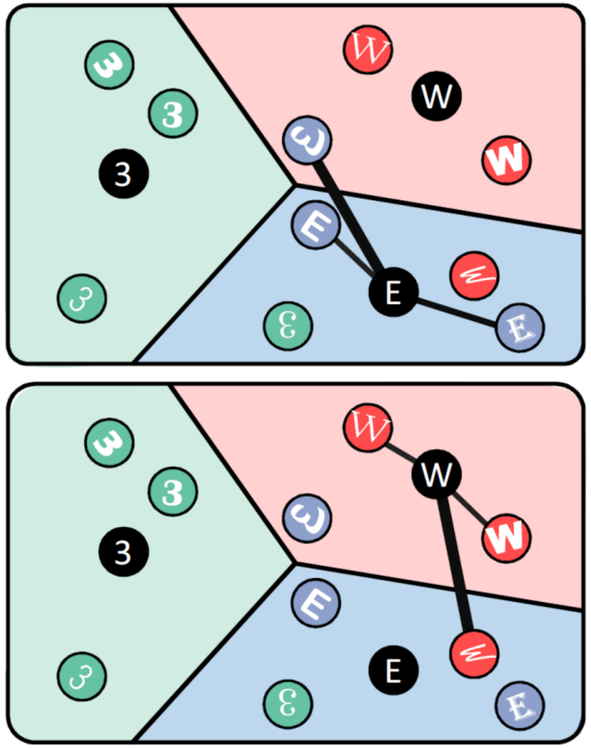
In the case of positive examples

Proxy NCA



Uniform scale
for all gradients

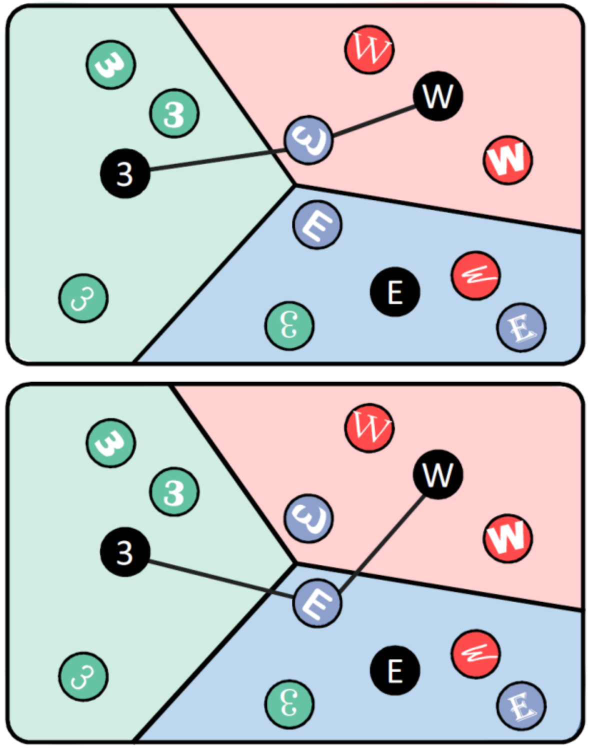
Proxy Anchor



Scales weighted by
relative hardness

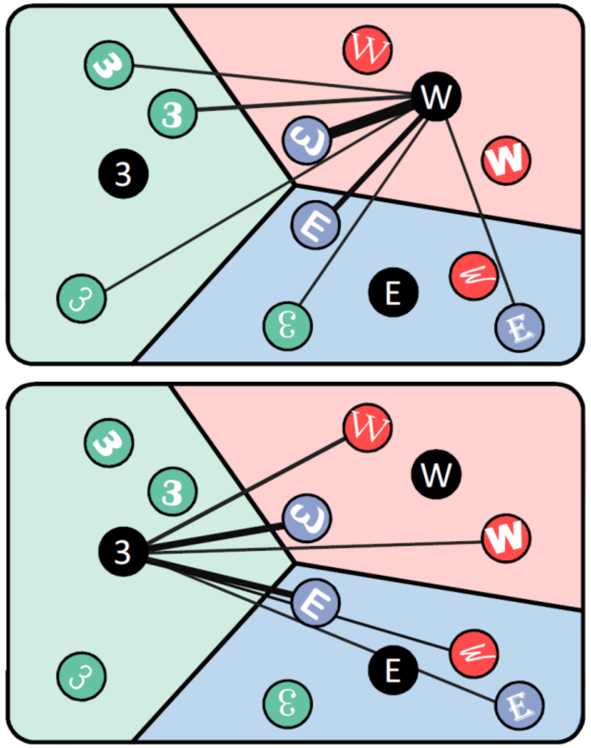
In the case of negative examples

Proxy NCA



Pushing only a small
number of data with
uniform strength

Proxy Anchor



Pushing all data with
consideration of their
distribution

Complexity Analysis

Type	Loss	Training Complexity
Proxy	Proxy Anchor	$O(MC)$
	Proxy NCA[6]	$O(MC)$
	SoftTriplet[8]	$O(MCU^2)$
Pair	Contrastive[4]	$O(M^2)$
	Triplet[1]	$O(M^3)$
	N-pair[5]	$O(M^3)$
	Lifted Structure[7]	$O(M^3)$

The same complexity, but Proxy Anchor converges faster & performs better since it considers relative hardness of data.

M: # of data
C: # of classes ($C \ll M$)
U: # of proxies per class

[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015
[4] Learning a similarity metric discriminatively with application to face verification, CVPR 2005
[5] Improved deep metric learning with multi-class N-pair loss objective, NeurIPS 2016
[6] No fuss distance metric learning using proxies, ICCV 2017
[7] Deep metric learning via lifted structured feature embedding, CVPR 2016
[8] Softtriple loss: Deep metric learning without triplet sampling, ICCV 2019

Experiments

- Evaluation on the 4 image retrieval benchmarks
  - Caltech-UCSD Bird 200 (CUB-200-2011)
  - Cars-196
  - Stanford Online Product (SOP)
  - In-Shop Clothes Retrieval (In-Shop)
- Proxy setting: 1 proxy per class
- Image setting
  - Default: 224 X 224 (as in most previous work)
  - Larger: 256 X 256 (for comparison to HORDE[9])
- Hyper-parameters: $\alpha = 32$, $\delta = 10^{-1}$

[9] High-order regularizer for deep embeddings, ICCV 2019

Experiments

- Quantitative results on the CUB-200-2011 and Cars-196

Recall@K		CUB-200-2011 1	CUB-200-2011 2	CUB-200-2011 4	CUB-200-2011 8	Cars-196 1	Cars-196 2	Cars-196 4	Cars-196 8
Clustering^{64}	BN	48.2	61.4	71.8	81.9	58.1	70.6	80.3	87.8
Proxy-NCA^{64}	BN	49.2	61.9	67.9	72.4	73.2	82.4	86.4	87.8
Smart Mining^{64}	G	49.8	62.3	74.1	83.3	64.7	76.2	84.2	90.2
MS^{64}	BN	57.4	69.8	80.0	87.8	77.3	85.3	90.5	94.2
SoftTriple^{64}	BN	<u>60.1</u>	<u>71.9</u>	<u>81.2</u>	<u>88.5</u>	<u>78.6</u>	<u>86.6</u>	<u>91.8</u>	<u>95.4</u>
Proxy-Anchor^{64}	BN	**61.7**	**73.0**	**81.8**	**88.8**	**78.8**	**87.0**	**92.2**	**95.5**
Margin^{128}	R50	63.6	74.4	83.1	90.0	79.6	86.5	91.9	95.1
HDC^{384}	G	53.6	65.7	77.0	85.6	73.7	83.2	89.5	93.8
A-BIER^{512}	G	57.5	68.7	78.3	86.2	82.0	89.0	93.2	96.1
ABE^{512}	G	60.6	71.5	79.8	87.4	<u>85.2</u>	90.5	94.0	96.1
HTL^{512}	BN	57.1	68.8	78.7	86.5	81.4	88.0	92.7	95.7
RLL-H^{512}	BN	57.4	69.7	79.2	86.9	74.0	83.6	90.1	94.1
MS^{512}	BN	<u>65.7</u>	<u>77.0</u>	<u>86.3</u>	<u>91.2</u>	84.1	90.4	94.0	96.5
SoftTriple^{512}	BN	65.4	76.4	84.5	90.4	84.5	<u>90.7</u>	<u>94.5</u>	<u>96.9</u>
Proxy-Anchor^{512}	BN	**68.4**	**79.2**	**86.8**	**91.6**	**86.1**	**91.7**	**95.0**	**97.3**
†Contra+HORDE^{512}	BN	66.3	76.7	84.7	90.6	83.9	90.3	94.1	96.3
†Proxy-Anchor^{512}	BN	**71.1**	**80.4**	**87.4**	**92.5**	**88.3**	**93.1**	**95.7**	**97.5**

Experiments

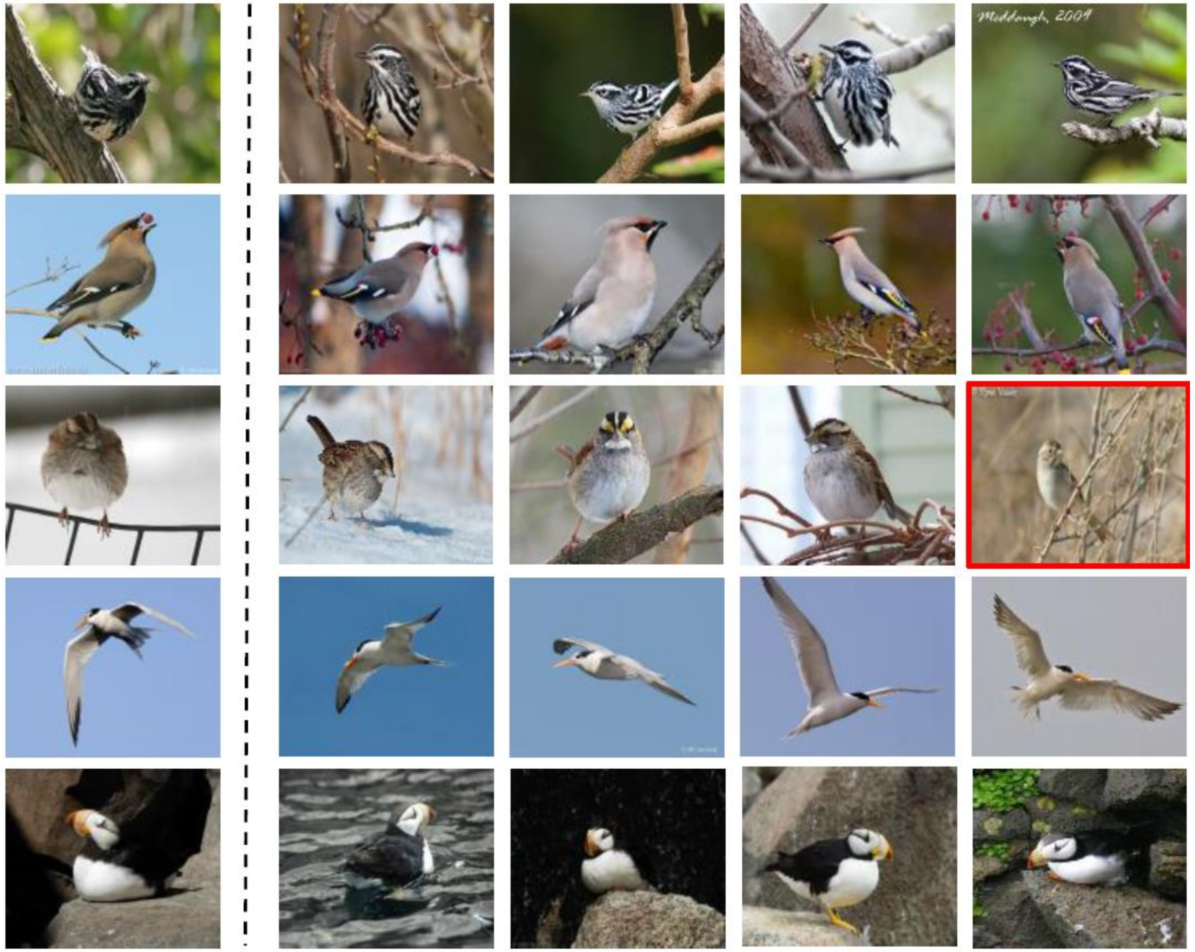
- Quantitative results on the SOP (*left*) and In-Shop (*right*)

