Camera On-boarding for Person Re-identification using Hypothesis Transfer Learning
(Supplementary Material)

Sk Miraj Ahmed[1,*], Aske R Lejbølle[2,*], Rameswar Panda[2], Amit K. Roy-Chowdhury[1]
[1] University of California, Riverside, [2] Aalborg University, Denmark, [3] IBM Research AI, Cambridge
{sahme047@, alejboel@, rpand002@, amitrc@ece}.ucr.edu

Table 1: Supplementary Material Overview.

[*] Joint first authors

1. Dataset Descriptions

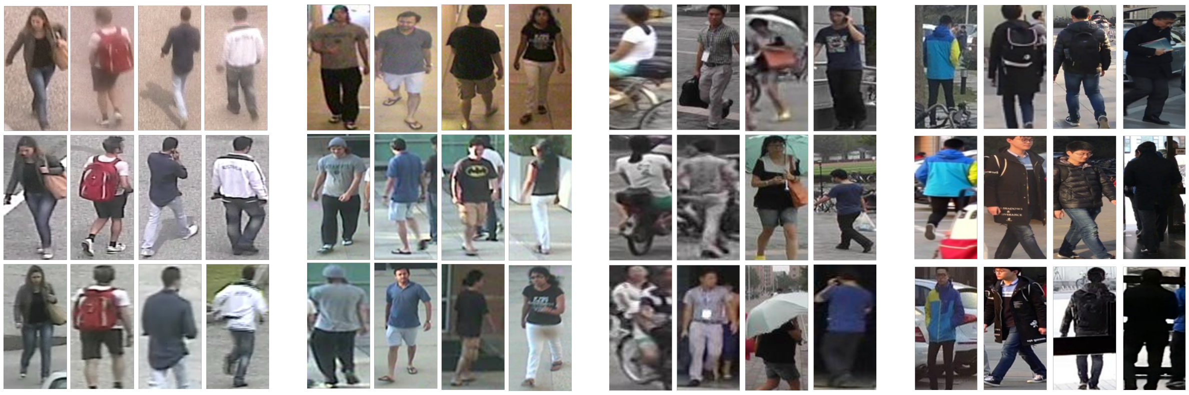
This section contains detailed descriptions of the datasets used in our experiemnts (see Figure 1 for sample images).

WARD [4] was collected from three outdoor cameras. The dataset contains 4,786 images of 70 different persons and includes variations in illumination.

RAiD [1] was collected from four cameras; two indoor and two outdoor. 6,920 images were captured of 43 different persons. However, two of these persons were only seen by two of the four cameras. As a result of having both indoor and outdoor cameras, the dataset includes large illumination and viewpoint variations.

Market1501 [9] was collected from six cameras and used a Deformable Part Model [2] to annotate images. This resulted in 32,668 images of 1,501 different persons, but also 2,793 "distractors" that are badly drawn bounding boxes. The dataset includes variations in both detection precision, resolution and viewpoint.

MSMT17 [8] is the largest person re-identification dataset to date, and contains images collected by no more than 15 cameras; 3 indoor and 12 outdoor. Data was collected over the course of four different days in a month, and Faster RCNN [7] was using for bounding box detection, resulting in 126,441 images of 4,101 different persons. Due to the diversity in data collection, this dataset contains large variations in illumination and viewpoint.



(a) WARD (b) RAiD (c) Market1501 (d) MSMT17

Figure 1: A total of 48 Sample images from the 4 datasets used in our experimentation. In each row 4 different persons are shown whereas for each column 3 different views of the same person from 3 different cameras are shown. We can see the that across cameras, the viewpoint of the same person is very diverse because of change in illumination condition or occlusion.

2. Detailed Description of the Optimization Steps

In this section we will rigorously discuss all the necessary derivations of the steps of our proposed algorithm that could not be shown in the main paper due to space constraint. We first present the notations that we will use throughout this section.
Notations:

- $\frac{1}{n_s} \sum\limits_{(i,j)\in S} x_{ij}x_{ij}^\top = \Sigma_S$
- $\frac{1}{n_d} \sum\limits_{(i,j)\in D} x_{ij}x_{ij}^\top = \Sigma_D$
- $\mathcal{C}_1 = \{M \mid \frac{1}{n_d} \sum\limits_{(i,j)\in D} (x_{ij}^\top M x_{ij}) - b \geq 0\}$
- $\mathcal{C}_2 = \{M \mid M \succeq 0\}$

- $\mathcal{C}_3 = \{\beta \mid \|\beta\|_2 \leq 1\}$
- $\Pi_{\mathcal{C}}(X) = \underset{\hat{X} \in \mathcal{C}}{\text{minimize}} \ \frac{1}{2}\|\hat{X} - X\|_F^2$
- $f(M, \beta) = \frac{1}{n_s} \sum\limits_{(i,j) \in S} x_{ij}^\top M x_{ij} + \lambda \|M - \sum_{j=1}^N \beta_j M_j\|_F^2$

The proposed optimization problem in the main paper is defined below.

$$\begin{aligned} \underset{M, \beta}{\text{minimize}} \quad & \frac{1}{n_s} \sum_{(i,j) \in S} x_{ij}^\top M x_{ij} + \lambda \|M - \sum_{j=1}^N \beta_j M_j\|_F^2 \\ \text{subject to} \quad & \frac{1}{n_d} \sum_{(i,j) \in D} (x_{ij}^\top M x_{ij}) - b \geq 0, \, M \succeq 0, \\ & \qquad\qquad\qquad\qquad \beta \geq 0, \, \|\beta\|_2 \leq 1 \end{aligned} \tag{1}$$

Step 1: Gradient w.r.t M with fixed β.

$$\begin{aligned} \nabla_M(f(M, \beta)) &= \frac{1}{n_s} \sum_{(i,j) \in S} x_{ij} x_{ij}^\top + 2\lambda(M - \sum_{j=1}^N \beta_j M_j) \\ &= \Sigma_S + 2\lambda(M - \sum_{j=1}^N \beta_j M_j) \end{aligned} \tag{2}$$

Step 2: Projection of M onto $\mathcal{C}_1$ and $\mathcal{C}_2$.

