Large scale pre-training for person re-identification with noisy labels

In this material, we will 1) show demo cases for scene changing for specific person in LUPerson-NL, 2) share the training details for PNL, 3) provide the detailed algorithm table for PNL, 4) compare models pre-trained on SYSU30K and LUPerson-NL 5) analyze the hyper parameters of our PNL, 6) demonstrate results of our method using a stronger backbones, 7) explore the impact of pre-training image scale, and 8) list more detailed results for our "small-scale" and "few-shot" experiments. All these experiments are under MGN settings.

1. Scene changing in LUPerson-NL

Our LUPerson-NL are driven from street view videos, Figure 1 shows a demo person in our LUPerson-NL. As we can see, our LUPerson-NL is able to cover multiple scenes, since the videos we use have lots of moving cameras and moving persons, and only traklets with $\geq$ 200 frames are selected. It is approximate close to the real person Re-ID scenario.

Figure 1. Scene changing for a specific person in LUPerson-NL.

2. Training details

During training, all images are resized to 256×128 and pass through the same augmentations verified in [12], which are Random Resized Crop, Horizontal Flip, Normalization, Random Gaussian Blur, Random Gray Scale and Random Erasing. Specifically, the images are normalized with mean and std of $[0.3452, 0.3070, 0.3114], [0.2633, 0.2500, 0.2480]$, which are calculated from all images in LUPerson-NL. We train our model on $8 \times V100$ GPUs for 90 epochs with a batch size of $1,536$. The initial learning rate is set to 0.4 with a step-wise decay by 0.1 for every 40 epochs. The optimizer is SGD with a momentum of 0.9 and a weight decay of 0.0001. We set the hyper-parameters $\tau = 0.1$ and $T = 0.8$. The momentum m for updating both the momentum encoder E_k and the prototypes is set to 0.999. We design a large queue with a size of $65,536$ to increase the occurrence of positive samples for the label guided contrastive learning. The label correction according to Equation 3 starts from the 10-th epoch. The deployment of the label guided contrastive loss $\mathcal{L}^i_{lgc}$ starts from the 15-th epoch.

3. Algorithm for PNL

Algorithm 1 shows the procedure of training PNL, we train our framework for 90 epochs, and apply label correction from 10 epochs. Once the rectification begins, we keep rectifying for every iteration.

4. Compare with SYSU30K

As show in Table 1, we compare the pre-trained models between LUPerson-NL and SYSU30K. We pre-train PNL on both LUPerson-NL and SYSU30K for 40 epochs with same experiment settings for a fair comparison. The performance of LUPerson-NL pre-training is much better than SYSU30K pre-training, showing the superiority of our LUPerson-NL, and also suggesting that a large number of images with limited diversity does not bring more representative representation, but large-scale images with diversity does.

5. Hyper Parameter Analysis

In our PNL, there are two key hyper-parameters: temperature factor τ and the threshold for correction T. Here, we provide the analysis of these two parameters.

5.1. Temperature Factor τ

Table 2 shows the performance comparison with different τ values with a fixed label correction threshold $T = 0.8$. As we can see, the setting $\tau = 0.1$ achieves the best results on both DukeMTMC and MSMT17, while $\tau = 0.07$ achieves the second best. When we use a larger τ, the performance drops rapidly. It may be because Re-ID is a

Algorithm 1: PNL algorithm.

Input: total epochs N, correction start epochs N_s, prototype feature vectors $\{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_K\}$, where K is the number of identities, temperature τ, threshold T, momentum m, encoder network $E_q(\cdot)$, classifier $h(\cdot)$, momentum encoder $E_k(\cdot)$, loss weight λ_{pro} and λ_{lgc}.

for *epoch* = $1 : N$ **do**

  $\{(\vec{x}_i, y_i)\}_{i=1}^{b}$ sampled from data loader.

  for $i \in \{1, ..., b\}$ **do**

    $\tilde{\vec{x}}_i = \text{aug}_1(\vec{x}_i)$, $\tilde{\vec{x}}'_i = \text{aug}_2(\vec{x}_i)$

    $\vec{q}_i = E_q(\tilde{\vec{x}}_i)$, $\vec{k}_i = E_k(\tilde{\vec{x}}'_i)$

    $\vec{p}_i = h(\vec{q}_i)$

    $\vec{s}_i = \{s_i^k\}_{k=1}^K, s_i^k = \frac{\exp(\vec{q}_i \cdot \vec{c}_k/\tau)}{\sum_{k=1}^K \exp(\vec{q}_i \cdot \vec{c}_k/\tau)}$