Recall@K	1	10	100	1000
Clustering^{64}	67.0	83.7	93.2	-
Proxy-NCA^{64}	73.7	-	-	-
MS^{64}	74.1	87.8	94.7	**98.2**
SoftTriple^{64}	<u>76.3</u>	**89.1**	**95.3**	-
Proxy-Anchor^{64}	**76.5**	<u>89.0</u>	<u>95.1</u>	**98.2**
Margin^{128}	72.7	86.2	93.8	98.0
HDC^{384}	69.5	84.4	92.8	97.7
A-BIER^{512}	74.2	86.9	94.0	97.8
ABE^{512}	76.3	88.4	94.8	98.2
HTL^{512}	74.8	88.3	94.8	98.4
RLL-H^{512}	76.1	89.1	95.4	-
MS^{512}	78.2	<u>90.5</u>	<u>96.0</u>	**98.7**
SoftTriple^{512}	<u>78.3</u>	90.3	95.9	-
Proxy-Anchor^{512}	**79.1**	**90.8**	**96.2**	**98.7**
†Contra+HORDE^{512}	80.1	91.3	96.2	**98.7**
†Proxy-Anchor^{512}	**80.3**	**91.4**	**96.4**	**98.7**

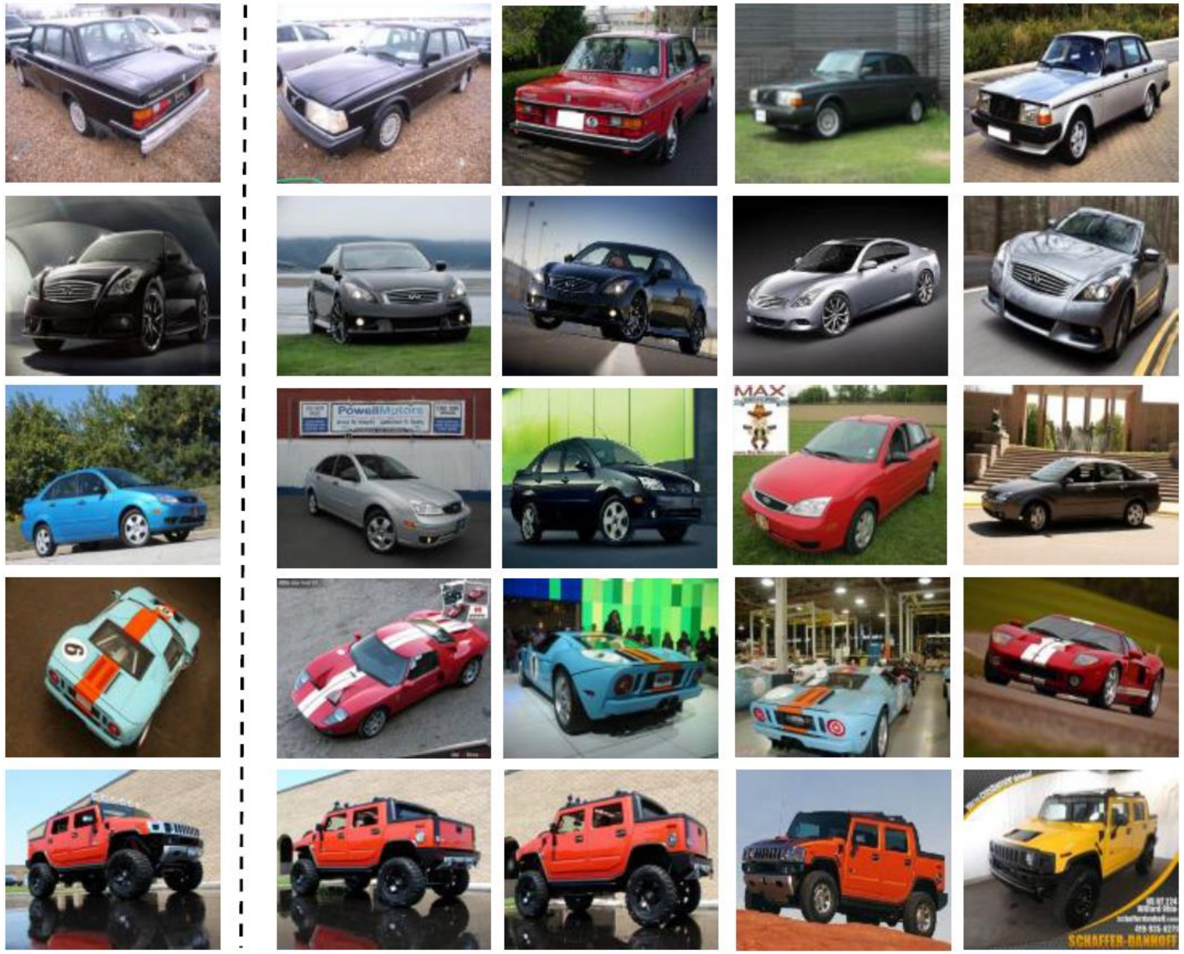
Recall@K	1	10	20	40
HDC^{384}	62.1	84.9	89.0	92.3
HTL^{128}	80.9	94.3	95.8	97.4
MS^{128}	<u>88.0</u>	<u>97.2</u>	<u>98.1</u>	<u>98.7</u>
Proxy-Anchor^{128}	**90.8**	**97.9**	**98.5**	**99.0**
FashionNet^{4096}	53.0	73.0	76.0	79.0
A-BIER^{512}	83.1	95.1	96.9	97.8
ABE^{512}	87.3	96.7	97.9	98.5
MS^{512}	<u>89.7</u>	<u>97.9</u>	<u>98.5</u>	<u>99.1</u>
Proxy-Anchor^{512}	**91.5**	**98.1**	**98.8**	**99.1**
†Contra+HORDE^{512}	90.4	97.8	98.4	98.9
†Proxy-Anchor^{512}	**92.6**	**98.3**	**98.9**	**99.3**

Our method achieves state-of-the-art performance in almost all settings on the all 4 benchmarks.