This can be done by solving a constrained optimization problem.

$$\begin{aligned} \Pi_{\mathcal{C}_1}(M) = \arg\min_{\hat{M}} \ & \frac{1}{2}\|\hat{M} - M\|_F^2 \\ \text{Subject to} \ & \frac{1}{n_d} \sum_{(i,j) \in D} (x_{ij}^\top \hat{M} x_{ij}) - b \geq 0 \end{aligned}$$

We can write the lagrangian as follows,

$$\mathcal{L}(\hat{M}, \psi) = \frac{1}{2}\|\hat{M} - M\|_F^2 + \psi(b - \frac{1}{n_d} \sum_{(i,j) \in D} x_{ij}^\top \hat{M} x_{ij}) \tag{3}$$

The KKT conditions for this problem are:

1. $$\nabla_{\hat{M}} \mathcal{L}(\hat{M}, \psi)|_{\hat{M} = \hat{M}^\star} = 0 \implies (\hat{M}^\star - M) - \frac{\psi}{n_d} \sum_{(i,j) \in D} x_{ij} x_{ij}^\top = 0 \implies (\hat{M}^\star - M) - \psi \Sigma_D = 0 \implies \hat{M}^\star = (M + \psi \Sigma_D)$$

2. $\psi^\star(b - \frac{1}{n_d} \sum\limits_{(i,j) \in D} x_{ij}^\top \hat{M}^\star x_{ij}) \geq 0$

3. $\psi^\star \geq 0$

The optimization problem is convex, so strong duality should hold. So, we put the value of $\hat{M}^\star$ from KKT condition 1 in the equation (3) to get the dual objective function as follows,

$$\begin{aligned}
g(\psi) = \mathcal{L}(\hat{M}^\star, \psi) &= \frac{1}{2}\|M + \psi\Sigma_D - M\|_F^2 + \psi\big(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top (M + \psi\Sigma_D)x_{ij}\big) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \frac{\psi^2}{n_d}\sum_{(i,j)\in D} x_{ij}^\top \Sigma_D x_{ij} \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \frac{\psi^2}{n_d}\sum_{(i,j)\in D} \text{trace}(x_{ij}^\top \Sigma_D x_{ij}) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \frac{\psi^2}{n_d}\sum_{(i,j)\in D} \text{trace}(\Sigma_D x_{ij} x_{ij}^\top) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \psi^2\text{trace}(\Sigma_D \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij} x_{ij}^\top) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \psi^2\text{trace}(\Sigma_D^\top \Sigma_D) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \psi^2\|\Sigma_D\|_F^2 \\
&= -\frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij})
\end{aligned} \tag{4}$$

To get the optimal $\psi^\star$ we have to maximize $g(\psi)$.

$$\begin{aligned}
& g'(\psi^\star) = 0 \\
\implies & -\psi^\star\|\Sigma_D\|_F^2 + (b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) = 0 \\
\implies & \psi^\star = \frac{(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij})}{\|\Sigma_D\|_F^2}
\end{aligned}$$

But also from KKT condition (3), we know $\psi \geq 0$. Combining with the last equation we get

$$\psi^\star = \max\left\{0, \frac{(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij})}{\|\Sigma_D\|_F^2}\right\} \tag{5}$$

So, putting the value of $\psi^\star$, finally we can write the projection from KKT condition 1 as,

$$\Pi_{C_1}(M) = M + \max\left\{0, \frac{(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij})}{\|\Sigma_D\|_F^2}\right\}\Sigma_D \tag{6}$$

projection onto $\mathcal{C}_2$ is standard, so we are not discussing it here.

Step 3: Gradient w.r.t β with fixed M.

$$\begin{aligned} f(M^{k+1},\beta) &= \frac{1}{n_s}\sum_{(i,j)\in S} x_{ij}^\top M^{k+1} x_{ij} + \lambda \| M^{k+1} - \sum_{j=1}^{N}\beta_j M_j\|_F^2 \\ &= K + \lambda \| M^{k+1} - \sum_{j=1}^{N}\beta_j M_j\|_F^2 \\ &= K + \lambda \text{trace}\Big((M^{k+1} - \sum_{j=1}^{N}\beta_j M_j)^\top (M^{k+1} - \sum_{j=1}^{N}\beta_j M_j)\Big) \\ &= K + \lambda \beta_i^2 \text{trace}(M_i^\top M_i) - 2\lambda\beta_i \text{trace}(M_i^\top (M^{k+1} - \sum_{j=1, j\neq i}^{N} \beta_j M_j)) \end{aligned} \tag{7}$$

K is term which is independent of β. Now differentiating equation (7) w.r.t β_i we get ,

$$\nabla_{\beta_i} f(M^{k+1},\beta) = 2\lambda\beta_i \text{trace}(M_i^\top M_i) - 2\lambda \text{trace}(M_i^\top (M^{k+1} - \sum_{j=1, j\neq i}^{N} \beta_j M_j)) = a_i \tag{8}$$

So, derivative of $f(M^{k+1},\beta)$ w.r.t β is given by,

$$\nabla_\beta f(M^{k+1},\beta) = \begin{bmatrix} a_1 & a_2 & \dots & a_N \end{bmatrix}^\top \tag{9}$$

Step 4: Projection of β onto $\mathcal{C}_3$.

$$\Pi_{\mathcal{C}_3}(\beta) = \max\left\{ 0, \frac{\beta}{\max\{1, \|\beta\|_2\}} \right\} \tag{10}$$

The intuition here is that, when the norm of β is greater than 1 then $\max\{1, \|\beta\|_2\} = \|\beta\|_2$ which implies the normalization of β. Similarly when the norm of β is lesser or equal to 1 then $\max\{1, \|\beta\|_2\} = 1$, which means keeping the β as it is since it already lies in the unit norm ball. The maximum with 0 essentially denotes the projection of any vector within the unit norm ball to the first quadrant of that ball only.

3. Proof of the Theorems

As mentioned in the paper the optimization proposed by us can be written in the same format as [6]

$$\underset{M\succeq 0}{\text{minimize}} \;\; L_T(M) + \lambda \|M - M_S\|_F^2 \tag{11}$$

where $M_S = \sum_{j=1}^{N} \beta_j M_j$ and

$$L_T(M) = \frac{1}{n_s}\sum_{(i,j)\in S} x_{ij}^\top M x_{ij} + \mu^\star \big(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij} \big) \tag{12}$$

Theorem 1. *For the convex and k-Lipschitz loss defined in (12) the average bound can be expressed as*

$$\mathbb{E}_{T\sim \mathcal{D}_{\mathcal{T}}^n}[L_{\mathcal{D}_\mathcal{T}}(M^\star)] \leq L_{\mathcal{D}_\mathcal{T}}(\widehat{M_S}) + \frac{8k^2}{\lambda n}, \tag{13}$$

where n is the number of target labeled example, $M^\star$ is the optimal metric computed from Algorithm 1, $\widehat{M_S}$ is the average of all source metrics defined as $\frac{\sum_{j=1}^{N} M_j}{N}$, $\mathbb{E}_{T\sim \mathcal{D}_{\mathcal{T}}^n}[L_{\mathcal{D}_\mathcal{T}}(M^\star)]$ is the expected loss by $M^\star$ computed over distribution $\mathcal{D}_\mathcal{T}$ and $L_{\mathcal{D}_\mathcal{T}}(\widehat{M_S})$ is the loss of average of source metrics computed over $\mathcal{D}_\mathcal{T}$.

