

Deep Group-shuffling Random Walk for Person Re-identification

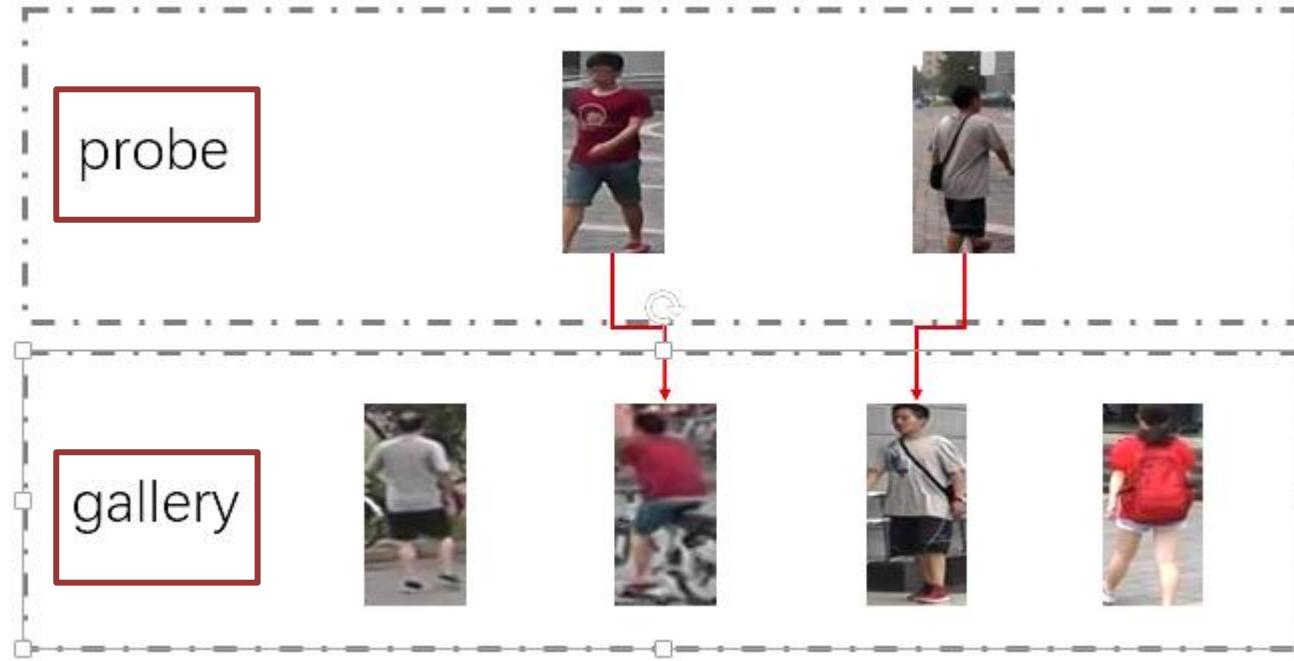
Yantao Shen, Hongsheng Li, Tong Xiao, Shuai Yi, Dapeng Chen, Xiaogang Wang
CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
SenseTime Research

Reference: http://openaccess.thecvf.com/content_cvpr_2018/papers/Shen_Deep_Group-Shuffling_Random_CVPR_2018_paper.pdf

Intelligent Information Fusion Research Group

行人重识别(Person Re-identification)

是一个图像检索问题，给定一组图片集(**probe**)，对于probe中的每张图片，从候选图片集 (**gallery**) 中找到最可能属于同一个行人的图片。



一般都是通过挖掘**P到G**的内部联系，从而来增强由P到G的映射关系
(**时空模型，图像风格转换SPGAN...**)

