Convolutional Feature Transfer via Camera-Specific Discriminative Pooling for Person Re-Identification

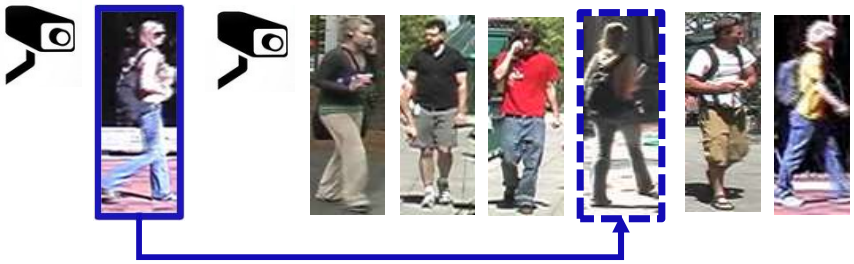
Tetsu Matsukawa Einoshin Suzuki

Kyushu University, JAPAN

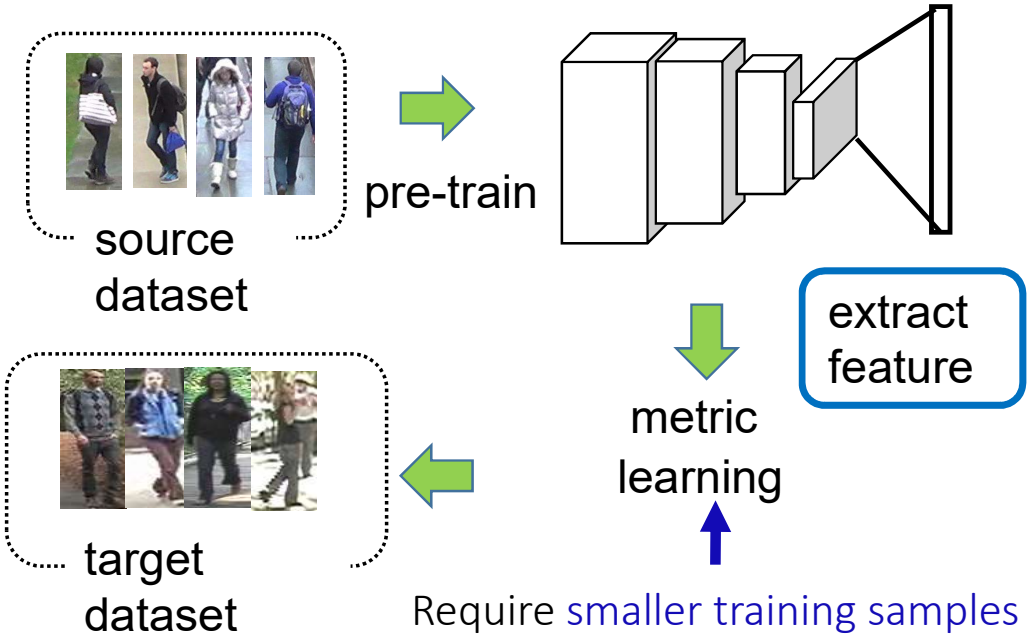
Background

Practical issue

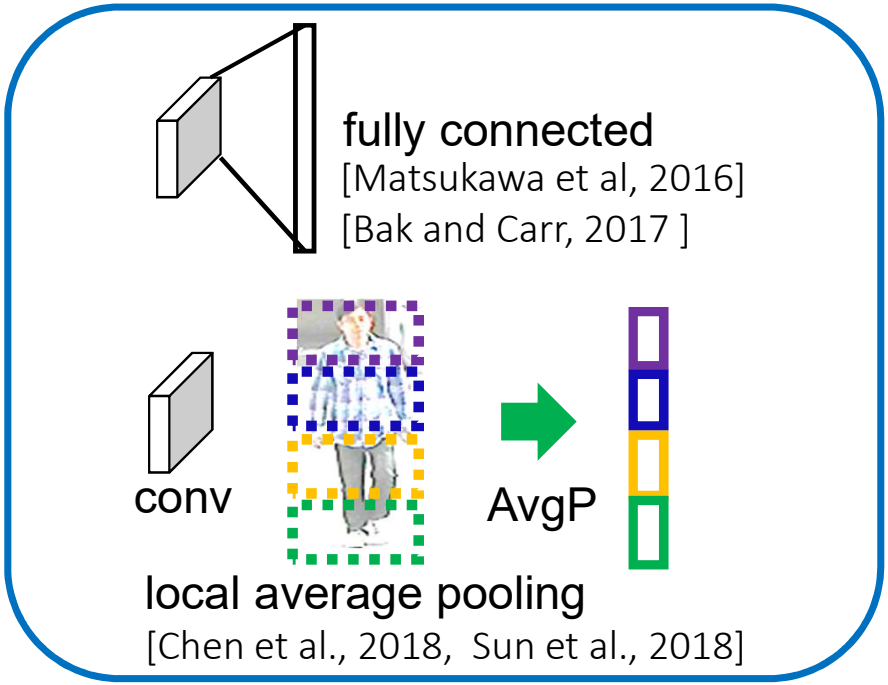
- Lack of training person IDs
- Require GPU for fine-tuning