Proof. If there is a single source metric is available for transfer , the proof has been shown in [6]. In case of multiple metric for any fixed β, we can directly replace M_S by $\sum_{j=1}^{N}\beta_j M_j$ in the **Theorem 2** in [6] to get,

$$\mathbb{E}_{T\sim\mathcal{D}_{\mathcal{T}^n}}[L_{\mathcal{D}_\mathcal{T}}(M^\star)] \leq L_{\mathcal{D}_\mathcal{T}}\big(\sum_{j=1}^{N}\beta_j M_j\big) + \frac{8k^2}{\lambda n} \tag{14}$$

which is true $\forall \beta \in \mathcal{C}_3$. Where,

$$\beta = \begin{bmatrix}\beta_1 & \beta_2 & \dots & \beta_N\end{bmatrix}^\top \in \mathbb{R}^N \tag{15}$$

Clearly without loss of generality we can write $\beta = \beta'$ where,

$$\beta' = \begin{bmatrix}\frac{1}{N} & \frac{1}{N} & \dots & \frac{1}{N}\end{bmatrix}^\top \in \mathcal{C}_3 \tag{16}$$

since, $\beta' \geq 0$ and $\|\beta'\|_2 = \frac{1}{\sqrt{N}} \leq 1$. So, plugging β' in equation (14) we get equation (13), which completes the proof. □

Theorem 2. *With probability* $(1-\delta)$*, for any metric* M *learned from Algorithm 1 we have,*

$$L_{\mathcal{D}_\mathcal{T}}(M) \leq L_T(M) + \mathcal{O}\big(\frac{1}{n}\big) + \left(\sqrt{\frac{L_T(\sum_{j=1}^{N}\beta_j M_j)}{\lambda}} + \quad \|\sum_{j=1}^{N}\beta_j M_j\|_F\right)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}, \tag{17}$$

where $L_{\mathcal{D}_\mathcal{T}}(M)$ *is the loss over the original target distribution (true risk),* $L_T(M)$ *is the loss over the existing target data (empirical risk), and* n *is the number of target samples.*

Proof. In [6], $L_T(M)$ is defined as,

$$L_T(M) = \frac{1}{n^2}\sum_{(z_i,z_j)\in T} l(M, z_i, z_j) \tag{18}$$

□

The authors in [6] have used a specific loss for analysis,

$$l(M, z_i, z_j) = [yy'((z_i - z_j)^\top M(z_i - z_j) - \gamma_{yy'})]_+ \tag{19}$$

For our case,

$$\begin{aligned}
L_T(M) &= \frac{1}{n_s}\sum_{(i,j)\in S} z_{ij}^\top M z_{ij} + \mu^\star\big(b - \frac{1}{n_d}\sum_{(i,j)\in D} z_{ij}^\top M z_{ij}\big)\\
&= \frac{1}{(n_s+n_d)}\frac{(n_s+n_d)}{n_s}\sum_{(i,j)\in S} z_{ij}^\top M z_{ij} + \frac{\mu^\star b(n_s+n_d)}{(n_s+n_d)} - \frac{\mu^\star(n_s+n_d)}{n_d}.\frac{1}{(n_s+n_d)}\sum_{(i,j)\in D} z_{ij}^\top M z_{ij}\\
&= \frac{1}{n^2}\sum_{(i,j)\in T}(\zeta_{ij}(z_i - z_j)^\top M(z_i - z_j) + \gamma)
\end{aligned} \tag{20}$$

In our case we took similar and dissimilar pairs in equal number. So, for our case $n_s = n_d = \frac{n^2}{2}$ which implies $(n_s + n_d) = n^2$. Also, $\zeta_{ij} = (1 + \frac{n_d}{n_s}) = 2$ if $(i,j) \in S$ and $\zeta_{ij} = -\mu^\star(1 + \frac{n_s}{n_d}) = -2\mu^\star$ if $(i,j) \in D$ are soft labels. Also $\gamma = \mu^\star b(n_s + n_d) = \mu^\star b n^2$. so for our case,

$$l(M, z_i, z_j) = (\zeta_{ij}(z_i - z_j)^\top M(z_i - z_j) + \gamma) \tag{21}$$

Also unlike [6] our source metric is defined as $M_S = \sum_{j=1}^{N}\beta_j M_j$. With the loss in equation (21) if we follow the exact same steps as in proof of the **Lemma 2** of [6] then we will end up with the fact that our proposed loss is (σ, m) admissible with $m = 2(1+\mu^\star)\max_{x,x'}\|x - x'\|_2^2\left(\sqrt{\frac{L_T(\sum_{j=1}^{N}\beta_j M_j)}{\lambda}} + \|\sum_{j=1}^{N}\beta_j M_j\|_F\right)$ and $\sigma = 0$.

Now putting these values of σ and m in the equation of inequality of **Theorem 4** of [6] which is,

$$L_{\mathcal{D}_\mathcal{T}}(M) \leq L_T(M) + \mathcal{O}\big(\frac{1}{n}\big) + (4\sigma + 2m + c)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}, \tag{22}$$

and ignoring c and the constant factor which are not functions of source metrics or their weights we conclude our proof.

3.1. Finding lipschitz constant for our loss

Goal: Our goal is to show the k in equation (13) has a finite value. According to the definition the loss $l(M, x, x')$ is k-lipschitz with respect to its first argument if for any pair of matrices M and M' and pair of samples x and x' we have the inequality as follows for a finite non-negative k ($0 \leq k < \infty$)

$$|l(M, x, x') - l(M', x, x')| \leq k\|M - M'\|_F \tag{23}$$

Lemma 3. *The loss defined in equation (21) is k-lipschitz with $k = 2\max(1, \mu^\star)\max_{x,x'}\|x - x'\|_2^2$*

Proof.