但**G到G**之间是否也存在一定的关系，从而能帮助我们更好地增强P到G的映射关系呢？

答案是肯定的，理由如下：

中介

P



P2G

正面和侧影关系分数还是比较高



G

正面和背影
由于角度问题（比较难抽到类似的特征），
之间的关系分数会比较低

P2G

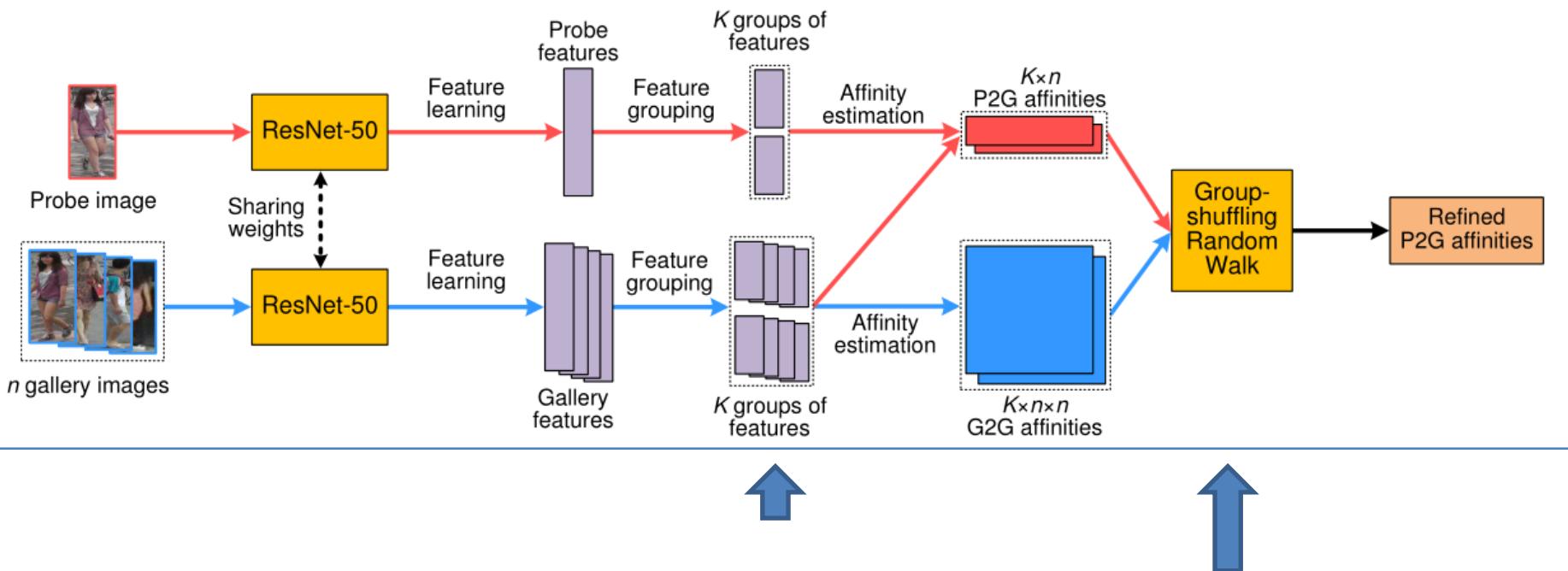


G

G2G

P2G：代表P到G的关系
G2G：代表G到G的关系

DGRW (Deep Group-shuffling Random Walk):