CNN-feature transfer

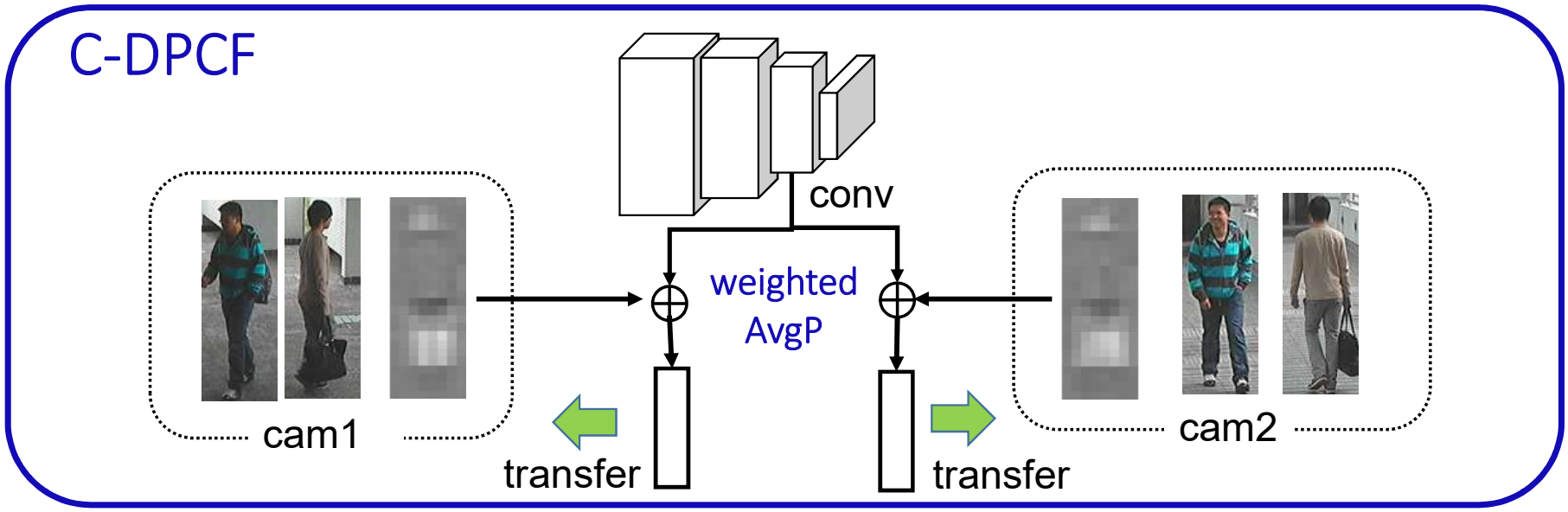


Require smaller training samples

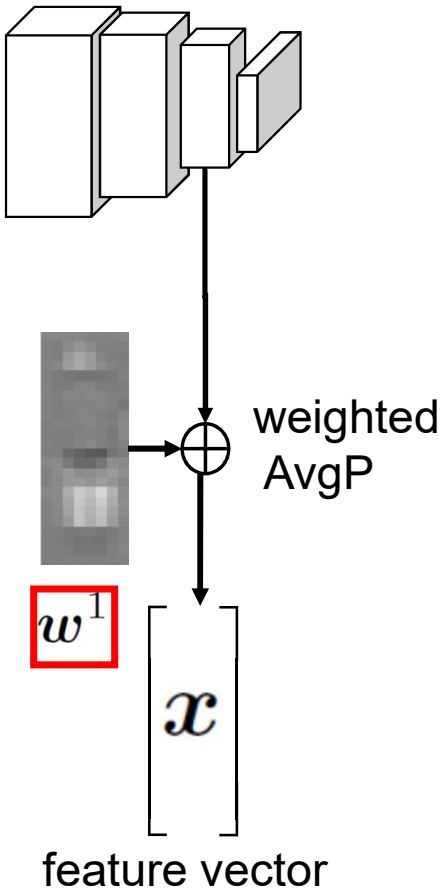