- Qualitative results: Top 4 retrievals



CUB-200-2011

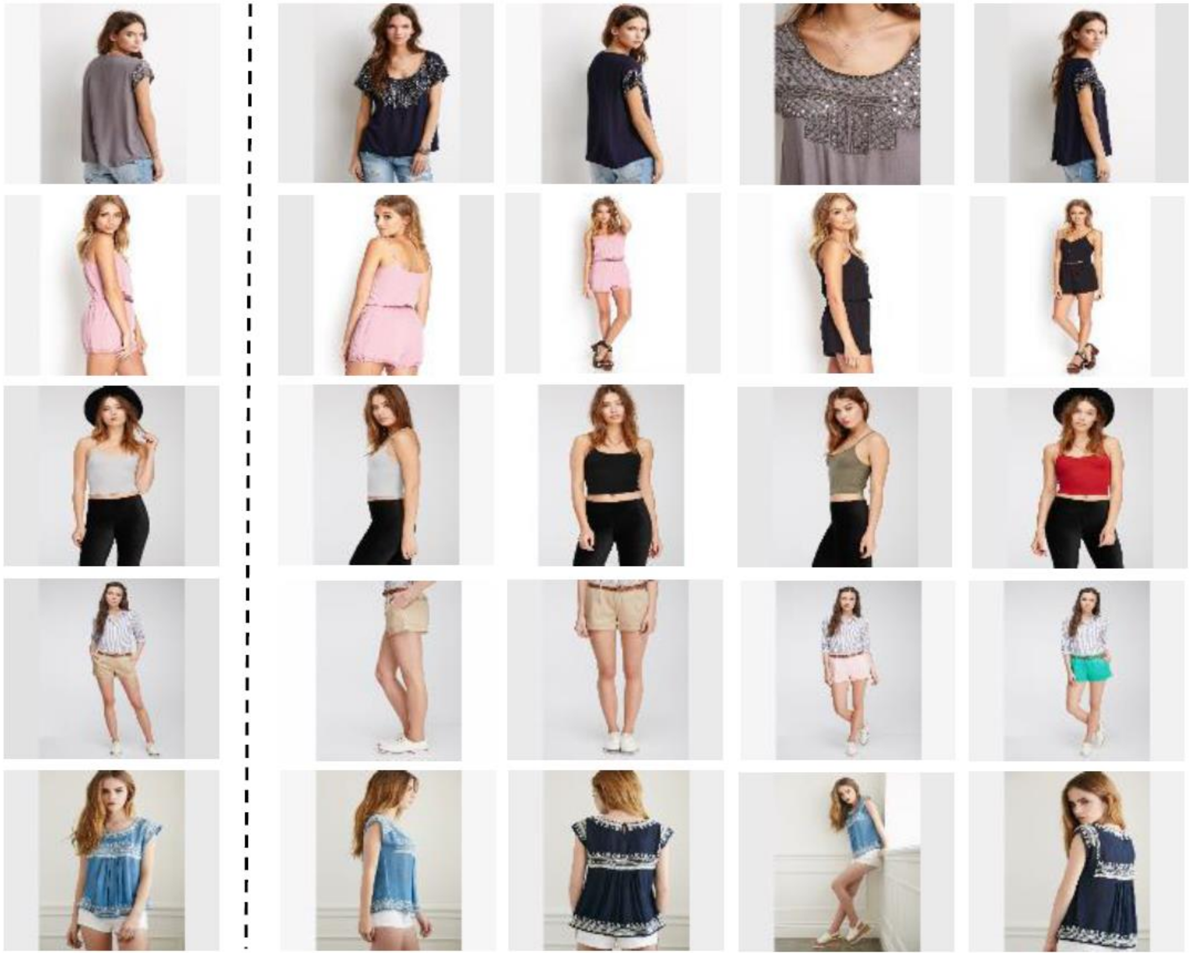


Cars-196

Experiments

- Qualitative results: Top 4 retrievals

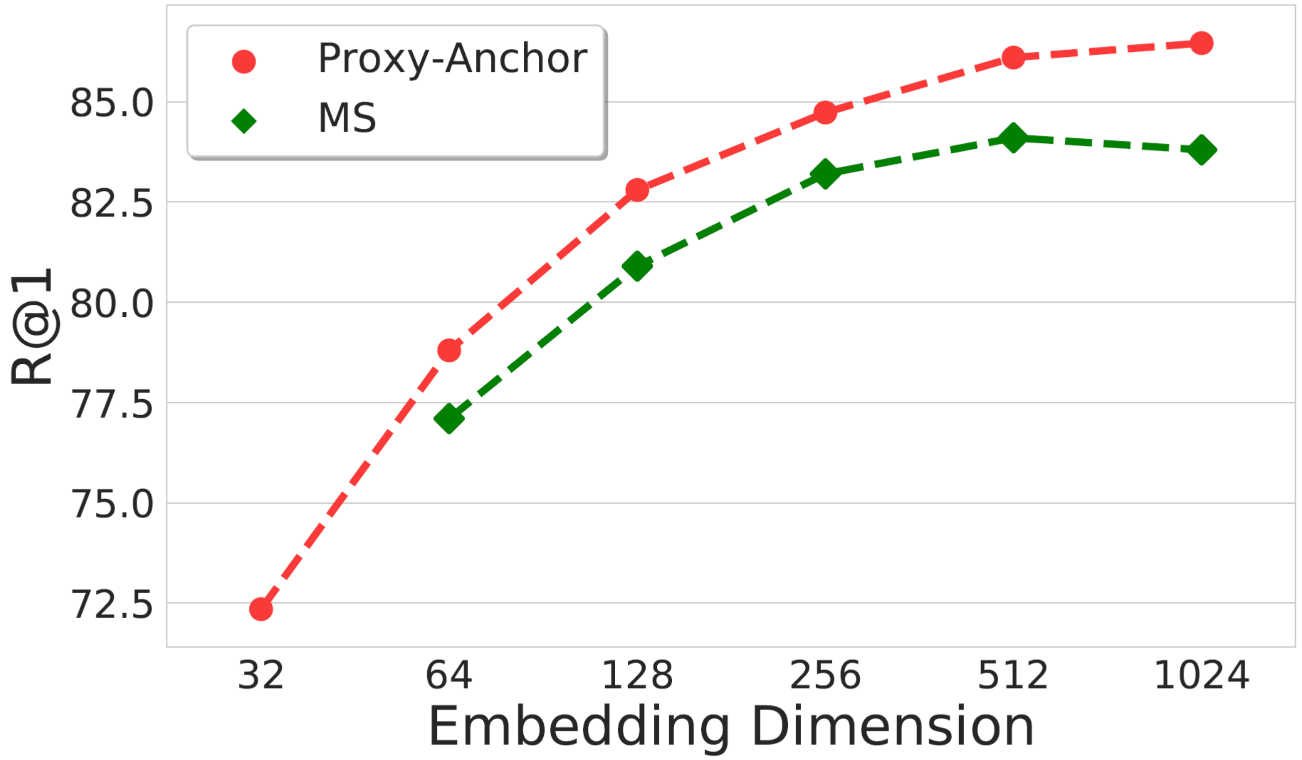
SOP



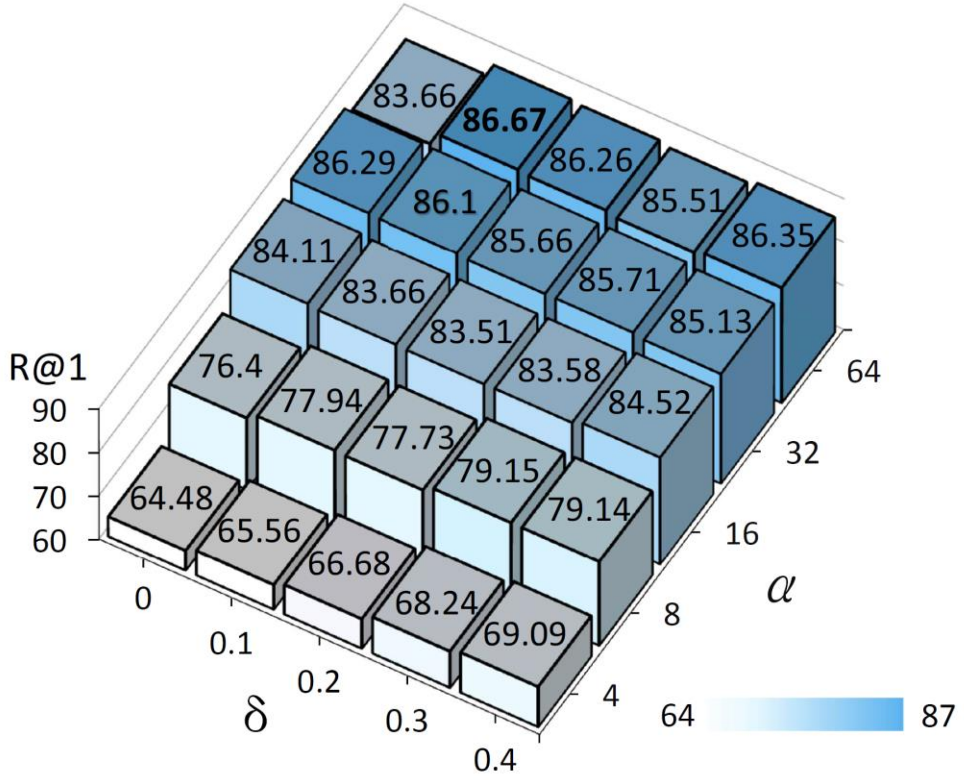
In-Shop

Experiments

- Impact of hyper-parameters



Accuracy vs. embedding dimension



Accuracy vs. α and δ

The performance is stable and high enough when the embedding dimension ≥ 128 and $\alpha \geq 16$.

Experiments

- Ablation studies

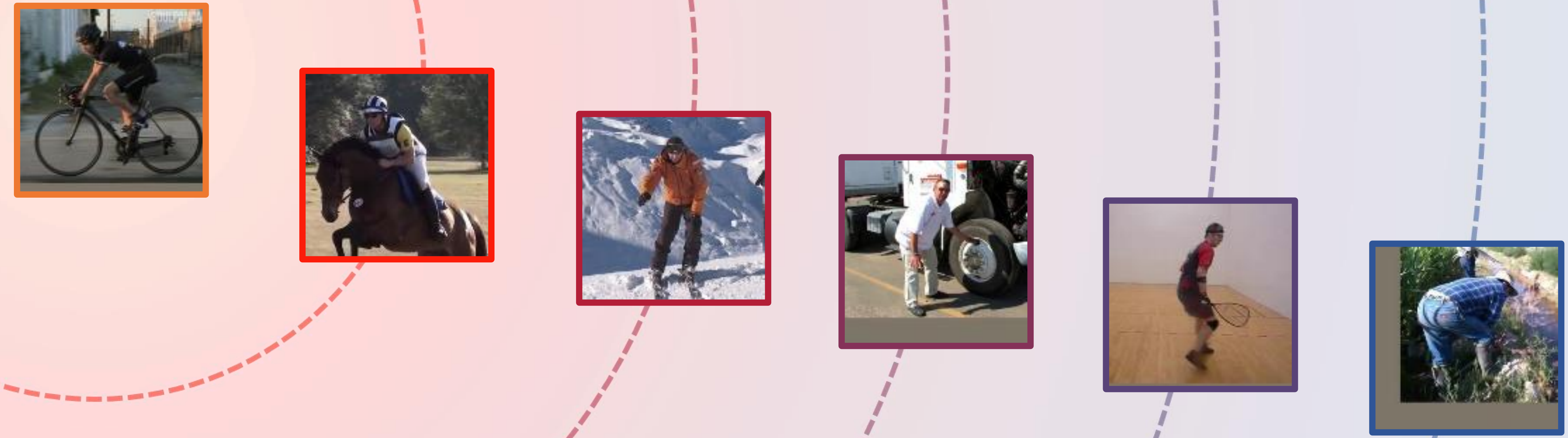
Network	Image Size	CUB-200-2011				Cars-196			
		R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8
GoogleNet	224×224	63.8	74.4	83.6	90.4	84.3	90.4	94.1	96.7
Inception-BN		68.4	79.2	86.8	91.6	86.1	91.7	95.0	97.3
ResNet-50		69.7	80.0	87.0	92.4	87.7	92.9	95.8	97.9
ResNet-101		**70.8**	**81.0**	**88.1**	**93.0**	**87.9**	**93.0**	**96.1**	**97.9**
Inception-BN	256×256	71.1	80.4	87.4	92.5	88.3	93.1	95.7	97.5
	324×324	74.0	82.9	88.9	93.2	91.1	94.9	96.9	98.3
	448×448	**77.3**	**85.6**	**91.1**	**94.2**	**92.9**	**96.1**	**97.7**	**98.7**

Strong backbone and large input improve performance.

Conclusion

- Contributions
  - A new metric learning loss based on proxy
  - Current state of the art on public benchmarks for image retrieval
  - Fastest convergence speed

- Future directions
  - Analysis on generalizability
  - Improving test time efficiency



Deep Metric Learning Beyond Binary Supervision

Sungyeon Kim Minkyo Seo Ivan Laptev Minsu Cho Suha Kwak

{tjddus9597, mkseo, mscho, suha.kwak}@postech.ac.kr, ivan.laptev@inria.fr

POSTECH Inria

Existing Losses in Deep Metric Learning

- A common issue
  - Existing (deep) metric learning approaches rely on binary relations between images: "*same*" or "*not*".