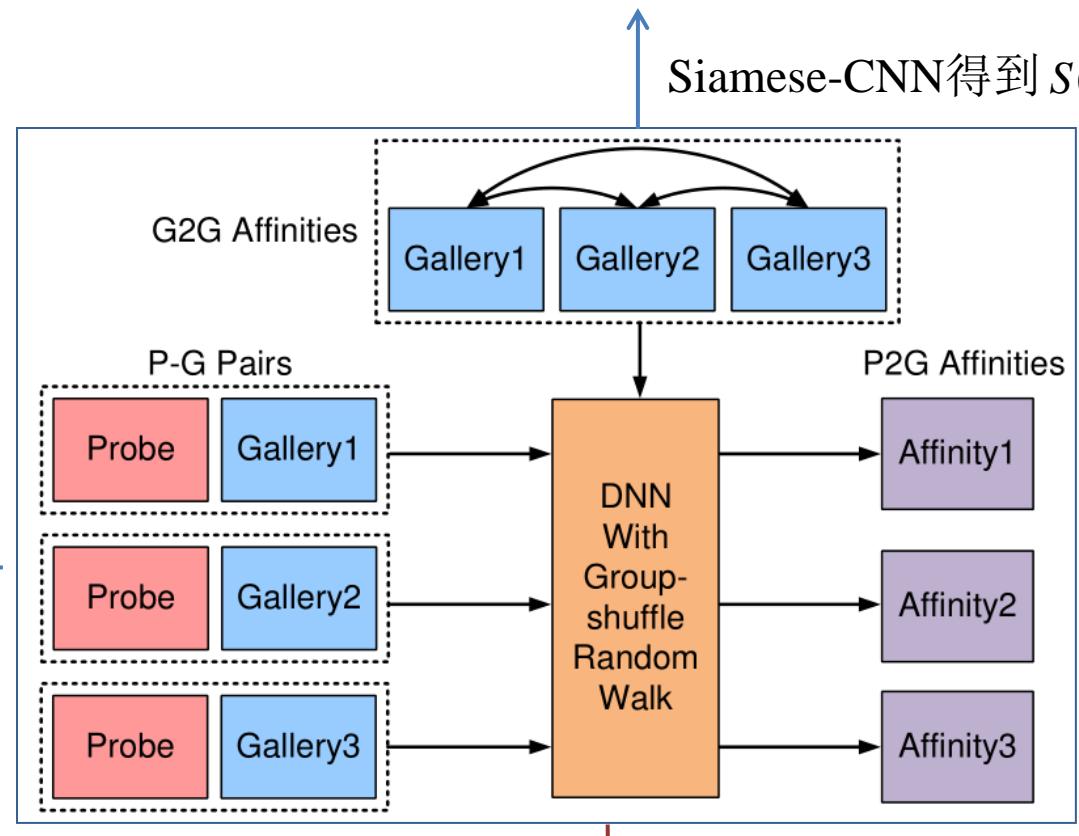


Spot:

- ① utilize G2G affinity in both train and test phrase
- ② use the G2G affinity to refine the P2G affinity
- ③ Feature grouping and **group shuffling**.

具体实现：

$$W(i, j) = \frac{\exp(S(i, j))}{\sum_{j \neq i} \exp(S(i, j))} \quad (\text{n*n 的矩阵})$$



$$y^{(1)}(i) = W(i, 1) \cdot y^{(0)}(1) + \cdots + W(i, n) \cdot y^{(0)}(n)$$

$$y^{(t+1)} = W y^{(t)}$$

具体实现：

为了控制每次更新值与原始值得距离
(不要越来越远) ,
每次更新的时候加 λ 来调节 ($\lambda \in [0,1]$) :

$$y^{(1)} = \lambda W y^{(0)} + (1 - \lambda) y^{(0)}$$

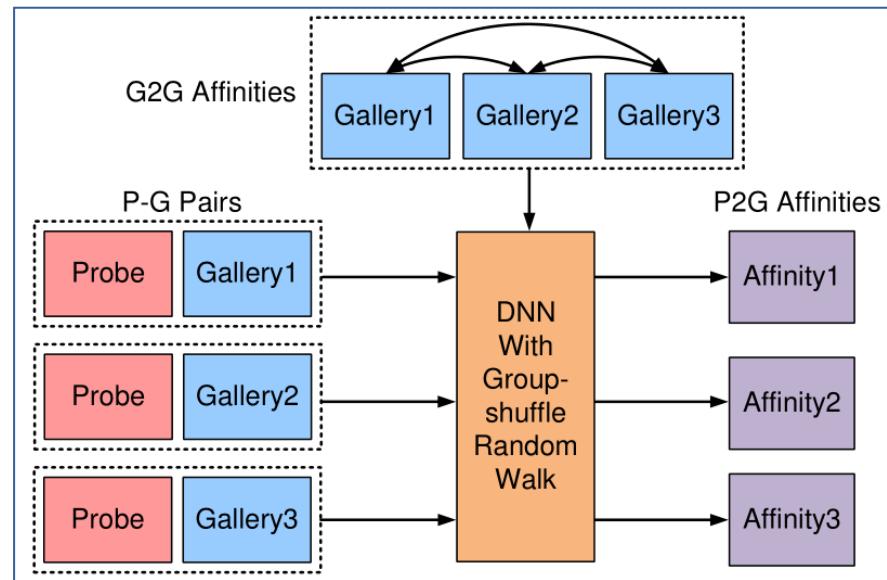
$$y^{(2)} = \lambda W y^{(1)} + (1 - \lambda) y^{(0)}$$

.....

$$y^{(t+1)} = \lambda W \underline{y^{(t)}} + (1 - \lambda) y^{(0)}$$

↓ 展开

$$y^{(t+1)} = (\lambda W)^{t+1} y^{(0)} + (1 - \lambda) \sum_{i=0}^t (\lambda W)^i y^{(0)}$$