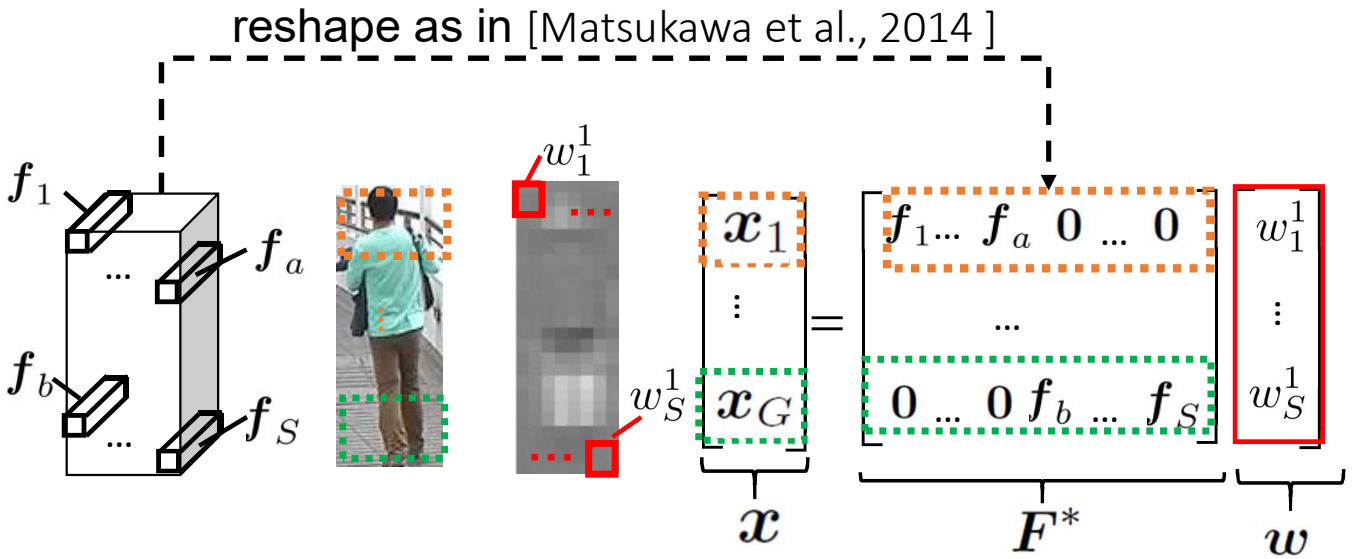
Approach

Existing features are less transferable to different camera/datasets due to spatial bias