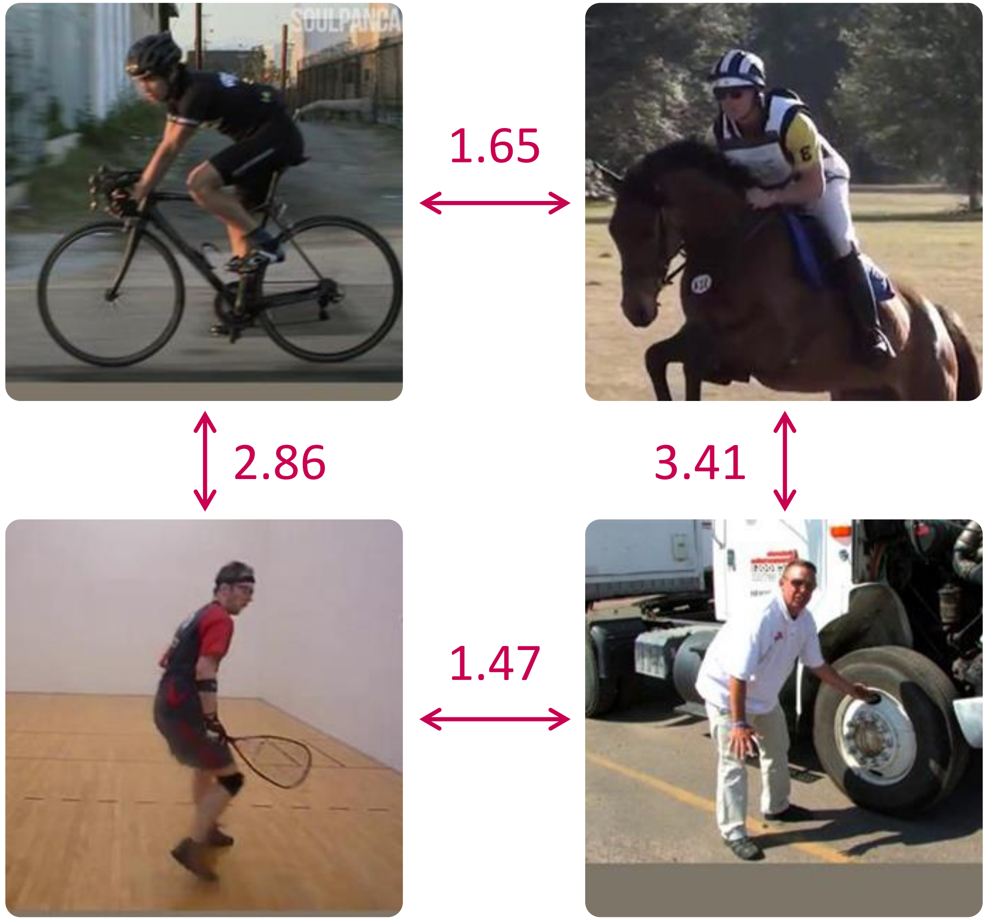
Face verification

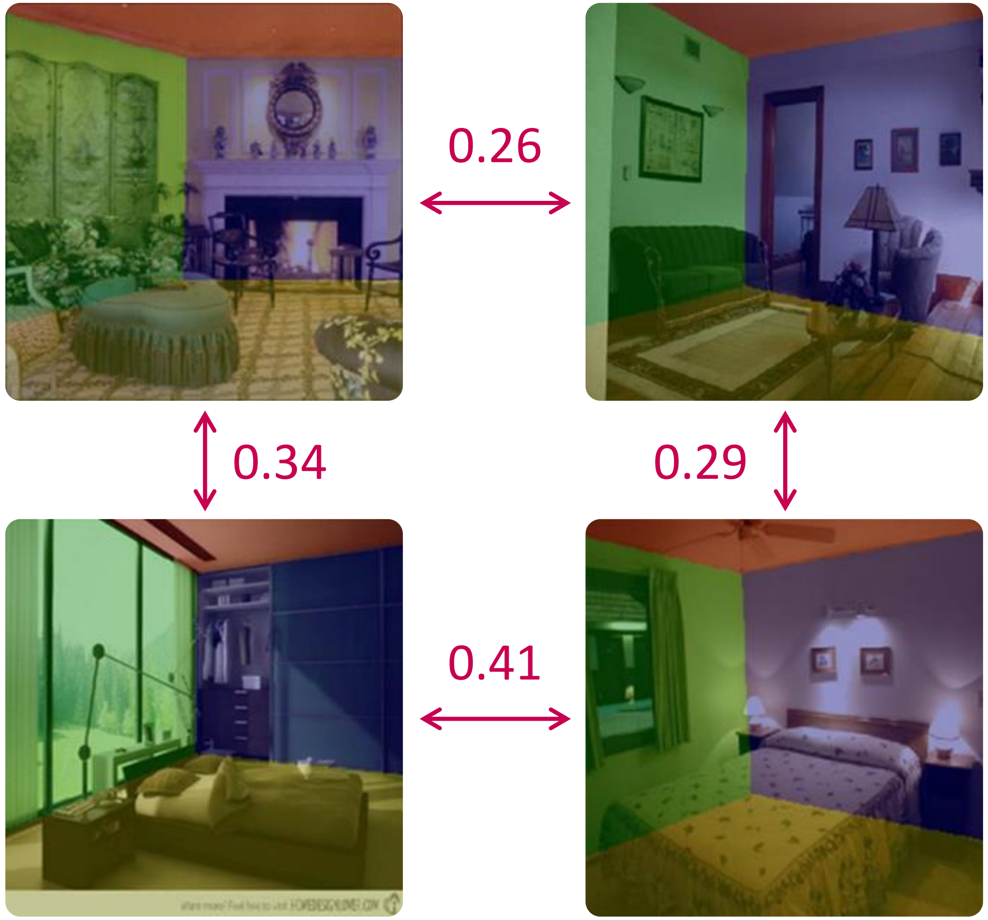
Content-based image retrieval

Person re-identification

Existing Losses in Deep Metric Learning

- A common issue
  - However, relations between real world images are *not binary* but often represented as *continuous similarities*.



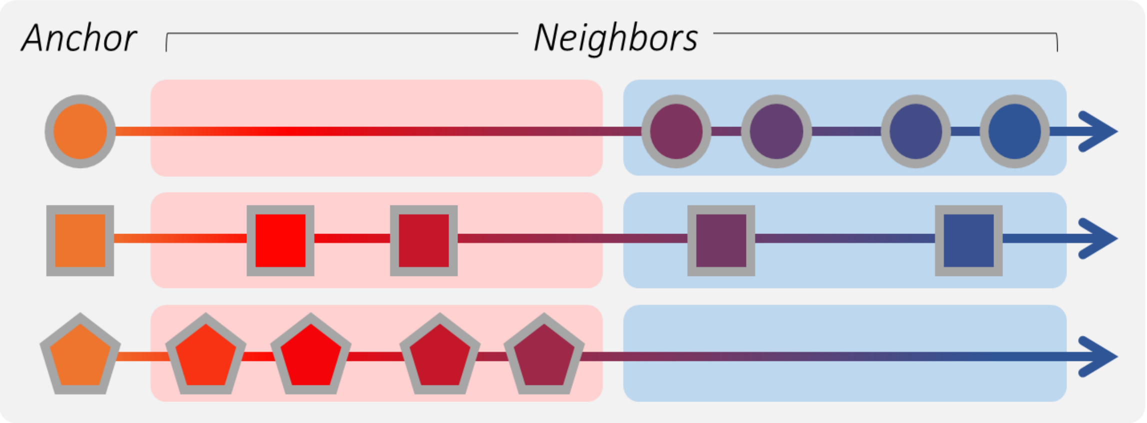


Existing Losses in Deep Metric Learning

- Conventional methods to handle the issue
  - Existing metric learning loss + *similarity quantization*

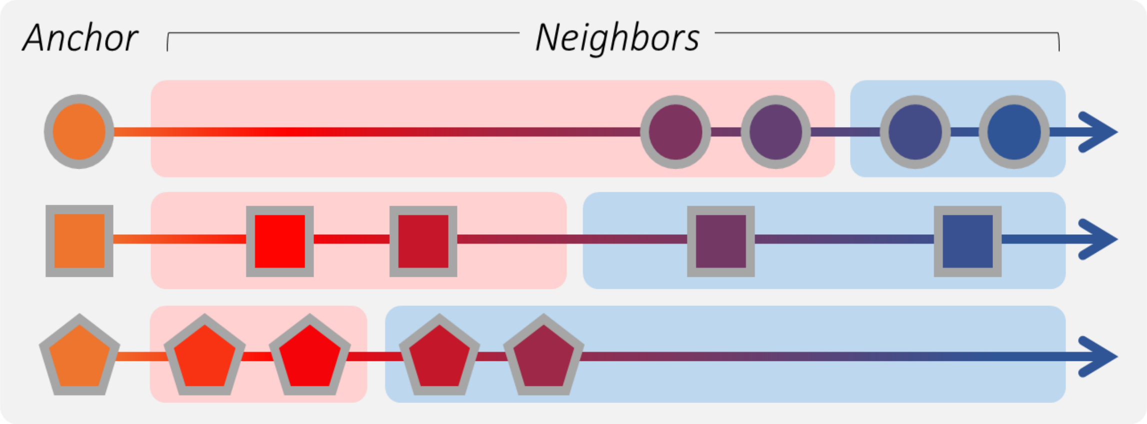
Binary thresholding[9]

Populations of positive and negative examples would be significantly imbalanced.



Nearest neighbor search[10]

Positive neighbors of a rare example would be dissimilar and negative neighbors of a common example would be too similar.



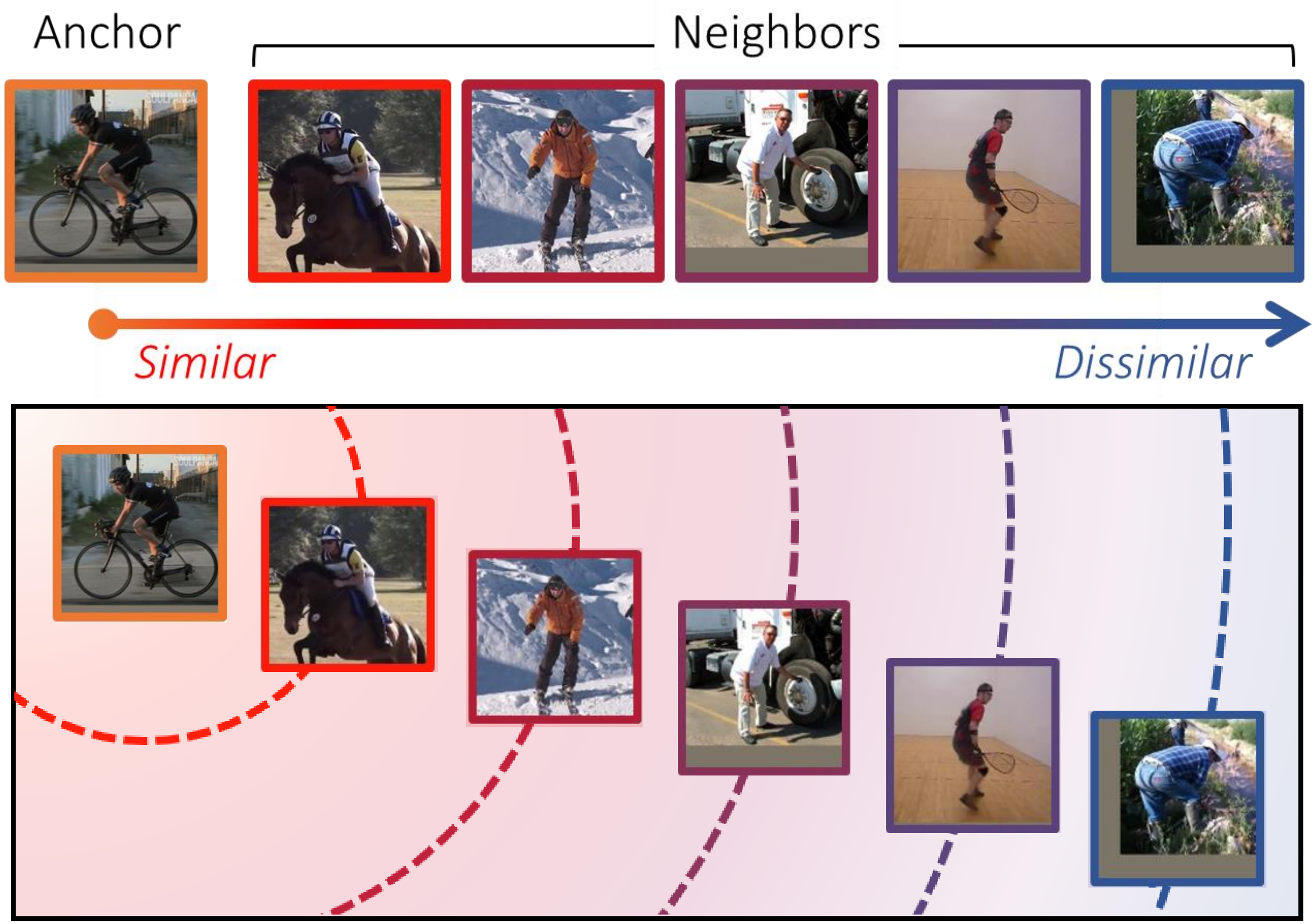
[9] Pose embeddings: A deep architecture for learning to match human poses, arXiv 2015
[10] Thin-slicing for pose: Learning to understand pose without explicit pose estimation, CVPR 2016

Existing Losses in Deep Metric Learning

- Conventional methods to handle the issue
  - Degree of similarity is ignored in the learned embedding space.