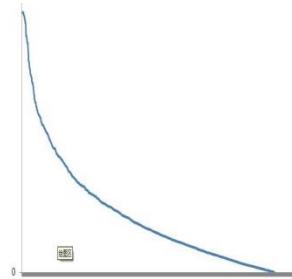


具体实现:

$$y^{(t+1)} = (\lambda W)^{t+1} y^{(0)} + (1 - \lambda) \sum_{i=0}^t (\lambda W)^i y^{(0)}$$

$$\lambda \in [0, 1]$$

$$\lambda \in [0, 1]$$



0

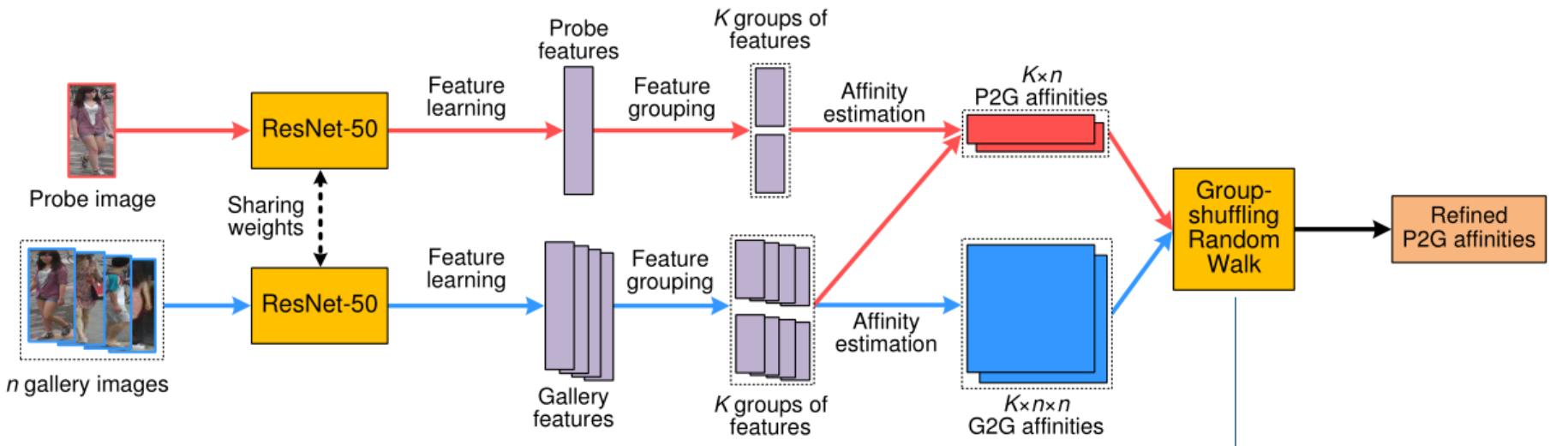
几何级数: $\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n \quad |x| < 1$



$$\lim_{t \rightarrow \infty} \sum_{i=0}^t (\lambda W)^i = (I - \lambda W)^{-1}$$

$$y^{(\infty)} = (1 - \lambda)(I - \lambda W)^{-1} y^{(0)}$$

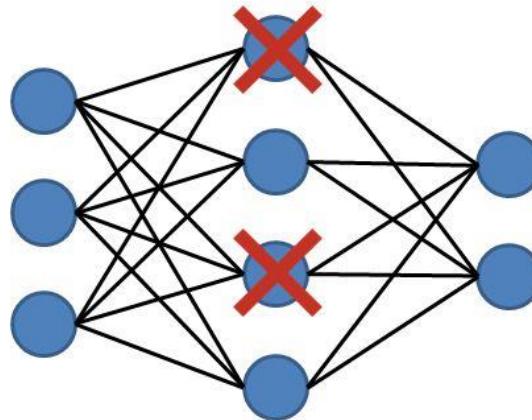
DGRW (Deep Group-shuffling Random Walk):