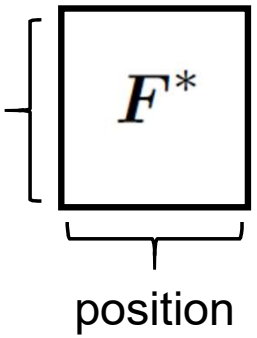


Weighted Local AvgP



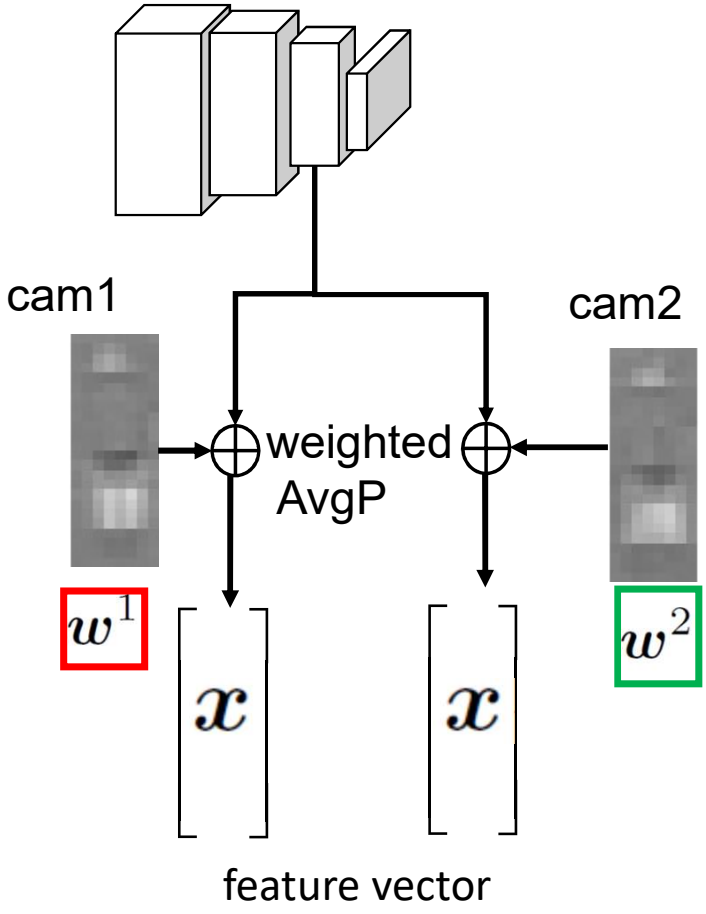


feature channel
(high dimensional)
x
pooling area



We propose
an efficient training
algorithm

Camera-specific pooling



Zero-padding to feature matrix

cam1

$$x = \left[\begin{array}{c|c|c} F^* & 0_{D\times S} & \begin{matrix} w^1 \\ w^2 \end{matrix} \end{array}\right]$$

F w

cam2

$$x = \left[\begin{array}{c|c|c} 0_{D\times S} & F^* & \begin{matrix} w^1 \\ w^2 \end{matrix} \end{array}\right]$$

Weight map learning: Problem formulation

- Given a training data $\{\boldsymbol{F}_i, p_i, c_i\}_{i=1}^{N}$

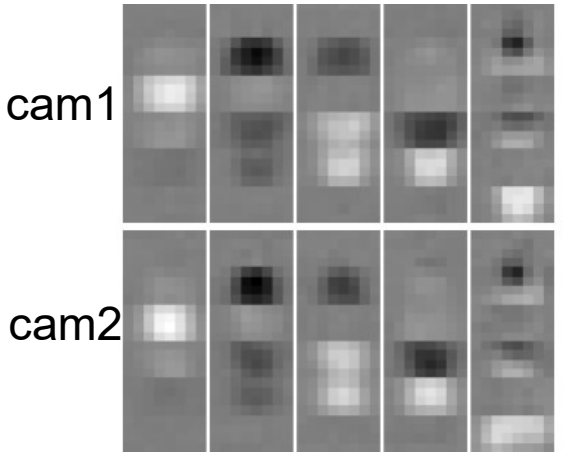
  Person ID, Cam ID

- Optimize sum of distances of K-weight map pairs

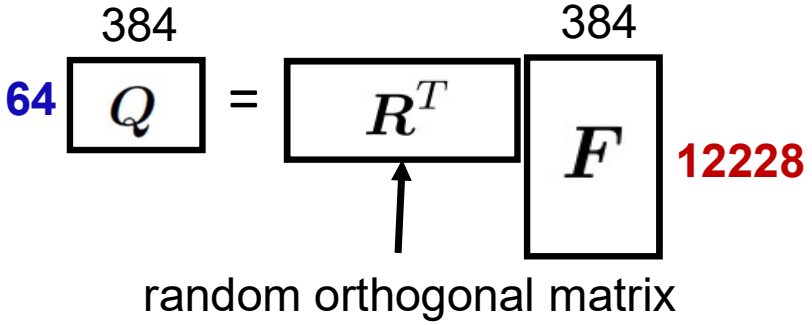
$$\delta^2_{\boldsymbol{W}}(i,j) = \sum_{k=1}^{K} \delta^2_{\boldsymbol{w}_k}(i,j)$$