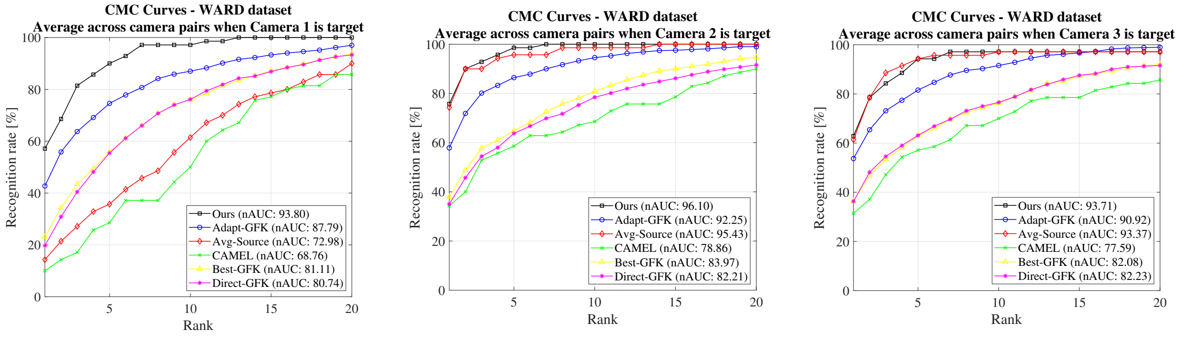
$$\begin{aligned} |l(M, x_i, x_j) - l(M', x_i, x_j)| &\leq |(\zeta_{ij}(x_i - x_j)^\top M(x_i - x_j) + \gamma) - (\zeta_{ij}(x_i - x_j)^\top M'(x_i - x_j) + \gamma)| \\ &\leq |\zeta_{ij}(x_i - x_j)^\top (M - M')(x_i - x_j)| \\ &\leq \max(|\zeta_{ij}|)\,|(x_i - x_j)^\top (M - M')(x_i - x_j)| \\ &\leq \max(2, 2\mu^\star)\,|(x_i - x_j)^\top (M - M')(x_i - x_j)| \\ &\leq 2\max(1, \mu^\star)\,\|x_i - x_j\|_2^2\|M - M'\|_F \\ &\leq 2\max(1, \mu^\star)\max_{x,x'}\|x - x'\|_2^2\|M - M'\|_F \end{aligned} \tag{24}$$

Comparing this inequality with eq. (23) we get $k = 2\max(1, \mu^\star)\max_{x,x'}\|x - x'\|_2^2$, which is clearly non-negative and finite. □

4. On-boarding a Single New Camera

This section covers the camera wise experimental results of on-boarding a single new camera (See Figure (2,4,5,6). We show for each dataset the camera wise CMC curves that are averaged to a single CMC curve in the main paper. We also showed the comparison of GFK based methods in their original setting where source data is used during target adaptation in WARD dataset (See Figure 3).

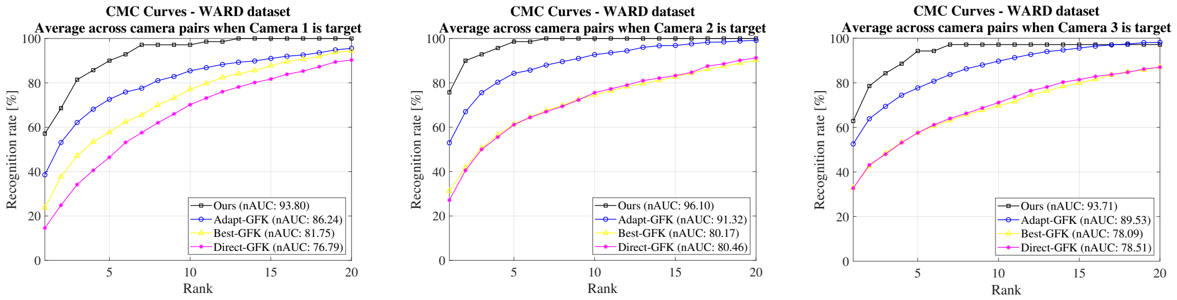
Camera wise CMC curves for WARD dataset



(a) (b) (c)

Figure 2: CMC curves for WARD[4] with 3 cameras. In this experiment each camera is shown as target while other two cameras served as source. The percentage label of new persons between the new target camera and the existing source cameras is taken to be 20% in this case. The most competitive method here is Adapt-GFK which is outperformed by our method in nAUC with margins 6%, 3.5% and 2.79% for camera 1,2 and 3 as target (plot a, b and c) respectively. In this case Adapt-GFK is calculated using the GFK matrix calculated by only using the limited labelled target data after the installation of new camera. Moreover for camera 1 as target (plot (a)) our method outperforms Adapt-GFK by a large rank-1 margin of almost 16%. Notable thing in this case is that there is only one source metric available for this dataset which is also handled by our multiple source metric transfer algorithm efficiently. Our method significantly outperform the semisupervised method CAMEL for all the plots which shows the strength of our method when a little target labeled data availabe. Also, our method outperforms Avg-Source for all the plots which is a proof of implication of Theorem 1.

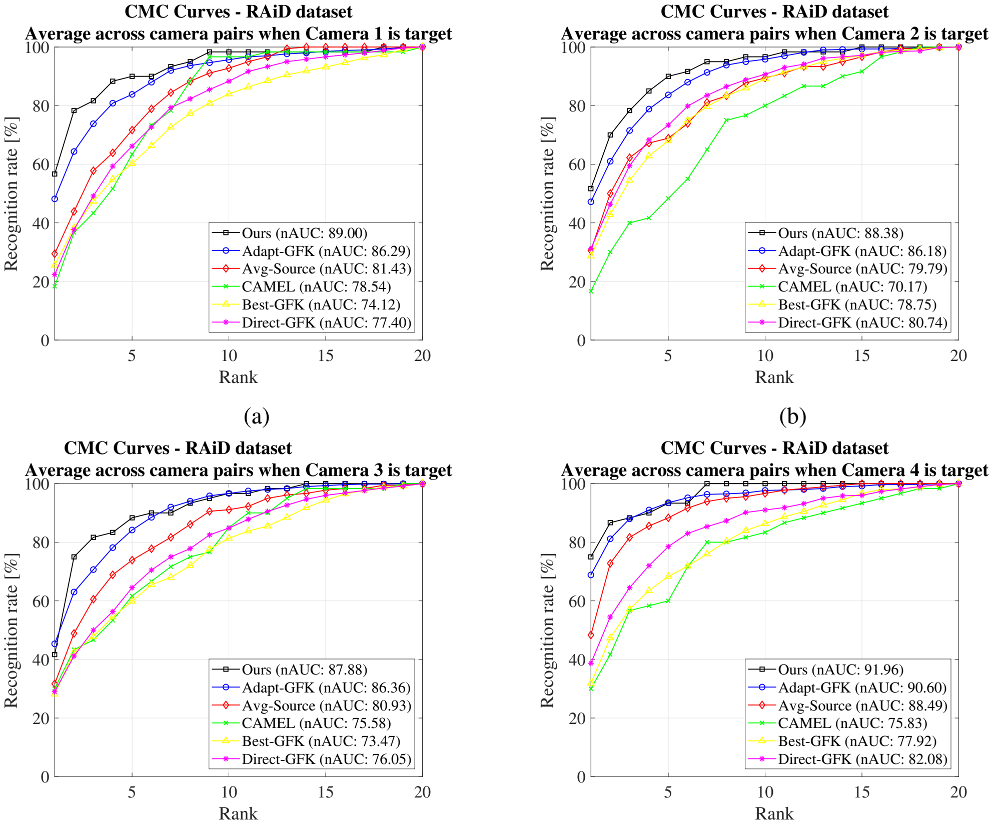
Camera wise CMC curves for WARD dataset
(GFK computed for other relevant methods using old source data and new target data)