Our Method

- Our goal
  - Learning a metric space that reflects the degree of similarity directly



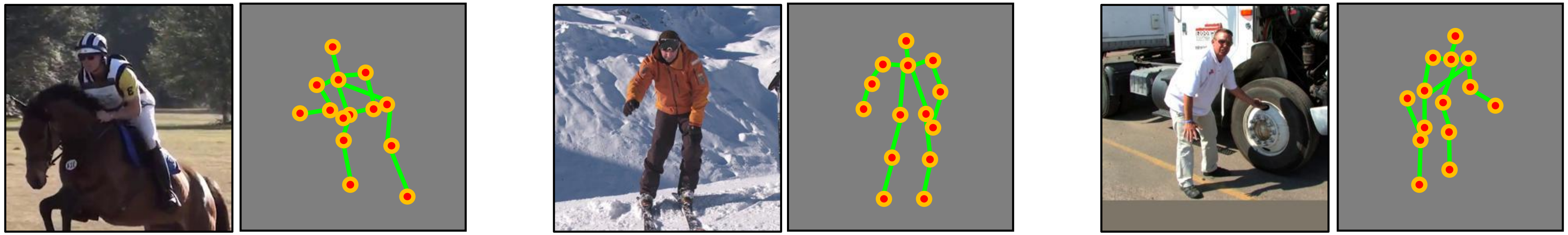
Our Method

- Our goal
  - Learning a metric space that reflects the degree of similarity directly

- Contributions
  - A new triplet loss: *Log-ratio loss*
  - A new triplet sampling technique: *Dense triplet sampling*
  - Various applications
    - Human pose retrieval
    - Room layout retrieval
    - Caption-aware image retrieval
    - Representation learning for image captioning

Log-ratio Loss

- Definition



$\mathbf{x}_a$ $\boldsymbol{y}_a$ $\mathbf{x}_i$ $\boldsymbol{y}_i$ $\mathbf{x}_j$ $\boldsymbol{y}_j$

$$\ell_{\mathrm{lr}}(a,i,j) = \left\{ \log \frac{D(f_a, f_i)}{D(f_a, f_j)} - \log \frac{D_{\mathbf{y}}(\boldsymbol{y}_a, \boldsymbol{y}_i)}{D_{\mathbf{y}}(\boldsymbol{y}_a, \boldsymbol{y}_j)} \right\}^2$$

where $f_i \coloneqq f(\mathbf{x}_i)$ is the embedding vector of image i,
and $D(\cdot)$ denotes the squared Euclidean distance.

The distance between two images in the learned metric space
will be proportional to their distance in the label space.

Log-ratio Loss

- Analysis on its gradients

$$\frac{\partial \ell_{\mathrm{lr}}(a,i,j)}{\partial f_a} = -\frac{\partial \ell_{\mathrm{lr}}(a,i,j)}{\partial f_i} - \frac{\partial \ell_{\mathrm{lr}}(a,i,j)}{\partial f_j}$$

$$\frac{\partial \ell_{\mathrm{lr}}(a,i,j)}{\partial f_i} = \frac{(f_i - f_a)}{D(f_a, f_i)} \cdot \ell'_{\mathrm{lr}}(a,i,j)$$

$$\frac{\partial \ell_{\mathrm{lr}}(a,i,j)}{\partial f_j} = \frac{(f_a - f_j)}{D(f_a, f_j)} \cdot \ell'_{\mathrm{lr}}(a,i,j)$$

Direction between
the anchor and neighbors

Discrepancy between
the label distance ratio and
the embedding distance ratio

$$4\left\{\log \frac{D(f_a, f_i)}{D(f_a, f_j)} - \log \frac{D_{\mathbf{y}}(\mathbf{y}_a, \mathbf{y}_i)}{D_{\mathbf{y}}(\mathbf{y}_a, \mathbf{y}_j)}\right\}$$

Log-ratio Loss

- Comparison to the triplet rank loss

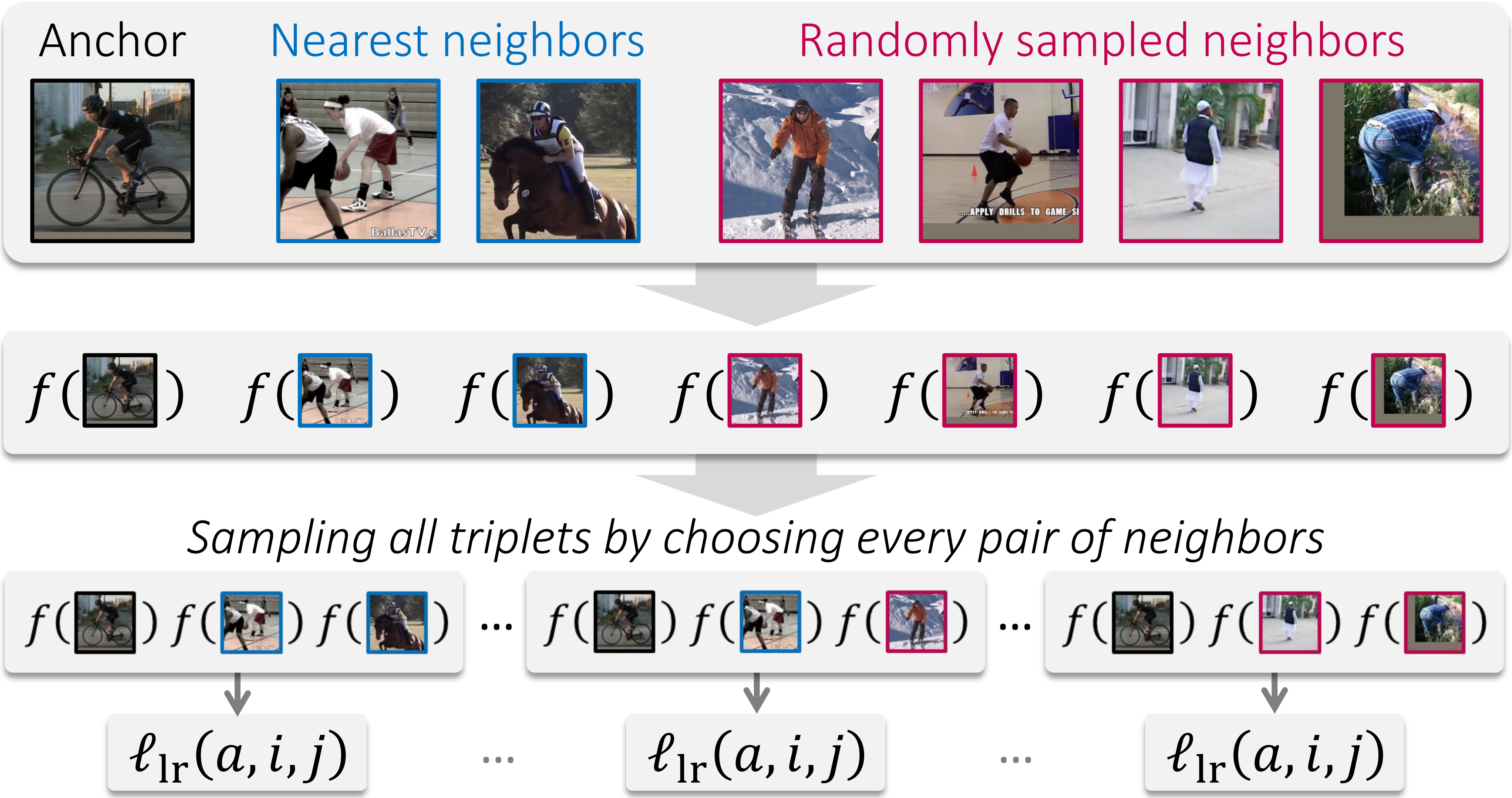
Log-ratio loss	*Triplet rank loss*
$\ell_{\mathrm{lr}}(a,i,j) = \left\{\log\frac{D(f_a,f_i)}{D(f_a,f_j)} - \log\frac{D(y_a,y_i)}{D(y_a,y_j)}\right\}^2$	$\ell_{\mathrm{tri}}(a,i,j) = \left[D(f_a,f_i) - D(f_a,f_j) + \delta\right]_+$
$\frac{\partial\ell_{\mathrm{lr}}(a,i,j)}{\partial f_a} = -\frac{\partial\ell_{\mathrm{lr}}(a,i,j)}{\partial f_i} - \frac{\partial\ell_{\mathrm{lr}}(a,i,j)}{\partial f_j}$	$\frac{\partial\ell_{\mathrm{tri}}(a,i,j)}{\partial f_a} = -\frac{\partial\ell_{\mathrm{tri}}(a,i,j)}{\partial f_i} - \frac{\partial\ell_{\mathrm{tri}}(a,i,j)}{\partial f_j}$
$\frac{\partial\ell_{\mathrm{lr}}(a,i,j)}{\partial f_i} = \frac{(f_i - f_a)}{D(f_a,f_i)} \cdot \ell'_{\mathrm{lr}}(a,i,j)$	$\frac{\partial\ell_{\mathrm{tri}}(a,i,j)}{\partial f_i} = 2(f_i - f_a) \cdot \mathbb{I}(\ell_{\mathrm{tri}}(a,i,j) > 0)$
$\frac{\partial\ell_{\mathrm{lr}}(a,i,j)}{\partial f_j} = \frac{(f_a - f_j)}{D(f_a,f_j)} \cdot \ell'_{\mathrm{lr}}(a,i,j)$	$\frac{\partial\ell_{\mathrm{tri}}(a,i,j)}{\partial f_j} = 2(f_a - f_j) \cdot \mathbb{I}(\ell_{\mathrm{tri}}(a,i,j) > 0)$
Although the rank constraint holds, the gradients' magnitudes could be significant if $\ell'_{\mathrm{lr}}(a,i,j)$ is large.	The gradients are zero if the triplet satisfies the rank constraint due to the indicator $\mathbb{I}(\ell_{\mathrm{tri}}(a,i,j) > 0)$.

Log-ratio Loss

- Compared to the triplet rank loss, our loss
  - Captures continuous similarities between images better,
    (the triplet rank loss focuses only on partial ranks of similarities.)
  - Does not require any hyperparameter,
    (for the triplet rank loss the margin should be tuned carefully.)
  - Does not demand L_2 normalization of the embedding vectors,
    (such a normalization is essential for the triplet rank loss.)
  - Performs much better with a low embedding dimension.

Dense Triplet Sampling

- Main idea: Using all triplets within a minibatch



Dense Triplet Sampling

- Why not using existing sampling techniques[1,11]
  - They rely on binary relations between images.
  - They are designed to be combined with conventional triplet losses.
  - The notion of hardness is not clear in our setting.