    $\vec{l}_i = (\vec{p}_i + \vec{s}_i)/2$

    if *epoch* $> N_s$ *and* $\max_k l_i^j > T$ **then**

      $\hat{y}_i = \arg\max_j \vec{l}_i^j$

    else

      $\hat{y}_i = \vec{y}_i$

    $\mathcal{L}_{ce}^i = -\log(\vec{p}_i[\hat{y}_i])$

    $\mathcal{L}_{\text{pro}}^i = -\log \frac{\exp(\vec{q}_i \cdot \vec{c}_{\hat{y}_i}/\tau)}{\sum_{j=1}^K \exp(\vec{q}_i \cdot \vec{c}_j/\tau)}$

    $\mathcal{L}_{lgc}^i =$

    $$\frac{-1}{|\mathcal{P}(i)|} \log \frac{\sum_{X} \exp\left(\frac{\vec{q}_i \cdot \vec{k}^+}{\tau}\right)}{\sum_{\vec{k}^+ \in \mathcal{P}(i)} \exp\left(\frac{\vec{q}_i \cdot \vec{k}^+}{\tau}\right) + \sum_{\vec{k}^- \in \mathcal{N}(i)} \exp\left(\frac{\vec{q}_i \cdot \vec{k}^-}{\tau}\right)}$$

    $\vec{c}_{\hat{y}_i} = m\vec{c}_{\hat{y}_i} + (1-m)\vec{q}_i$

    $\mathcal{L}^i = \mathcal{L}_{ce}^i + \lambda_{pro}\mathcal{L}_{pro}^i + \lambda_{lgc}\mathcal{L}_{lgc}^i$

    update networks E_q, h to minimize $\mathcal{L}^i$

    update networks E_q, h to minimize $\mathcal{L}^i$

    update momentum encoder E_k.

Dataset	LUPerson-NL	SYSU30K
MSMT17	66.1/84.6	55.2/76.7

Table 1. Comparison of applying our PNL on both LUPerson-NL and SYSU30K.

more fine-grained task, larger τ will cause smaller inter-class variations and make positive samples too close to negative samples. In all the experiments, we set $\tau = 0.1$.

5.2. Threshold T

Table 3 shows the results with different label correction threshold values T. As we can see, the performance is relatively stable to different T values varying in a large range of $0.6 \sim 0.8$. However, if T is too small or too large, the performance drops rapidly. For the former case, labels are easier to be modified, which may cause wrong rectifications, while for the latter case, the label noises become harder to be corrected, which also has consistently negative effects on the performance. In all the experiments, we set $T = 0.8$.

τ	DukeMTMC		MSMT17	
	40%	100%	40%	100%
0.05	76.1/87.2	83.5/91.4	49.8/73.3	65.4/83.8
0.07	76.4/**87.5**	83.6/91.4	50.7/74.1	67.2/85.3
0.1	**77.0**/87.3	**84.3/92.0**	**51.9/74.9**	**68.0/86.0**
0.2	75.8/86.8	83.4/91.0	50.1/73.7	66.4/85.0
0.3	74.7/86.2	82.7/90.6	48.6/72.0	65.5/83.7

Table 2. Performances under different τ values on DukeMTMC and MSMT17 with data percentages 40% and 100% under the small scale setting. The threshold is set as $T = 0.8$. The best scores are in bold and the second ones are underlined.

T	DukeMTMC		MSMT17	
	40%	100%	40%	100%
0.5	76.1/86.5	83.3/91.0	51.1/74.6	67.5/85.2
0.6	**77.1/87.7**	84.1/91.6	**52.3**/75.5	68.1/85.7
0.7	77.0/87.5	84.0/91.8	51.9/**75.6**	**68.2**/85.8
0.8	77.0/87.3	**84.3/92.0**	51.9/75.0	68.0/**86.0**
0.9	75.7/86.4	83.0/90.8	50.9/74.3	67.2/85.0

Table 3. Performances under different T values on DukeMTMC and MSMT17 with data percentages 40% and 100% under the small scale setting. The temperature factor is set as $\tau = 0.1$. The best scores are in bold and the second ones are underlined.