(a) (b) (c)

Figure 3: The setting in this case is exactly same as the setting of Figure 2. However this experiment is done only to compare our method with GFK methods in the original settings [5] where the assumption was of the availability of source data. In this case GFK is calculated using the old source data as well as new limited target data. Our method significantly outperforms all the GFK based methods in this case also. It proves that even if our method does not use source data, it still outperforms the doamin adaptation methods which uses source data.

Camera wise CMC curves for RAiD dataset

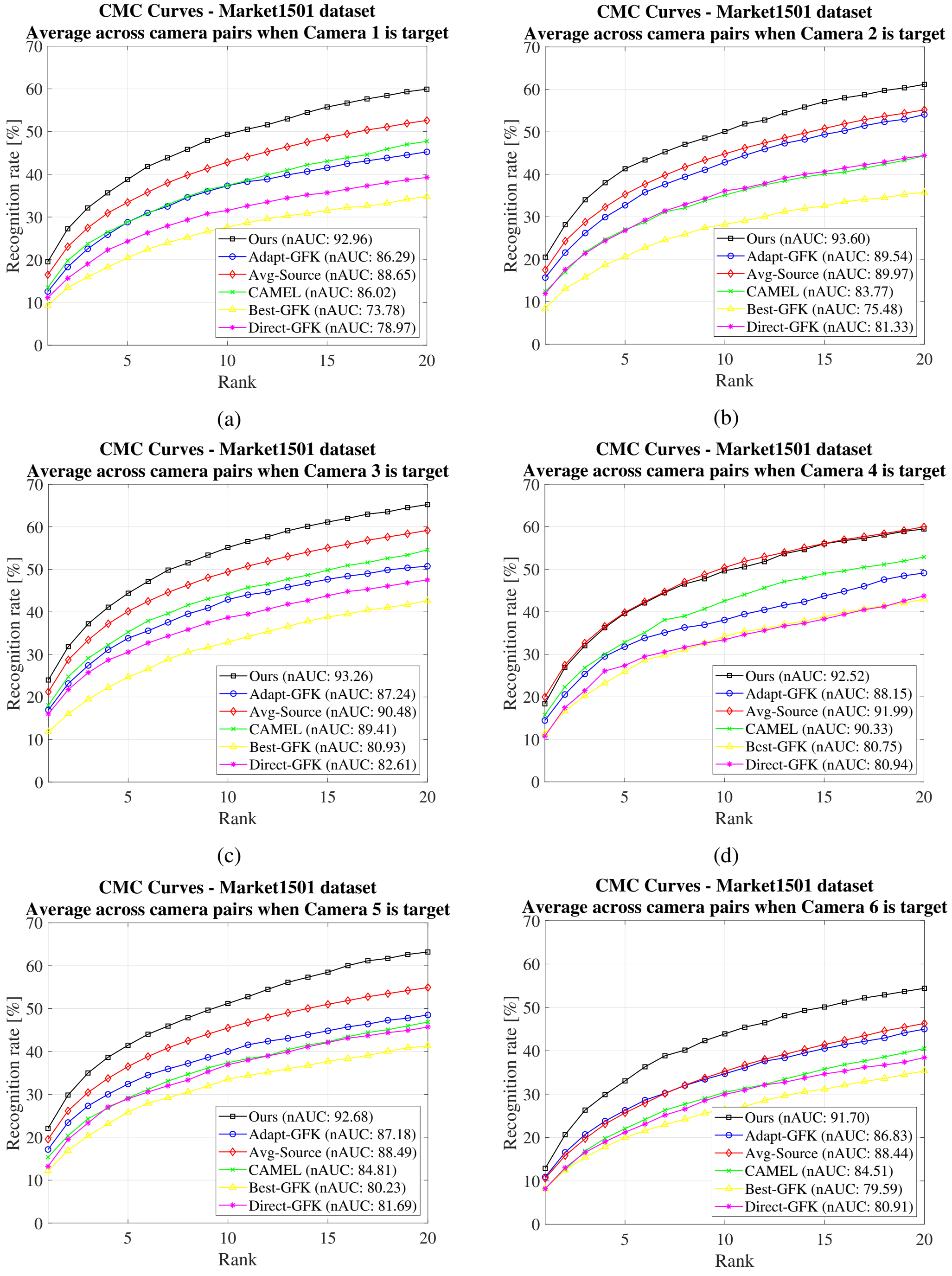


(a) (b)

(c) (d)

Figure 4: In this experiment RAiD dataset with 4 cameras [1] is used. Each of the camera has been set as target while rest of the 3 cameras with 3 pairwise metrics served as source metrics. plot (a,b,c,d) are generated from camera 1,2,3 and 4 as target target camera. The most competitive method here is Adapt-GFK which is outperformed by our method in nAUC with margins 2.71%, 2.2%, 1.52% and 1.36% for camera 1,2,3 and 4 as target respectively. Moreover for camera 1 as target (plot (a)) and camera 4 as target (plot (d)) our method outperforms Adapt-GFK by a rank-1 margin of almost 7% and 5% respectively. Also for each of the cameras our method outperforms Avg-source significantly both in rank-1 and nAUC which proves the Theorem 1. Moreover, for all the cases our method outperforms CAMEL significantly (Like in camera 4 rank-1 margin is almost 36%) which is equivalent to fully supervised learning with limited labels with no transfer from any sources.