  - Random projection distance

$$\begin{aligned}\delta^2_{\boldsymbol{w}_k}(i,j) &= \|\boldsymbol{R}^T\boldsymbol{x}_{k,i} - \boldsymbol{R}^T\boldsymbol{x}_{k,j}\|_2^2 \\ &= \|\boldsymbol{R}^T\boldsymbol{F}_i\boldsymbol{w}_k - \boldsymbol{R}^T\boldsymbol{F}_j\boldsymbol{w}_k\|_2^2 \\ &= \|\boldsymbol{Q}_i\boldsymbol{w}_k - \boldsymbol{Q}_j\boldsymbol{w}_k\|_2^2\end{aligned}$$



$$\boldsymbol{W} = \begin{bmatrix}\boldsymbol{w}_1 & \boldsymbol{w}_2 & \boldsymbol{w}_3 & \boldsymbol{w}_4 & \boldsymbol{w}_K\end{bmatrix}$$



Weight map learning: Optimization

- Maximum margin with orthogonal constraint

$$\max_{\boldsymbol{W}} \quad J(\boldsymbol{W}) = \underbrace{\mathrm{Tr}\left[\boldsymbol{W}^T \boldsymbol{\Sigma}_D \boldsymbol{W}\right]}_{\text{avg. distance of different persons}} - \underbrace{\mathrm{Tr}\left[\boldsymbol{W}^T \boldsymbol{\Sigma}_S \boldsymbol{W}\right]}_{\text{avg. distance of same person}}$$

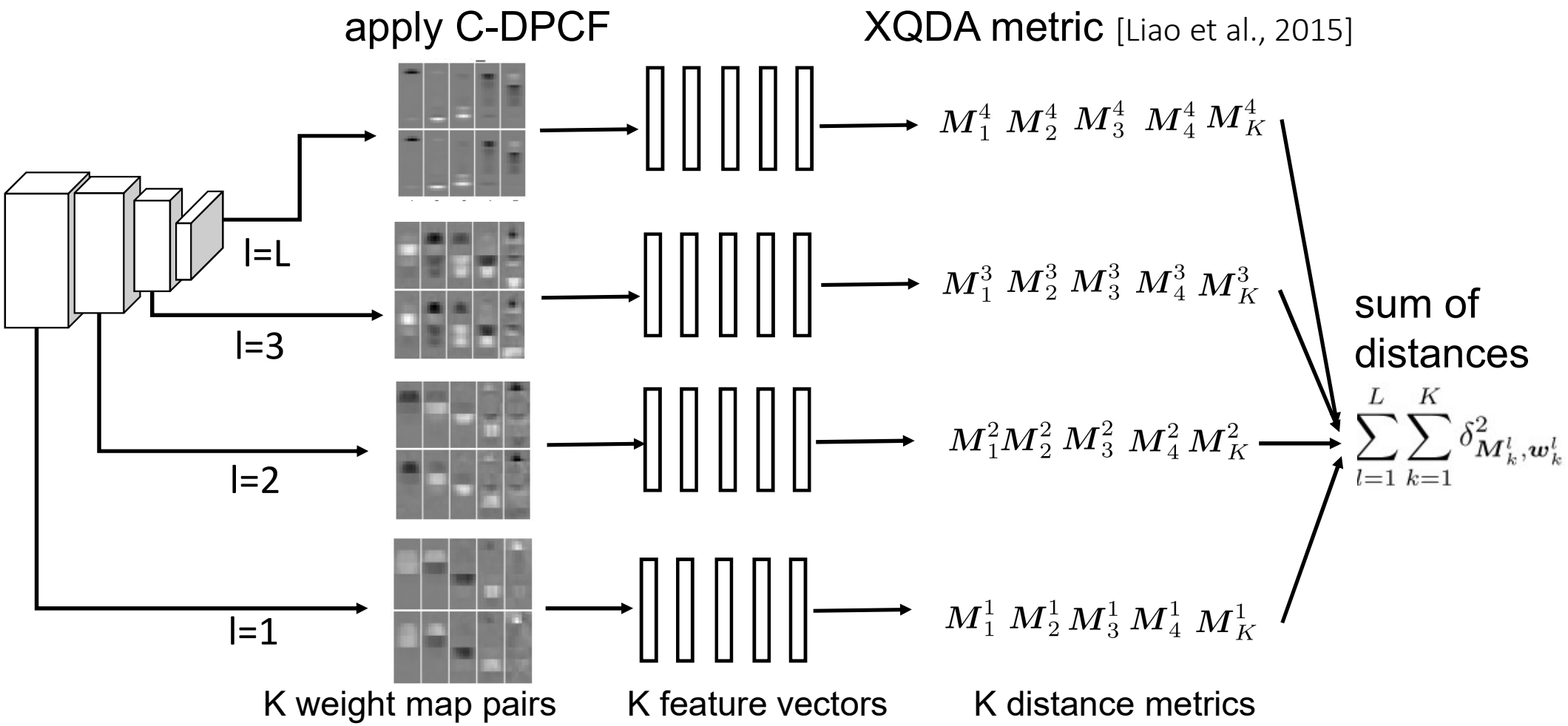
$$s.t. \quad \boldsymbol{W}^T \boldsymbol{W} = \boldsymbol{I}_K$$