Arch	CUHK03	Market1501	DukeMTMC	MSMT17
R50	80.4/80.9	91.9/96.6	84.3/92.0	68.0/86.0
R101	80.5/81.2	92.5/**96.9**	85.5/**92.8**	70.8/87.1
R152	**80.6/81.2**	**92.7**/96.8	**85.6**/92.4	**71.6/87.5**

Table 4. Results with different ResNet backbones. R50, R101 and R152 stand for ResNet50, ResNet101 and ResNet152 respectively.

6. Results for stronger backbones

We train our PNL using two stronger backbones ResNet101 and ResNet152, and report the results in Table 4. As we can see, the stronger ResNet bring more superior performances. These results also outperform the scores reported in Table 8 of our main submission. Most importantly, we are the **FIRST** to obtain a *mAP* score on MSMT17 that is larger than 70 without any post-processing for convolutional network.

7. Pre-training data scales

We study the impact of pre-training data scale. Specifically, we involve various percentages (10%,30%,100% pseudo based) of LUPerson-NL into pre-training and then evaluate the finetuing performance on the target datasets.

Scale	10%	30%	100%
MSMT17	57.4/79.2	62.2/82.1	68.0/86.0

Table 5. Comparison for different pre-training data scale

As shown in Table 5, the learned representation is much stronger with the increase of the pre-training data scale, indicating the necessity of building large-scale dataset, and our LUPerson-NL is very important.

8. More results for small-scale and few-shot

To complement Table 3 in the main text, we provide more detailed results under "small scale" and "few shot" settings in Table 6. As we can see, our weakly pre-trained models are consistently better than other pre-trained models. Our advantage is much larger with less training data, suggesting the potential practical value of our pre-trained models for real-world person ReID applications.

scale	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
#id	75	150	225	300	375	450	525	600	675	751
#images	1,170	2,643	3,962	5,226	6,408	7,814	9,120	11,417	11,727	12,936
IN sup.	53.1/76.9	67.7/86.8	75.2/90.8	79.1/92.5	81.5/93.5	81.5/93.5	84.8/94.5	85.9/95.2	86.9/95.2	87.5/95.1
IN unsup.	58.4/81.7	70.2/89.1	76.6/91.9	80.0/93.0	82.0/94.1	83.7/94.3	85.4/94.5	86.4/95.0	87.4/95.5	88.2/95.3
LUP unsup.	64.6/85.5	76.9/92.1	81.9/93.7	84.1/94.4	85.8/94.9	87.8/95.8	88.8/95.9	89.8/96.2	90.5/96.4	91.0/96.4
LUPws wsp.	**72.4/88.8**	**81.7/93.2**	**85.2/94.2**	**87.3/95.1**	**88.3/95.5**	**89.6/96.0**	**90.1/96.2**	**90.9/96.4**	**91.3/96.4**	**91.9/96.6**

(a) Market1501 small-scale

scale	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
#id	751	751	751	751	751	751	751	751	751	751
#images	1,293	2,587	3,880	5,174	6,468	7,758	9,055	10,348	11,642	12,936
IN sup.	21.1/41.8	53.2/75.1	68.1/87.6	75.4/90.4	80.2/92.8	83.0/93.6	84.2/94.0	86.3/94.7	86.7/94.6	87.5/95.1
IN unsup.	18.6/36.1	56.5/77.5	69.3/87.8	78.8/88.3	78.3/90.9	81.7/93.3	84.4/94.1	86.4/95.0	87.1/95.2	88.2/95.3
LUP unsup.	26.4/47.5	63.5/83.0	78.3/92.1	80.3/92.7	84.2/93.9	86.7/94.7	88.4/95.5	89.8/96.0	90.4/96.3	91.0/96.4
LUPws wsp.	**42.0/61.6**	**75.7/89.1**	**83.7/94.0**	**86.0/94.3**	**88.1/95.2**	**89.8/95.8**	**90.5/96.3**	**91.2/96.4**	**91.6/96.4**	**91.9/96.6**

(b) Market1501 few-shot

scale	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
#id	70	140	210	280	351	421	491	561	631	702
#images	1,670	3,192	5,530	6,924	8,723	10,197	11,939	13,500	15,111	16,522
IN sup.	45.1/65.3	58.3/75.4	64.7/80.2	68.5/83.0	71.8/84.6	74.1/85.6	75.5/86.8	76.8/87.3	78.0/88.3	79.4/89.0
IN unsup.	48.1/66.9	60.6/76.6	65.8/80.2	69.5/82.9	72.5/84.4	75.0/86.2	76.3/86.9	77.4/87.3	78.5/88.7	79.5/89.1
LUP unsup.	53.5/72.0	65.0/78.9	69.4/81.9	72.8/84.7	75.6/86.7	77.6/87.1	78.9/88.2	80.2/89.2	81.1/90.0	82.1/91.0
LUPws wsp.	**60.6/75.8**	**70.5/83.3**	**74.5/86.3**	**77.0/87.3**	**78.8/88.3**	**80.5/89.2**	**81.6/89.5**	**82.9/90.6**	**83.3/91.2**	**84.3/92.0**