Camera wise CMC curves for Market1501 dataset



(a)

(b)

(c)

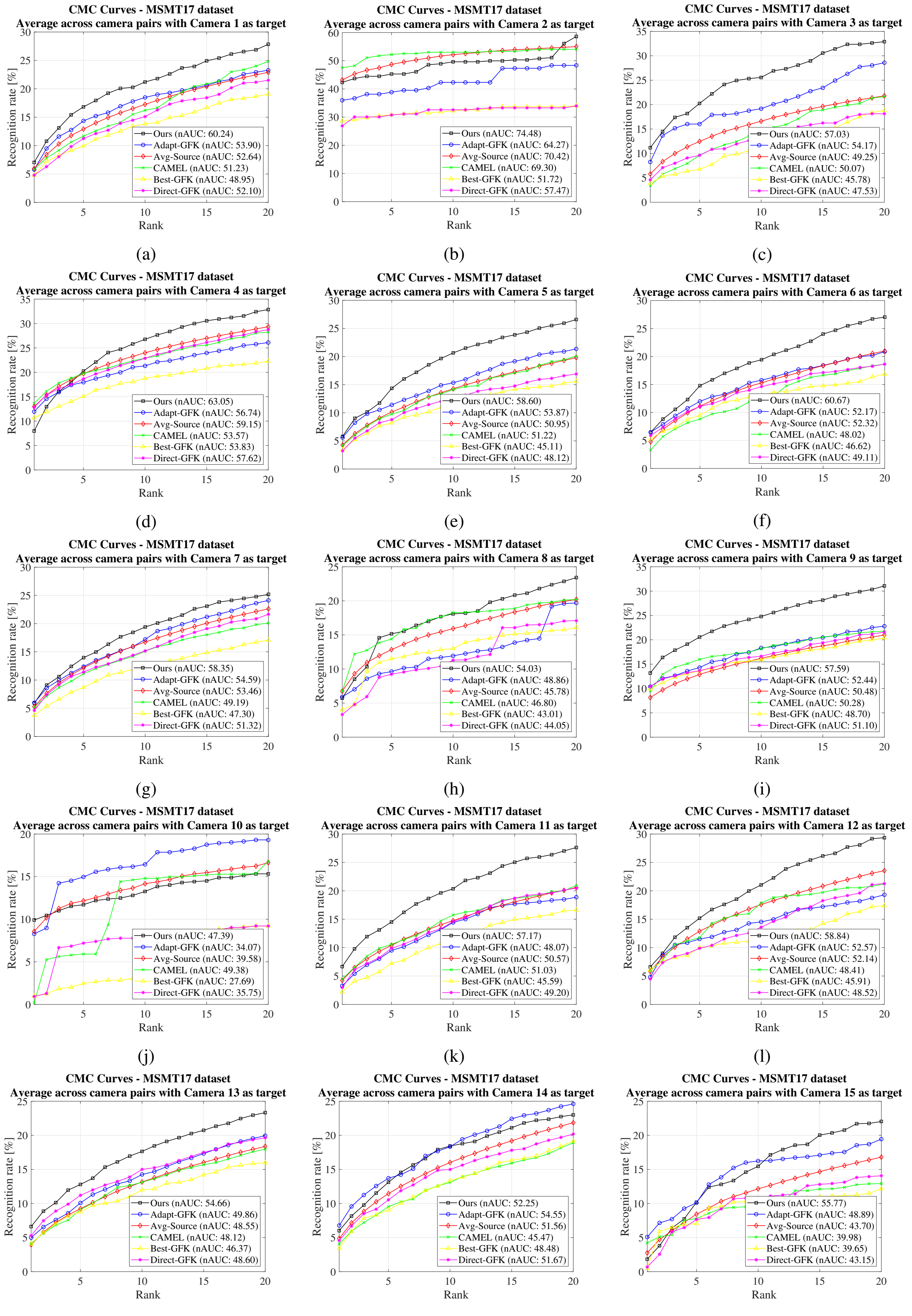
(d)

(e)

(f)

Figure 5: In this single camera insertion experiment Market1501 [9] dataset is used. In plots (a,b,c,d,e and f) cmc curves are shown for camera 1,2,3,4,5 and 6 as target respectively. Only 10% of the available data is used between each target-source pairs. Our method outperforms Adapt-GFK which was the most competitive one in case of RAiD and WARD by 6.67%,4.06%,6.02%,4.37%,5.5%,4.87% in nAUC. However, in this case we see that Adapt-GFK has lower accuracy than just the Avg-source, which we outperform in both rank-1 and nAUC for each and every camera as target. Also our method has very high accuracy both in rank-1 and nAUC than CAMEL which is equivalent to no transfer scenario. It is clear that our method gives theoretical guarantee that it would not perform worse than Avg-source case or no transfer case whereas other method has no guarantee which is depicted in this case where Adapt-GFk performed worse than just the Avg-source.

Camera wise CMC curves for MSMT dataset camera (1-15)



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

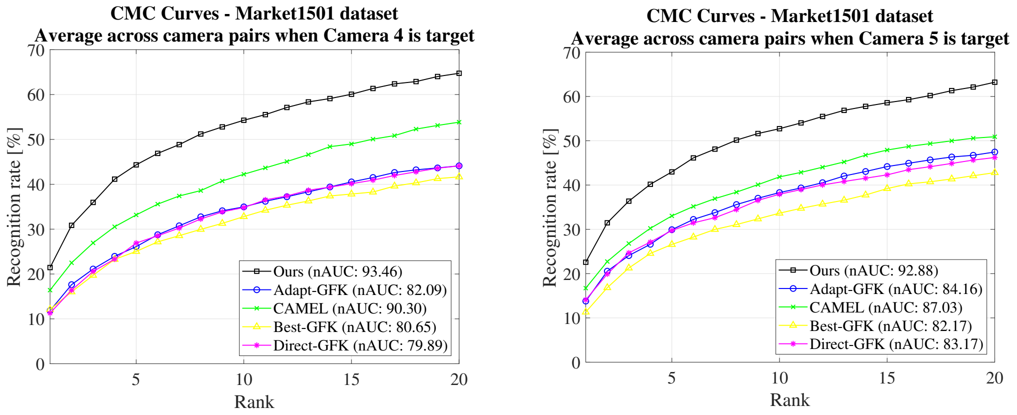
(m) (n) (o)

Figure 6: Total 15 plots from 15 cameras as target in MSMT dataset are shown. For all cameras our method outperforms other methods in nAUC. While rank-1 performances varied a lot across different cameras, our method on average performs the best as shown in the main paper. Best viewed in color.

5. On-boarding Multiple New Cameras

This section covers the camera wise experimental results of on-boarding multiple new cameras (See Figure (7,8,9). We show for each experiment the camera wise CMC curves that are averaged to a single CMC curve in the main paper.