- Our sampling strategy is well matched with the log-ratio loss.
  - The log-ratio loss enables every triplet to well contribute to training.

$$\frac{\partial \ell_{\mathrm{lr}}(a,i,j)}{\partial f_i} = \frac{(f_i - f_a)}{D(f_a, f_i)} \cdot 4\left\{\log \frac{D(f_a, f_i)}{D(f_a, f_j)} - \log \frac{D_{\boldsymbol{y}}(\boldsymbol{y}_a, \boldsymbol{y}_i)}{D_{\boldsymbol{y}}(\boldsymbol{y}_a, \boldsymbol{y}_j)}\right\}$$

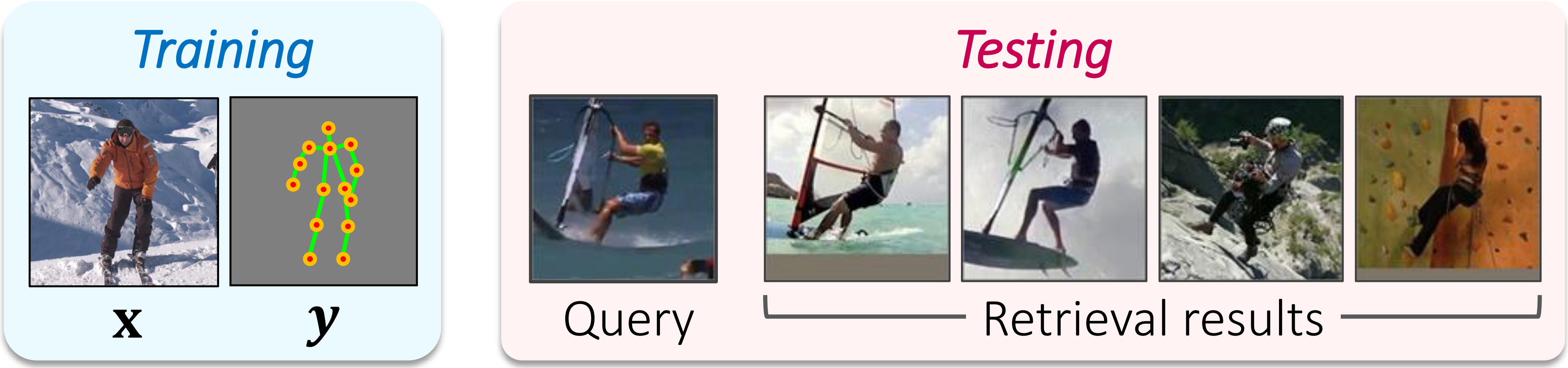
Non-trivial even if the triplet complies the rank constraint

  - *Exploiting all triplets improves embedding performance.*

[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015
[11] Sampling matters in deep embedding learning, ICCV 2017

Experiments – Three Retrieval Tasks

- Human pose retrieval

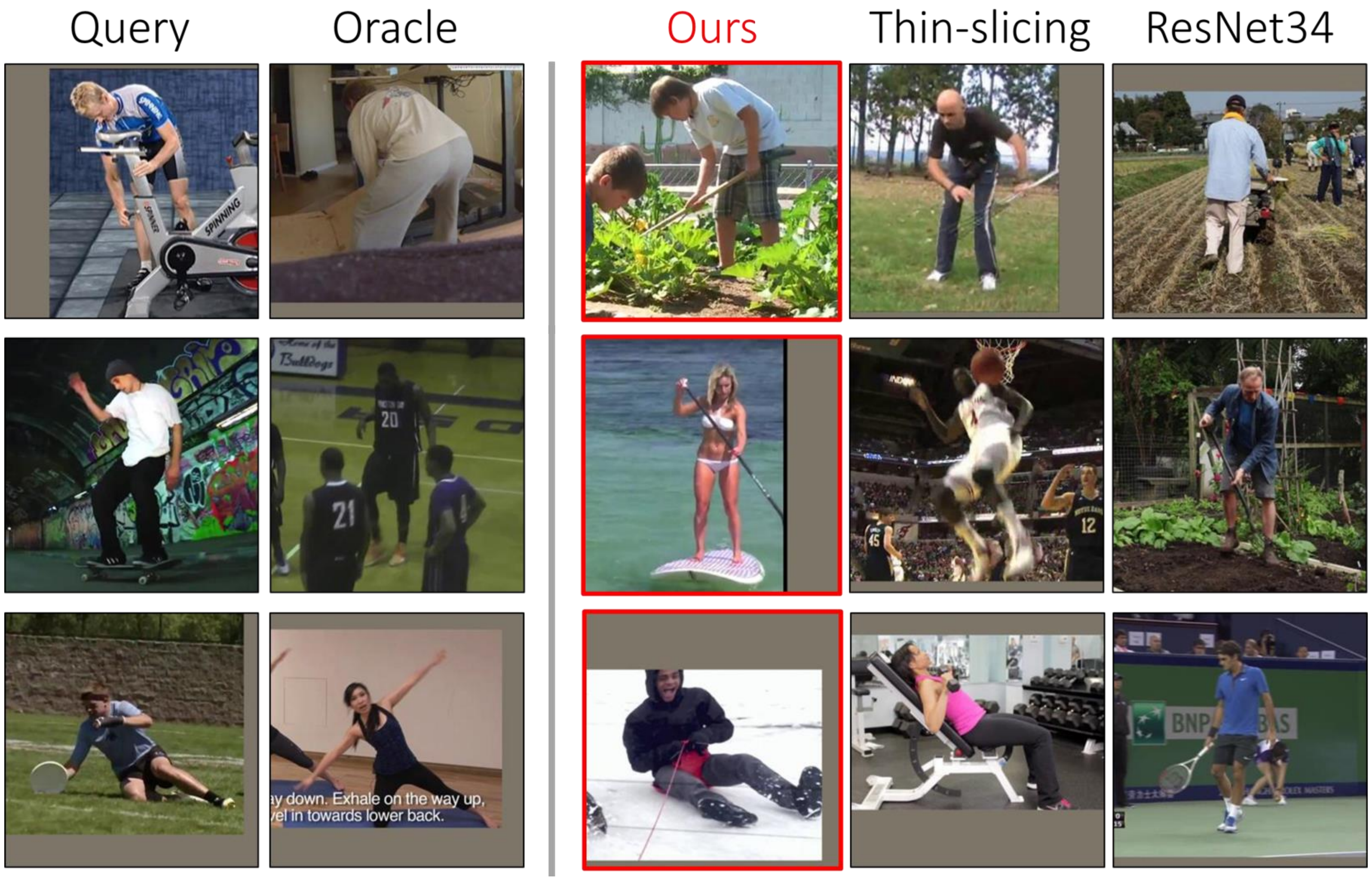


- Conducted on the *MPII human pose dataset*
- Application: *pose-aware representation for action recognition*
- Label distance between images:

$$D_{\mathbf{y}}(\boldsymbol{y}_i, \boldsymbol{y}_j) = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2.$$

Experiments – Three Retrieval Tasks

- Human pose retrieval



ResNet34: ImageNet pre-trained network

Typically focuses on objects or background other than human poses.

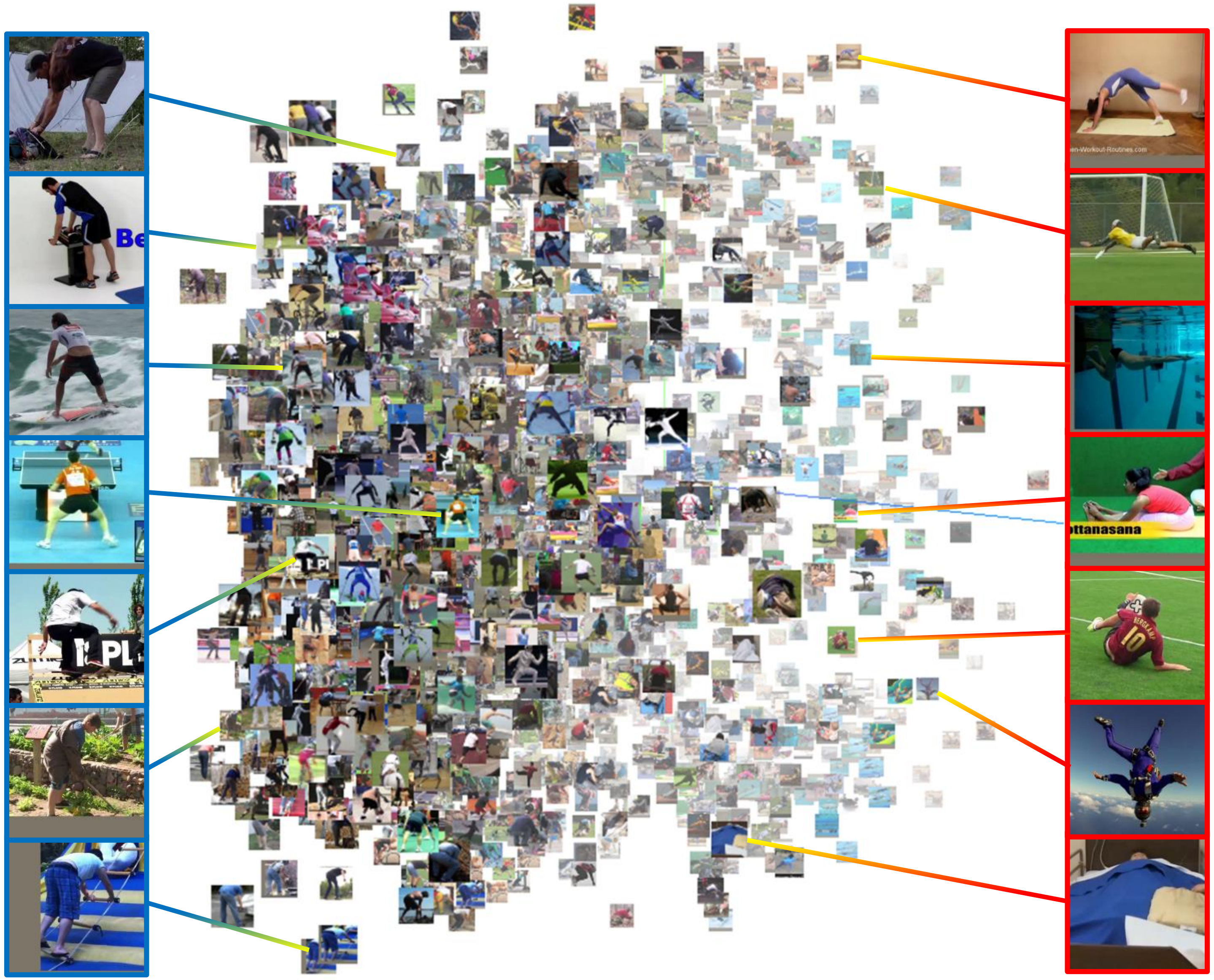
Thin-slicing[10]: A previous work on pose embedding

Often fails to address rare human poses.

[10] Thin-slicing for pose: Learning to understand pose without explicit pose estimation, CVPR 2016