(c) DukeMTMC small-scale

scale	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
#id	702	702	702	702	702	702	702	702	702	702
#images	1,679	3,321	4,938	6,599	8,278	9,923	11,564	13,201	14,860	16,522
IN sup.	31.5/47.1	56.2/72.1	65.4/79.8	71.0/83.9	73.9/85.7	75.8/86.6	77.2/87.8	78.3/88.6	79.1/88.8	79.4/89.0
IN unsup.	32.4/48.0	56.4/72.2	65.3/80.2	70.2/83.4	73.7/85.1	75.8/86.7	77.7/88.2	78.7/88.7	79.4/89.0	79.5/89.1
LUP unsup.	35.8/50.2	61.0/74.9	72.3/83.8	75.2/86.8	77.7/87.4	79.4/88.4	80.8/89.2	81.7/90.3	82.0/90.6	82.1/91.0
LUPws wsp.	**52.2/64.1**	**71.7/82.5**	**77.7/87.9**	**79.3/88.1**	**81.1/89.6**	**82.3/90.2**	**83.2/91.1**	**84.0/91.6**	**84.1/91.3**	**84.3/92.0**

(d) DukeMTMC few-shot

scale	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
#id	104	208	312	416	520	624	728	832	936	1,041
#images	3,659	6,471	9,787	13,006	15,917	19,672	22,680	26,335	29,529	32,621
IN sup.	23.2/50.2	34.6/**64.0**	41.9/70.8	46.7/74.5	50.3/76.9	53.9/79.4	56.9/81.2	59.6/82.4	61.9/84.2	63.7/85.1
IN unsup.	22.6/48.8	32.7/60.9	40.4/68.7	45.1/72.2	49.0/75.0	52.7/78.0	55.7/79.9	58.6/82.0	60.9/83.0	62.7/84.3
LUP unsup.	25.5/51.1	36.0/62.6	44.6/**71.4**	49.2/74.9	53.0/**77.7**	56.4/79.7	59.5/81.8	61.9/**83.6**	63.7/**85.0**	65.7/85.5
LUPws wsp.	**28.2/51.1**	**39.6**/63.7	**47.7**/71.2	**51.9/74.9**	**55.5**/77.2	**59.1/80.1**	**61.6/81.8**	**64.2**/83.3	**66.1**/84.8	**68.0/86.0**

(e) MSMT17 small-scale

scale	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
#id	1,041	1,041	1,041	1,041	1,041	1,041	1,041	1,041	1,041	1,041
#images	3,262	6,524	9,786	13,048	16,310	19,572	22,834	26,096	29,358	32,621
IN sup.	14.7/34.1	35.6/61.4	44.5/71.1	52.0/76.9	56.2/79.5	58.8/81.7	60.9/82.8	62.5/84.2	63.4/84.5	63.7/85.1
IN unsup.	13.2/29.2	33.5/58.6	41.4/67.1	47.7/72.7	53.3/77.6	56.5/79.6	59.1/81.5	60.9/82.3	62.4/83.8	62.7/84.3
LUP unsup.	17.0/36.0	37.4/61.4	49.0/73.6	53.9/78.5	57.4/80.5	60.0/82.1	62.9/83.5	64.2/84.5	65.0/85.1	65.7/85.5
LUPws wsp.	**24.5/42.7**	**45.6/67.2**	**53.2/74.4**	**58.6/78.8**	**62.2/81.0**	**64.1/82.6**	**65.8/83.8**	**67.2/84.7**	**67.4/85.3**	**68.0/86.0**

(f) MSMT17 few-shot

Table 6. Performance for small-scale and few-shot setting with MGN method for Market1501, DukeMTMC and MSMT17. "IN sup." and "IN unsup." refer to supervised and unsupervised pre-trained model on ImageNet, "LUP unsup." refers to unsupervised pre-trained model on LUPerson, "LUPws wsp." refers to our model pre-trained on LUPerson-WS using WSP. The first number is *mAP* and second is *cmc1*.