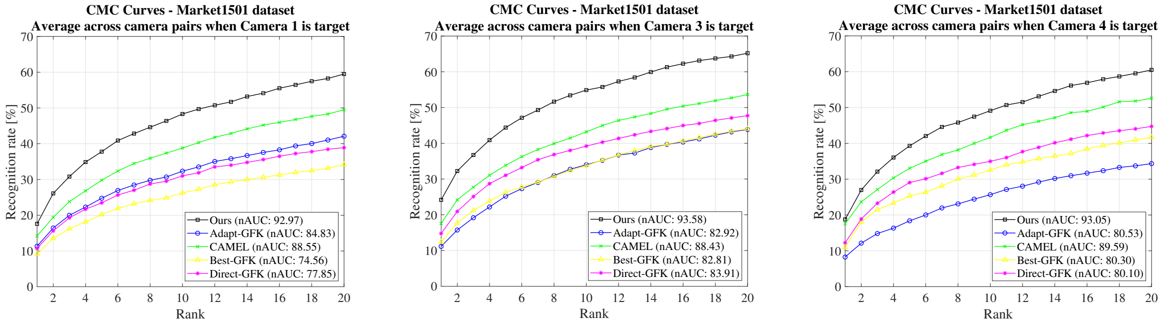
Camera wise CMC curves for Market1501 dataset: parallel addition of 2 cameras



(a) (b)

Figure 7: In this figure we used Market1501 dataset to show the effect of parallel on-boarding of multiple cameras (In this case 2 cameras). We effectively set camera 4 and 5 as target and compute 6 source metrics from the remaining cameras to transfer knowledge from. Accuracy is shown between camera 4 and camera (1,2,3,6) (plot(a)) and also between camera 5 and camera (1,2,3,6) (plot(b)) separately. We can see that our method significantly outperform other methods both in rank-1 and nAUC. This shows the effectiveness of our method for adaptation of multiple cameras in the network added in parallel.

Camera wise CMC curves for Market1501 dataset: parallel addition of 3 cameras

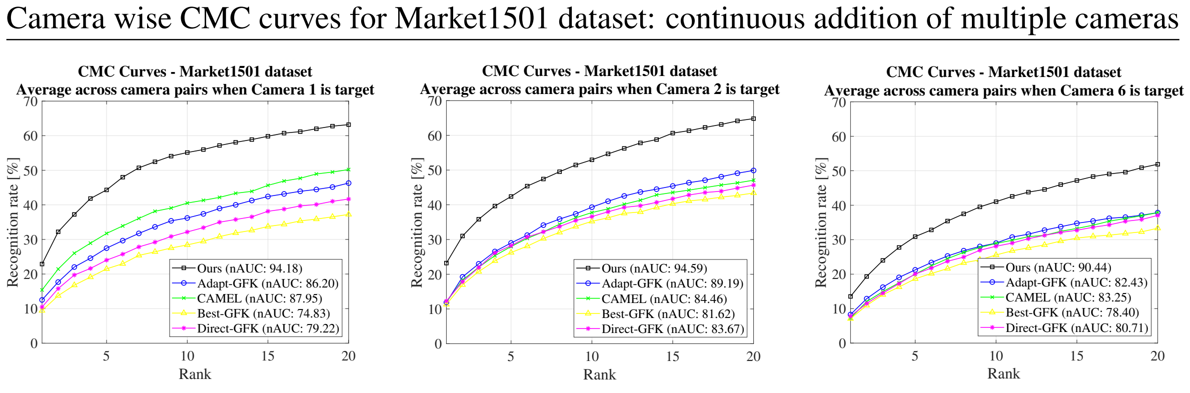


(a) (b) (c)

Figure 8: In this figure we used Market1501 dataset to show the effect of parallel on-boarding of multiple cameras (In this case 3 cameras). We effectively set camera 1,3 and 4 as target and compute 3 source metrics from the remaining cameras to transfer knowledge from. Accuracy is shown between camera 1 and camera (2,5,6) (plot(a)),camera 3 and camera (2,5,6) (plot(b)) and also between camera 4 and camera (2,5,6) (plot(c)) separately. We can see that our method significantly outperform other methods both in rank-1 and nAUC. This shows the effectiveness of our method for adaptation of multiple cameras in the network added in parallel. Best viewed in color.

6. Additional Experiments

Pairwise PCA vs Global PCA: We calculate one PCA projection matrix for the whole source network and use that in the target to project features in the main paper. Additionally to compare, we did pairwise PCA and observe that it significantly



(a) (b) (c)

Figure 9: In this figure we used Market1501 dataset to show the effect of sequential on-boarding of multiple cameras (In this case 3 cameras). Source cameras are camera 3,4 and 5 which has three source metrics between them. First camera 1 is added to the network and adapted. Accuracy for camera 1 as target is computed between camera 1 and camera (3,4,5) (plot(a)). Then camera 2 is added and adapted. For calculation of camera 2 adaptation accuracy we calculate matching score between camera 2 and camera (1,3,4,5) (plot(b)). In same fashion camera 6 is added afterwards and accuracy is calculated between camera 6 and camera (1,2,3,4,5) (plot(c)). We can see that our method significantly outperform other methods both in rank-1 and nAUC. This shows the effectiveness of our method for adaptation of multiple cameras in the network added sequentially.

lowers performance, e.g., rank-1 accuracy drops from 51.25 to 22.92 in RAiD, and drops from 62.86 to 25.71 in WARD. We believe this is due to lack of enough data across pair-wise cameras to give a reliable estimate of PCA subspaces. Combining different PCA projected metrics could be an interesting direction for future work.

Effect of $\lambda = 0$: When the existing pair-wise learned metrics are not considered (i.e., $\lambda = 0$), the rank-1 performance significantly drops from 62.86% to 27.14% on WARD. From that we conclude that a finite nonzero positive λ is a very crucial factor in order for the algorihm to work.

Initialization: Though the proposed optimization is non-convex, initialization has very little effect on the performance. We tried 2 different initializations such as identity and random positive semidefinite matrices with random weights within the first quadrant of unit-norm hypersphere, and found that both resulted minimal difference in rank-1 accuracy (RAiD: 51.25 vs 50.83 and WARD: 62.82 vs 62.38). This might be due to large convergence basin of the optimization problem.