where $\boldsymbol{\Sigma}_{\mathcal{D}} = \frac{1}{N_{\mathcal{D}}} \sum_{(i,j)\in\mathcal{D}} (\boldsymbol{Q}_i - \boldsymbol{Q}_j)^T (\boldsymbol{Q}_i - \boldsymbol{Q}_j)$

$\boldsymbol{\Sigma}_{\mathcal{S}} = \frac{1}{N_{\mathcal{S}}} \sum_{(i,j)\in\mathcal{S}} (\boldsymbol{Q}_i - \boldsymbol{Q}_j)^T (\boldsymbol{Q}_i - \boldsymbol{Q}_j)$

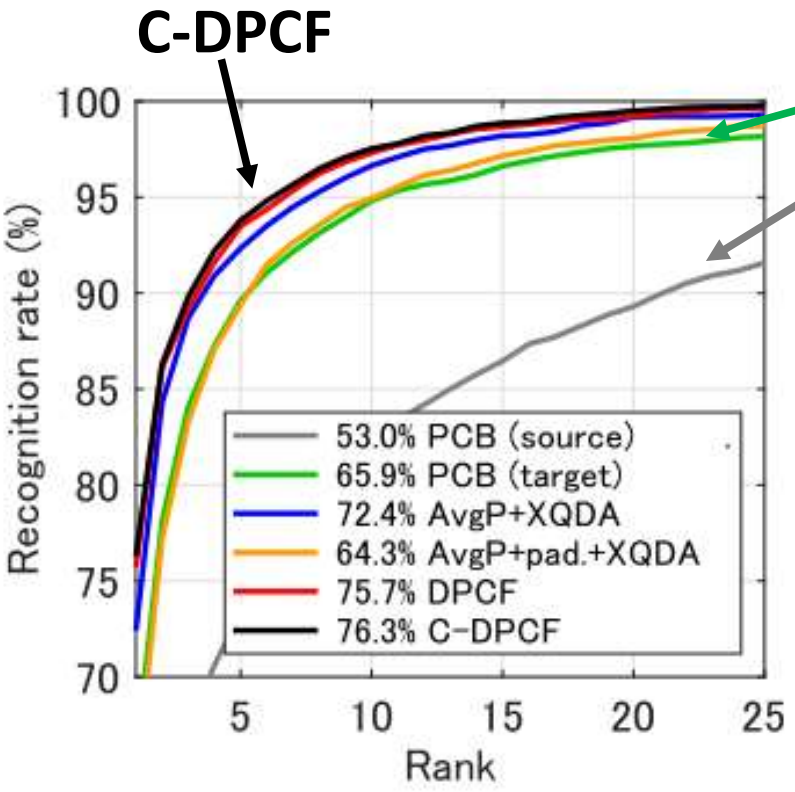
The solution is given by eigen decomposition of $\boldsymbol{\Sigma}_{\mathcal{D}} - \boldsymbol{\Sigma}_{\mathcal{S}}$