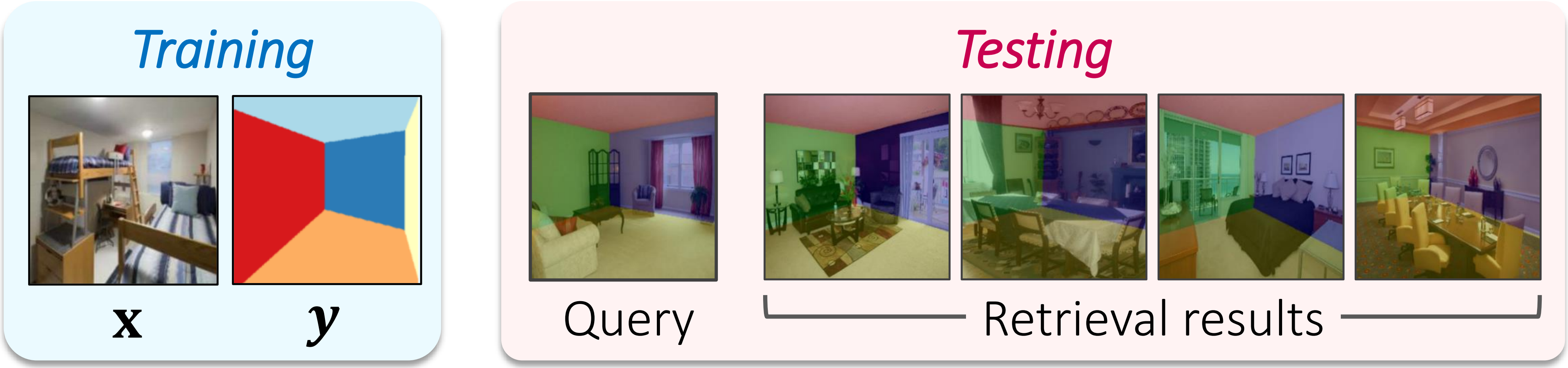
Experiments – Three Retrieval Tasks

- Human pose retrieval



Experiments – Three Retrieval Tasks

- Room layout retrieval



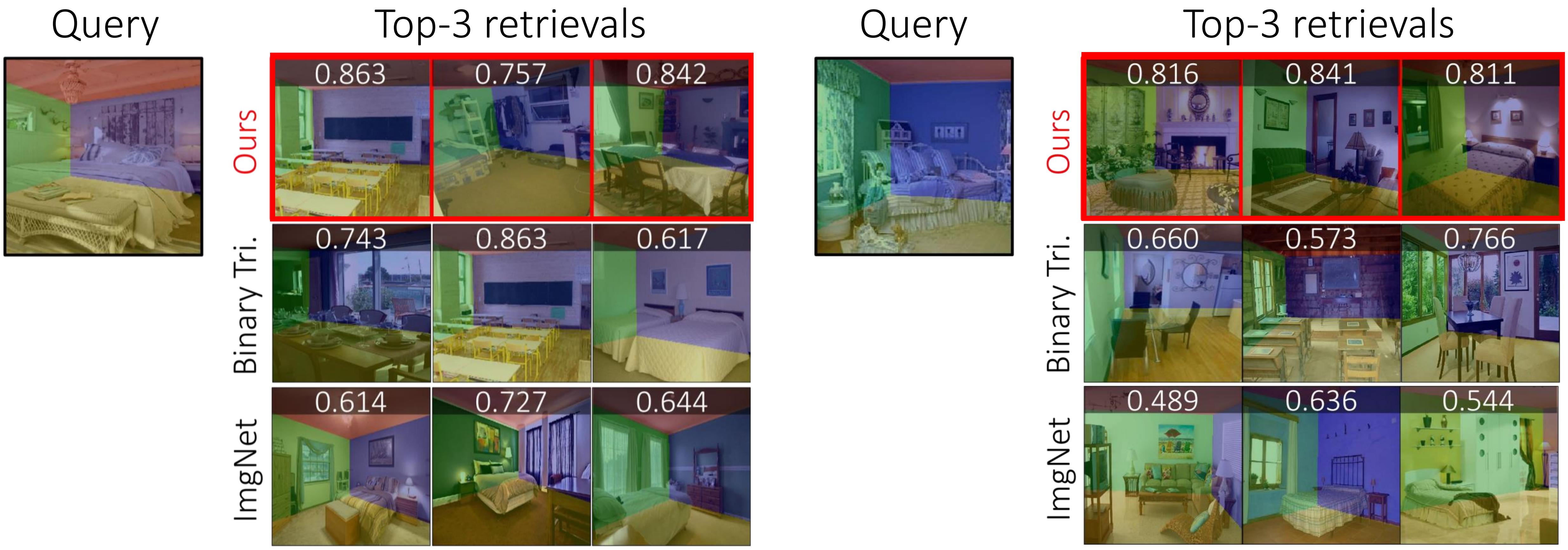
- Conducted on the *LSUN room layout dataset*
- Label distance between images:

$$D_{\boldsymbol{y}}(\boldsymbol{y}_i, \boldsymbol{y}_j) = 1 - \mathrm{mIoU}(\boldsymbol{y}_i, \boldsymbol{y}_j),$$

where $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ denote groundtruth room segmentations

Experiments – Three Retrieval Tasks

- Room layout retrieval



Binary Tri.: Triplet rank loss + Binary thresholding

ImgNet: ImageNet pre-trained ResNet101

Experiments – Three Retrieval Tasks

- Caption-aware image retrieval

Training

A girl holding a frisbee

$\mathbf{x}$ $\boldsymbol{y}$

Testing

Query



Retrieval results

- Conducted on the *MS-COCO 2014 caption dataset*
- Label distance between images:

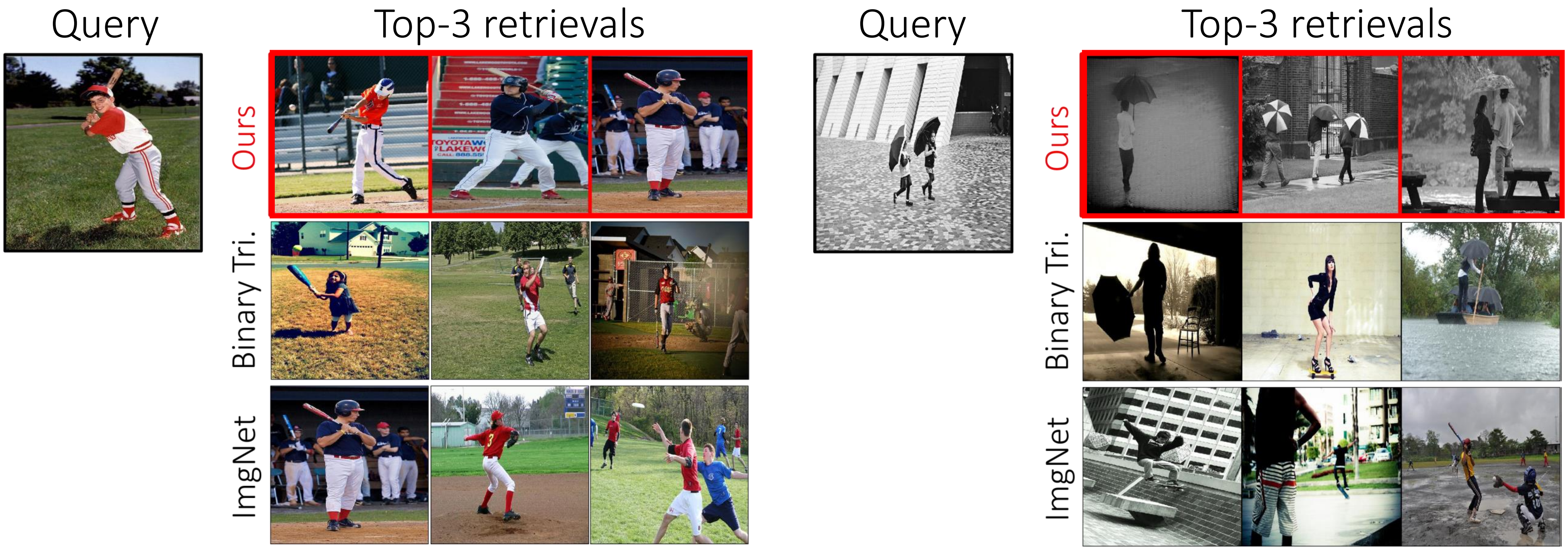
$$D_{\boldsymbol{y}}(\boldsymbol{y}_i, \boldsymbol{y}_j) = \sum_{c_i \in \boldsymbol{y}_i} \min_{c_j \in \boldsymbol{y}_j} W(c_i, c_j) + \sum_{c_j \in \boldsymbol{y}_j} \min_{c_i \in \boldsymbol{y}_i} W(c_i, c_j),$$

where $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are sets of 5 captions and $W(\cdot)$ is the WMD[12] between two captions