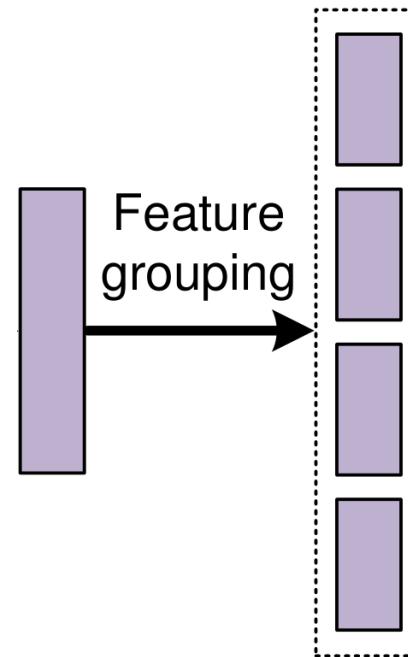
$$y^{(\infty)} = (1 - \lambda)(I - \lambda W)^{-1}y^{(0)}$$

Trick: shuffling

在神经元训练的时候，有可能上半身的图像特征比较突出，从而使得神经元训练几乎只是激活了取上半身图像特征的神经元；而采集下半身图片特征的神经元却一直没发挥作用。

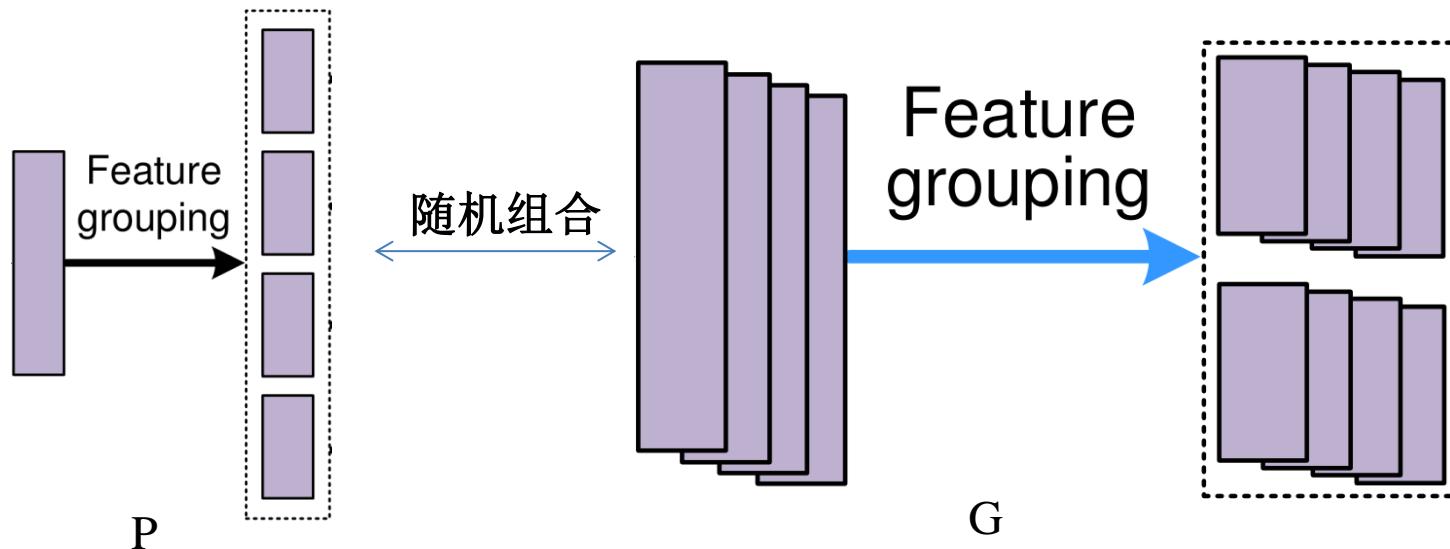


配合 random walk



一般采用 **dropout**

Trick:



如果分为两组，既 $K=2$ ，那么将产生4对P2G更新：

$$\{y_1^{(0)}, W_1\}, \{y_1^{(0)}, W_2\}, \{y_2^{(0)}, W_1\}, \{y_2^{(0)}, W_2\}$$