7. Finetuning with Deep Features

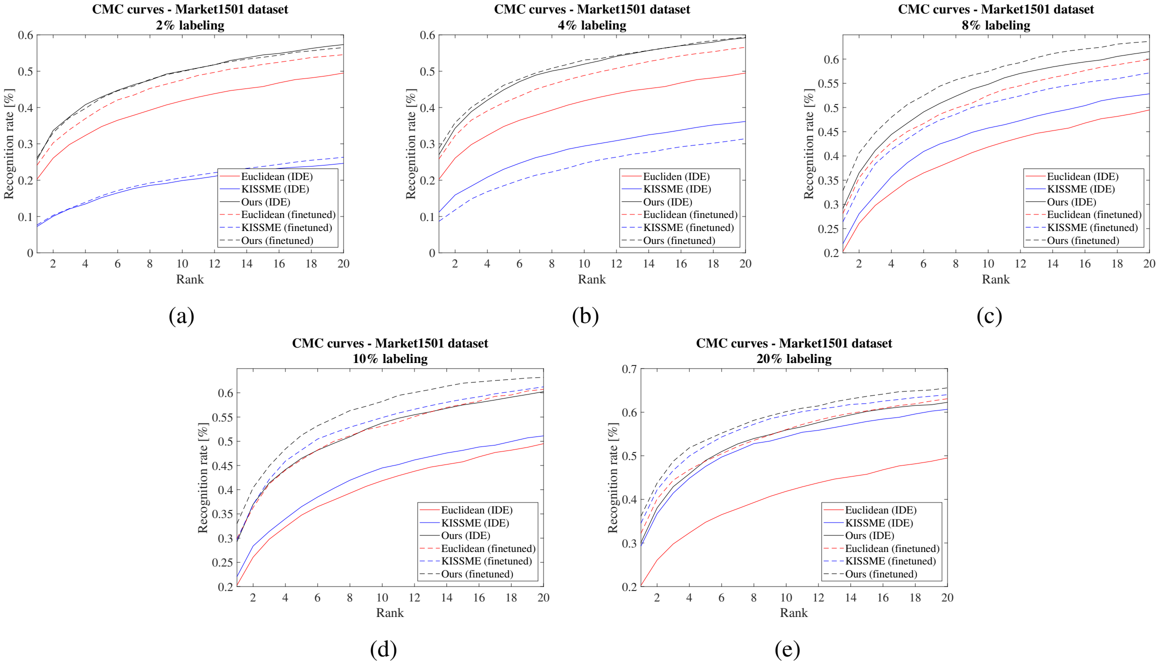
Goal: In this section our goal is to show the performance of our method (See Table 2 and Figure 10), if we have access to a deep model trained well using the source data.

Implementation details: This section covers the implementation details of finetuning deep features used in the experiments of Section 5.4 in the main paper. First, we train a ResNet model [3], pretrained on the Imagenet dataset, using the source camera data. We remove the last classification layer and add two fully connected layers; one which embeds average pooled features to size 1024 and another which works as a classifier. We use the optimized source features to train the source metrics that will later be used to calculate new target metrics. Afterwards we fine-tune the model using the new target data and use the new optimized target features along with the source metrics in optimization 1. The model is trained for 50 epochs using SGD, with a base learning rate of 0.001, which is decreased by a factor 10 after 20 and 40 epochs. We use a batch size of 32 and perform traditional data augmentation, such as cropping and flipping. We use the optimized source features to train the source metrics that will later be used to calculate new target metrics. Afterwards, we fine-tune the model for 30 epochs using the new target data. We fine-tune with a batch size of 32 and a base learning rate is 0.0001 and decreased by a factor 10 after 20 epochs. The new optimized target features are used along with the source metrics in optimization. From Figure 5 (b) of the main paper and Figure 10 in here, we observe that when we remove sixth camera in Market dataset, the accuracy of the test set between sixth and other cameras become very low as 20%, whereas in standard result for fully supervised deep model in Market dataset is around 80%. This drop in accuracy from 80 to 20% while removing 6^{th} camera in Market is due to two reasons. First, removing all the 151 person ids that appear in 6^{th} camera results in less labeled examples that leads to a less accurate deep model. Second, 6^{th} camera is the most uncorrelated with the other 5 cameras (see Fig. 7 in [9]). Figure 5(b) in main paper and Figure 10 in here clearly show that our approach works better than direct adaptation of the source model (even with finetuning) when feature distribution across source and target cameras are very different.

	Single-query		**Multi-query**	
Method	Top-1	mAP	Top-1	mAP
Euclidean	46.51	40.04	54.40	48.54
Euclidean-ft	51.51	45.52	59.66	54.36
KISSME	45.57	38.42	55.31	48.02
KISSME-ft	49.13	41.77	58.52	51.58
Ours	47.79	41.20	57.57	50.83
Ours-ft	**52.84**	**46.70**	**61.96**	**56.28**

Table 2: Results for Market1501 when we have a deep model trained using the data of 5 source cameras. We set each camera as target with 25% labeled data in it and show result of average across all the cameras. **Euclidean** denotes the accuracy of target camera if the trained source model is directly used to extract features in target test set. **KISSME** is direct metric learning between new camera and old cameras. *ft* stands for fine tuning. **Euclidean-ft** and **KISSME-ft** is same scheme that is described in the top lines of this section, except for the feature extraction policy. In these methods features are extracted using the fine tuned source model with limited target data. We can see that our proposed algorithm using features from fine-tuned model outperforms all the other accuracies.

CMC curves for Market1501 dataset with Camera 6 as target using deep learned features



(a) (b) (c)

(d) (e)

Figure 10: These plots show cmc curves for camera 6 of Market1501 dataset using the exact same scheme of Table 2 but with different percentage labels in the target. We can clearly see that our method outperforms all the other (That is direct euclidean, direct metric learning and even fine tuning with target data). When the percentage label increase then our method with non-finetuned features merges with the direct fine tuning, whereas if we use our method with the finetuned features, it exceeds all the accuracy. This shows the strength of our method even in the presence of deep learned source model.

References

[1] Abir Das, Anirban Chakraborty, and Amit K Roy-Chowdhury. Consistent re-identification in a camera network. In *ECCV*, pages 330–345. Springer, 2014. 2, 8

[2] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 12

[4] Niki Martinel and Christian Micheloni. Re-identify people in wide area camera network. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 31–36. IEEE, 2012. 2, 7

[5] Rameswar Panda, Amran Bhuiyan, Vittorio Murino, and Amit K Roy-Chowdhury. Unsupervised adaptive re-identification in open world dynamic camera networks. In *CVPR*, pages 7054–7063, 2017. 8

[6] Michaël Perrot and Amaury Habrard. A theoretical analysis of metric hypothesis transfer learning. In *ICML*, pages 1708–1717, 2015. 5, 6

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 2

[8] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *CVPR*, pages 79–88, 2018. 2

[9] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, pages 1116–1124, 2015. 2, 9, 12