[12] From word embeddings to document distances, ICML 2015

Experiments – Three Retrieval Tasks

- Caption-aware image retrieval

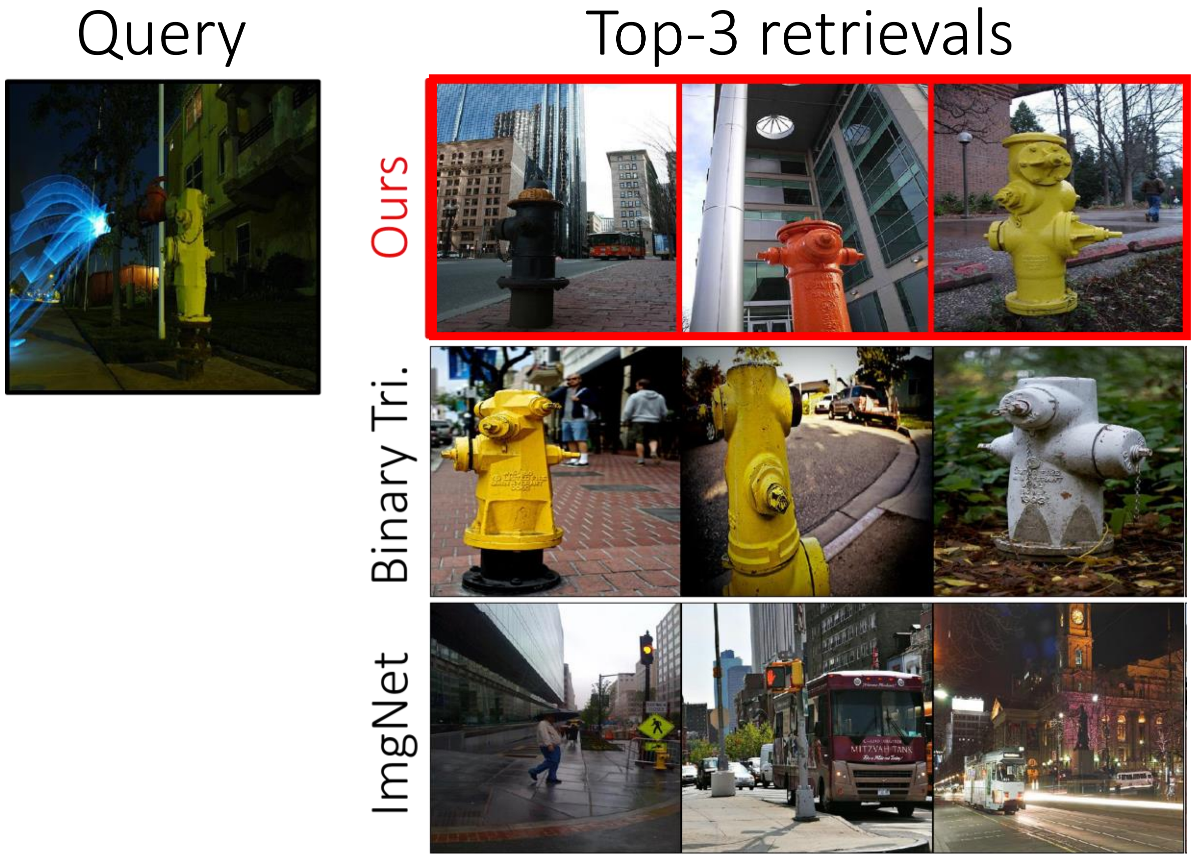


Binary Tri.: Triplet rank loss + Binary thresholding

ImgNet: ImageNet pre-trained ResNet101

Experiments – Three Retrieval Tasks

- Caption-aware image retrieval

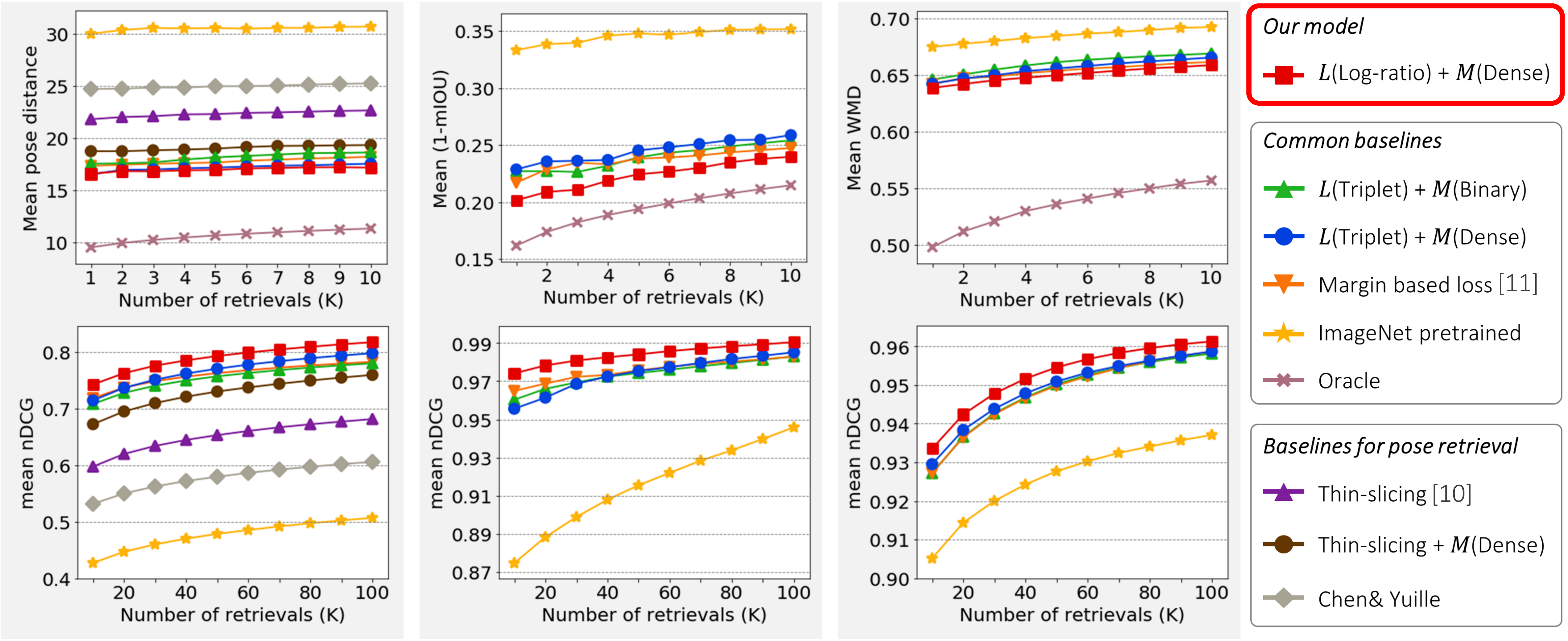


Binary Tri.: Triplet rank loss + Binary thresholding

ImgNet: ImageNet pre-trained ResNet101

Experiments – Three Retrieval Tasks

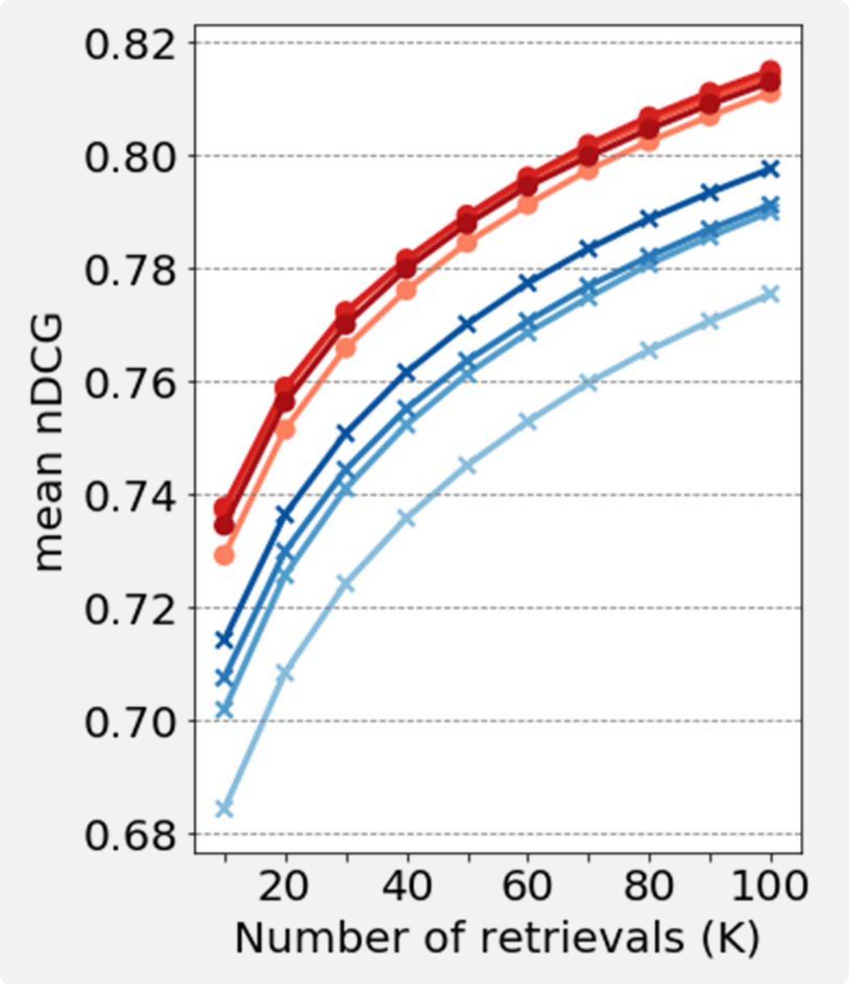
- Quantitative performance analysis



Human pose retrieval Room layout retrieval Caption-aware image retrieval

Experiments – Three Retrieval Tasks

- Embedding dimension vs. retrieval performance



Our models

- L(Log-ratio) + M(Dense) 128-D
- L(Log-ratio) + M(Dense) 64-D
- L(Log-ratio) + M(Dense) 32-D
- L(Log-ratio) + M(Dense) 16-D

Baselines

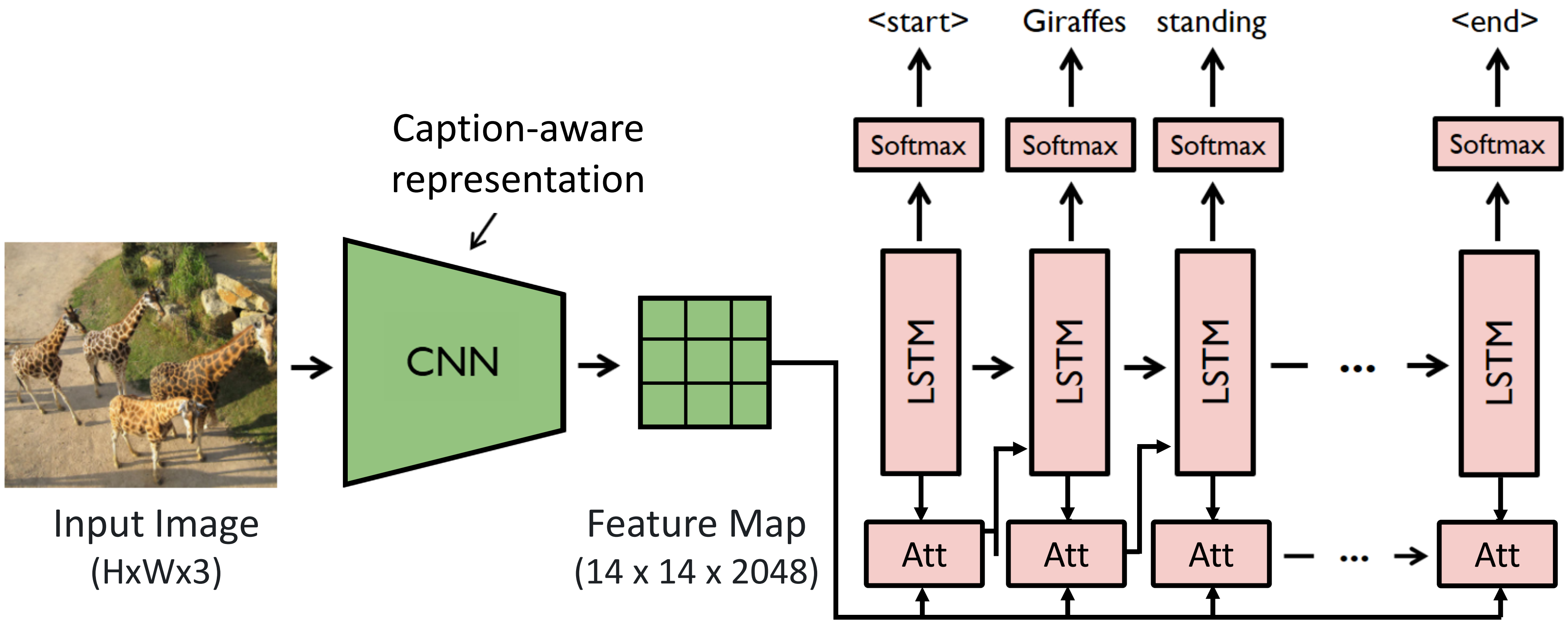
- L(Triplet) + M(Dense) 128-D
- L(Triplet) + M(Dense) 64-D
- L(Triplet) + M(Dense) 32-D
- L(Triplet) + M(Dense) 16-D

L(Log-ratio) + M(Dense): Log-ratio loss + Dense triplet sampling

L(Triplet) + M(Dense): Triplet rank loss + Dense triplet sampling

Experiments – Representation Learning

- Representation learning for image captioning



Our approach

Using the caption embedding network trained with caption similarities as an initial visual representation for image captioning

Experiments – Representation Learning

- Quantitative results

115.9 in CIDEr
Caption-aware feature + RL

+2.5%

113.1 in CIDEr
ImageNet pretrained feature + RL

34.65 in BLEU-4
Caption-aware feature + RL

+3.5%

33.48 in BLEU-4
ImageNet pretrained feature + RL

[13] Self-critical sequence training for image captioning, CVPR 2017
[14] Bottom-up and top-down attention for image captioning and visual question answering, CVPR 2018

Experiments – Representation Learning

- Qualitative results obtained by the top-down attention model

GT1	There are some zebras standing in a grassy field
GT2	A field with tall grass, bushes and trees, that has zebra standing in the field
Img XE	A group of zebras grazing in a field
Cap XE	Two zebras are standing in a grassy field
Img RL	A group of zebras are grazing in a field
Cap RL	A couple of zebras and a zebra standing in a field

GT1	A baseball batter swinging a bat over home plate
GT2	A baseball player swings a bat at a game
Img XE	A baseball player holding a bat on a field
Cap XE	A baseball player swinging a bat on top of a field
Img RL	A baseball player holding a bat on a field
Cap RL	A baseball player swinging a bat at a ball

Conclusion

- Summary
  - A new framework for metric learning with continuous labels
  - Various applications including visual representation learning
  - Performance boost over existing approaches

- Future directions
  - A better distance metric for continuous and structured labels
  - A hard triplet mining technique for continuous metric learning
  - More applications of semantic nearest neighbor search
  - A new benchmark for continuous metric learning

References

[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015

[2] Beyond triplet loss: A deep quadruplet network for person re-identification, CVPR 2017

[3] Learning to compare image patches via convolutional neural networks, CVPR 2015

[4] Learning a similarity metric discriminatively with application to face verification, CVPR 2005

[5] Improved deep metric learning with multi-class N-pair loss objective, NeurIPS 2016

[6] No fuss distance metric learning using proxies, ICCV 2017

[7] Deep metric learning via lifted structured feature embedding, CVPR 2016

[8] Softtriple loss: Deep metric learning without triplet sampling, ICCV 2019

[9] Pose embeddings: A deep architecture for learning to match human poses, arXiv 2015

[10] Thin-slicing for pose: Learning to understand pose without explicit pose estimation, CVPR 2016

[11] Sampling matters in deep embedding learning, ICCV 2017

[12] From word embeddings to document distances, ICML 2015

[13] Self-critical sequence training for image captioning, CVPR 2017

[14] Bottom-up and top-down attention for image captioning and visual question answering, CVPR 2018