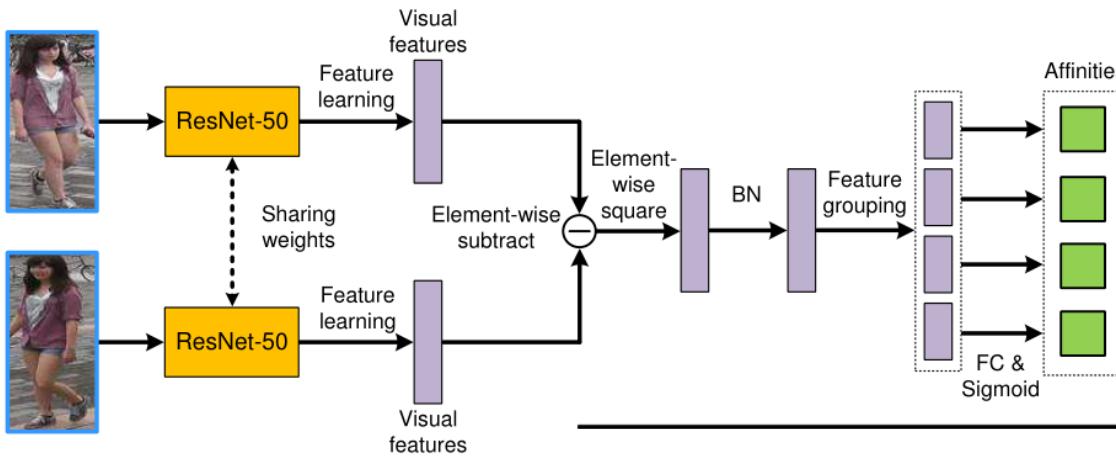
并最终得到四组更新的P2G值：

$$y_{11}^{(\infty)}, y_{12}^{(\infty)}, y_{21}^{(\infty)}, y_{22}^{(\infty)}$$

Q&A:

- 1.怎么切特征的？深度。
- 2.得到的几组最终值需要合起来？取平均值。

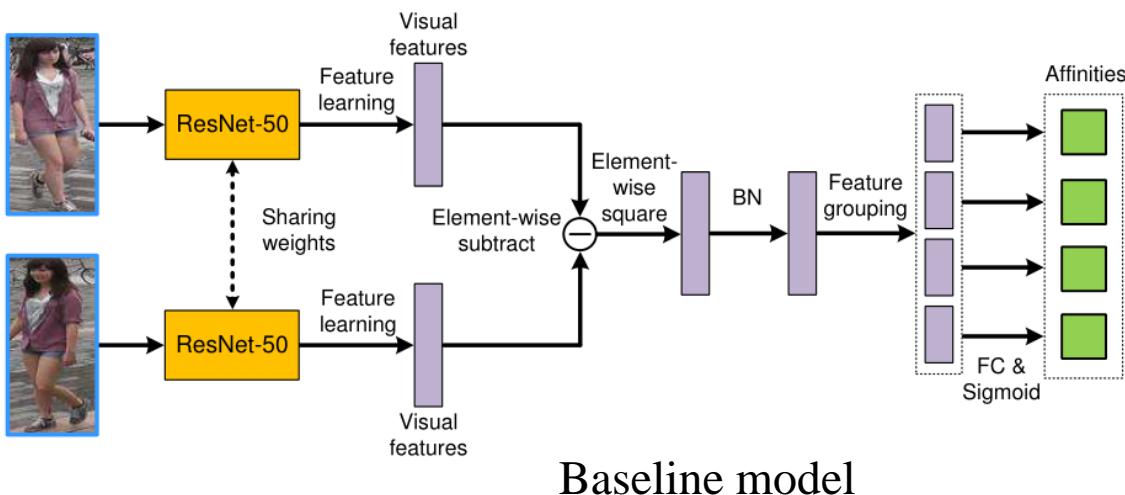
The effectiveness of each component:



Baseline model

Model	Market-1501		CUHK03	
	mAP	top-1	mAP	top-1
baseline	76.4	91.2	88.9	91.1
baseline+triplet [12]	68.3	84.5	-	-
baseline+dropout	77.6	91.3	89.1	91.2
baseline+group	77.7	91.1	91.6	93.0
baseline+group+dropout	78.1	91.3	91.3	93.3
baseline+k-reciprocal [44]	78.5	91.5	89.9	92.2
baseline+RW w/o train	79.2	91.5	90.2	92.3
baseline+random walk	81.4	91.4	91.5	92.4

Ablation studies:



Baseline model

Components			Market-1501 [42]				CUHK03 [17]				DukeMTMC [26]			
#groups	RW	shuffle	mAP	top-1	top-5	top-10	mAP	top-1	top-5	top-10	mAP	top-1	top-5	top-10
1	x	x	76.4	91.2	97.1	98.2	88.9	91.1	97.6	98.7	61.8	78.8	88.5	91.0
2	x	x	77.5	91.1	97.1	98.2	90.0	92.2	98.2	98.9	62.6	79.2	88.7	91.0
4	x	x	77.7	91.1	96.9	97.9	91.6	93.0	98.8	99.3	62.7	78.9	88.7	91.2
1	✓	x	81.4	91.4	96.8	98.2	91.5	92.4	97.4	98.8	65.2	79.2	88.8	91.1
2	✓	x	81.4	91.5	97.0	98.0	91.4	92.3	97.0	98.5	65.2	79.0	88.4	91.1
4	✓	x	81.6	91.5	97.2	98.3	92.9	93.8	97.3	98.2	65.4	79.7	88.9	91.4
2	✓	✓	82.0	91.8	96.9	98.0	93.1	93.9	98.2	99.0	65.4	78.6	88.1	90.8
4	✓	✓	82.5	92.7	96.9	98.1	94.0	94.9	98.7	99.3	66.4	80.7	88.5	90.8

Comparison with state-of-the-art methods:

Methods	Reference	Market-1501 [42]			
		mAP	top-1	top-5	top-10
OIM Loss [34]	CVPR 2017	60.9	82.1	-	-
CADL [20]	CVPR 2017	47.1	73.8	-	-
P2S [47]	CVPR 2017	44.3	70.7	-	-
MSCAN [15]	CVPR 2017	53.1	76.3	-	-
SSM [3]	CVPR 2017	68.8	82.2	-	-
DCA [16]	CVPR 2017	57.5	80.3	-	-
SpindleNet [40]	CVPR 2017	-	76.9	91.5	94.6
k-reciprocal [44]	CVPR 2017	63.6	77.1	-	-
VI+LSRO [43]	ICCV 2017	66.1	84.0	-	-
OL-MANS [46]	ICCV 2017	-	60.7	-	-
PDC [29]	ICCV 2017	63.4	84.1	92.7	94.9
PA [41]	ICCV 2017	63.4	81.0	92.0	94.7
SVDNet [30]	ICCV 2017	62.1	82.3	92.3	95.2
JLML [18]	IJCAI 2017	65.5	85.1	-	-
Proposed		82.5	92.7	96.9	98.1

Comparison with state-of-the-art methods:

Methods	Reference	CUHK03 [17]			
		mAP	top-1	top-5	top-10
OIM Loss [34]	CVPR 2017	72.5	77.5	-	-
MSCAN [15]	CVPR 2017	-	74.2	94.3	97.5
DCA [16]	CVPR 2017	-	74.2	94.3	97.5
SSM [3]	CVPR 2017	-	76.6	94.6	98.0
SpindleNet [40]	CVPR 2017	-	88.5	97.8	98.6
k-reciprocal [44]	CVPR 2017	67.6	61.6	-	-
Quadruplet [7]	CVPR 2017	-	75.5	95.2	99.2
OL-MANS [46]	ICCV 2017	-	61.7	88.4	95.2
PA [41]	ICCV 2017	-	85.4	97.6	99.4
SVDNet [30]	ICCV 2017	84.8	81.8	95.2	97.2
VI+LSRO [43]	ICCV 2017	87.4	84.6	97.6	98.9
PDC [29]	ICCV 2017	-	88.7	98.6	99.6
MuDeep [25]	ICCV 2017	-	76.3	96.0	98.4
JLML [18]	IJCAI 2017	-	83.2	98.0	99.4
Proposed		94.0	94.9	98.7	99.3

Methods	Reference	DukeMTMC [26]			
		mAP	top-1	top-5	top-10
BoW+KISSME [42]	ICCV 2015	12.2	25.1	-	-
LOMO+XQDA [19]	CVPR 2015	17.0	30.8	-	-
OIM Loss [35]	CVPR 2015	47.4	68.1	-	-
ACRN [27]	CVPR 2017	52.0	72.6	84.8	88.9
OIM Loss [34]	CVPR 2017	47.4	68.1	-	-
Basel+LSRO [43]	ICCV 2017	47.1	67.7	-	-
SVDNet [30]	ICCV 2017	56.8	76.7	86.4	89.9
Proposed		66.4	80.7	88.5	90.8