Distance for re-id



Comparison

Baseline: PCB [Sun et al., 2018] trained on (source/target) dataset



Source dataset: MSMT17
Target dataset: VIPeR
also compared on other 3 target datasets

Training time

C-DPCF	PCB(target)
42.5 sec (CPU)	312 sec (GPU)

SOTA

S: Supervised
U: Unsupervised
DG: Domain Generalization

Rank-1 rates

		Type	VIPeR	GRID	PRID	CUHK01
	CMDL[PAMI18]	S	66.4	30.9	52.0	78.2
	HGD[PAMI20]	S	52.8	28.2	-	-
CNN features	Synthesis[ECCV18]	U	43.0	-	-	54.9
CNN features	One-shot [CVPR17]	U+S	34.3	-	-	45.6
CNN features	CRAFT [PAMI18]	S	50.3	22.4	-	-
CNN features	C-DPCF [ours]	S	76.3	34.8	79.4	89.1
Mobilenet -V2	DIMN [CVPR19]	DG	51.2	29.3	-	-
Mobilenet -V2	DN [BMVC19]	DG	58.8	39.7	73.6	-
Mobilenet -V2	DN + ours	DG+S	73.9	42.3	84.1	-

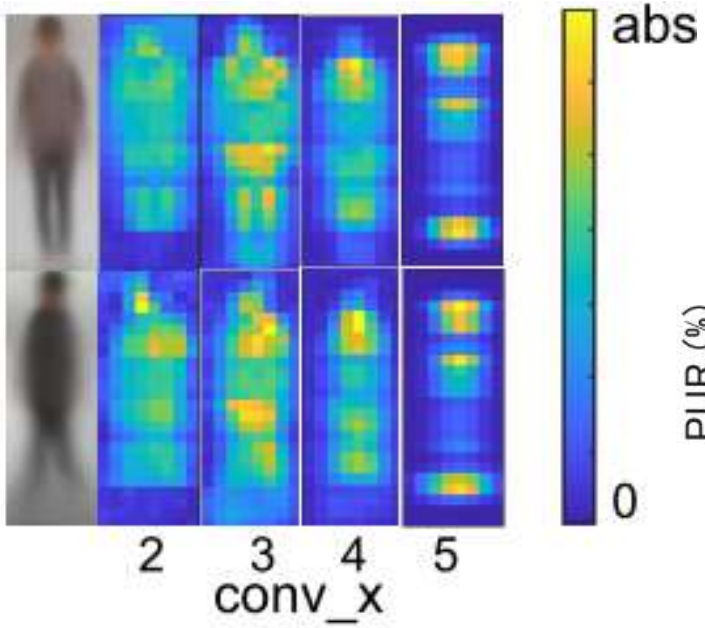
Analysis

Random projection

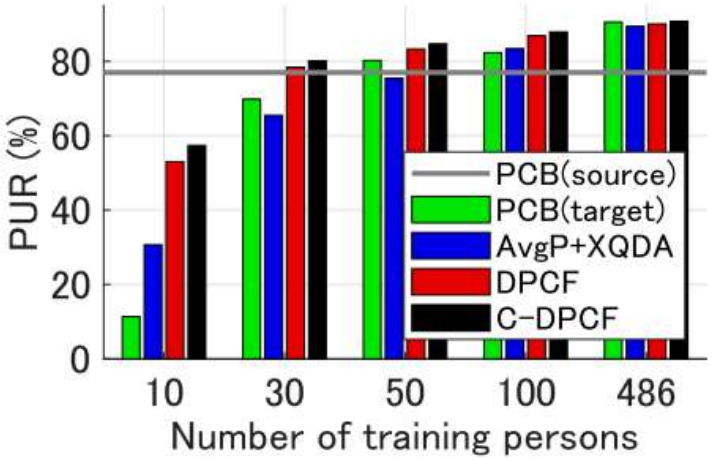
	PUR	Training time
w/o	90.5%	1684.4 sec
w/	91.3%	49.0 sec

34.4x faster

Weight maps



Accurate even with a small training data



- C-DPCF improves PCB(source) with 30 persons
- Camera-specific weight maps always outperforms common weight maps