Graph Sampling Based Deep Metric Learning for Generalizable Person Re-Identification: Appendix

Shengcai Liao*
Inception Institute of Artificial Intelligence (IIAI)
Masdar City, Abu Dhabi, UAE
scliao@ieee.org

Ling Shao
Terminus Group
China
ling.shao@ieee.org

A. Pseudocode of the GS sampler

A pseudocode of the GS sampler is shown in Algorithm 1.

Algorithm 1: Graph Sampler

Input: Data source $\mathbf{D}$, feature extractor $\mathbf{f}$, pairwise distance function $\mathbf{d}$, batch size B, number of instances per class K.
Output: Sample iterator of the dataset $\mathbf{D}$.
Initialization: $pids$: list of all class IDs;
$index\_dict$: dictionary of list containing all sample indices of each class.
Procedure:

```
index = []
for p in pids:
    index.append(random.choice(index_dict[p], size=1))  # randomly select one sample per class
dataset = D(index)  # construct a small sub-dataset
X = f(dataset)  # extract features
dist = d(X, X)  # calculate pairwise distance
dist[i,i] = Inf  # ignore the diagonal elements
P = B / K  # number of classes in a mini batch
topk_index = topk(-dist, size=P-1)  # find nearest neighboring classes
index = []
for p in shuffle(pids):
    index.extend(random.choice(index_dict[p], size=K))  # randomly select K samples per class
    for k in topk_index[p]:
        index.extend(random.choice(index_dict[k], size=K))  # randomly select K samples per class
```

Return: iter(index)

*Shengcai Liao is the corresponding author.

B. Alternative Loss Function and Analysis

B.1. Binary Cross Entropy Loss

Note that the batch hard triplet loss (Eq. (1) in the main paper) is usually used as an auxiliary to the classification loss, but not alone, in person re-identification. This is probably because random samplers including PK cannot provide informative mini batches for OHEM to mine, which makes Eq. (1) very small or even zero, and so the learning is not efficient. In contrast, with the proposed GS sampler, we prove that the OHEM triplet loss works well by itself with $K = 2$. We use this loss function alone because, as motivated in the main paper, we aim at removing classification layers for large-scale metric learning.

However, note that the GS sampler already provides almost the hardest mini batches, and the batch hard triplet loss further finds the hardest triplets within a mini batch for training. As a result, the model may suffer optimization difficulty, which in turn may impact convergence during training. In practice, we find that limiting $K = 2$ alleviates this problem significantly, while $K > 2$ usually makes the learning not able to converge.

Alternatively, pairwise verification or binary classification is another solution [3, 9] for pairwise matching or metric learning within mini batches. Specifically, we apply QAConv to compute similarity values between a pair of images, and formulate a pairwise verification or binary classification problem in mini-batch based learning. Accordingly, we compute the binary cross entropy loss as follows.

$$\ell(\boldsymbol{\theta}) = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j\neq i} y_{ij} log(p_{ij}(\boldsymbol{\theta})) + (1-y_{ij}) log(1-p_{ij}(\boldsymbol{\theta})), \tag{A}$$

where B is the mini-batch size, $\boldsymbol{\theta}$ is the network parameter, $p_{ij} \in [0, 1]$ is the QAConv similarity indicating binary classification probability, and $y_{ij} = 1$ indicates a positive pair, while a negative pair otherwise. By default, we choose $B = 64$ and $K = 4$ for this loss.

Train	Method	CUHK03		Market		MSMT17	
		R1	mAP	R1	mAP	R1	mAP
Market	Binary	16.4	15.7	-	-	41.2	15.0
	Triplet	**19.1**	**18.1**	-	-	**45.9**	**17.2**
MSMT	Binary	20.0	19.2	75.1	46.7	-	-
	Triplet	**20.9**	**20.6**	**79.1**	**49.5**	-	-
MS-all	Binary	27.2	27.1	80.6	55.6	-	-
	Triplet	**27.6**	**28.0**	**82.4**	**56.9**	-	-
RP	Binary	14.8	13.4	74.0	43.8	42.4	14.4
	Triplet	**18.4**	**16.1**	76.7	46.7	**45.1**	15.5

Table A. Comparison of loss functions. Binary: binary cross entropy loss. Triplet: hard triplet loss. Market: Market-1501 dataset. MSMT: MSMT17 dataset. MS-all: MSMT17 (all). RP: RandPerson dataset.

B.2. Experimental Comparison

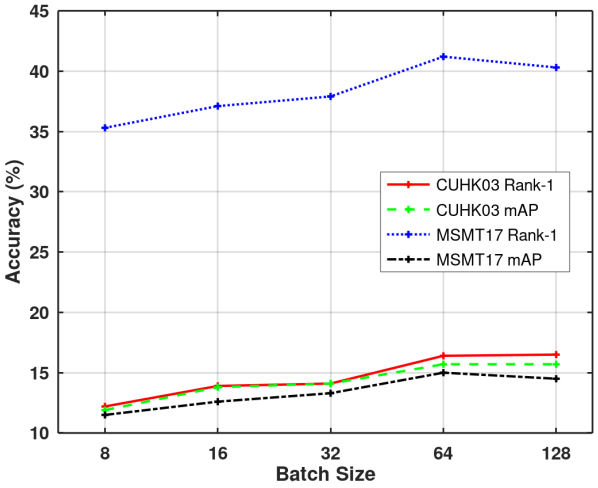
Table A shows a comparison between the hard triplet loss and the binary cross entropy loss for QAConv-GS. Results shown in the table indicate that, the hard triplet loss performs better than the binary cross entropy loss for all datasets, thanks to OHEM which further mines hard examples within mini batches provided by GS. However, the hard triplet loss used alone in the proposed pipeline is sensitive to K values as discussed. In contrast, the binary cross entropy loss is a more stable alternative, working well with different B and K values. This will be analyzed in the following subsection.

B.3. Parameter analysis

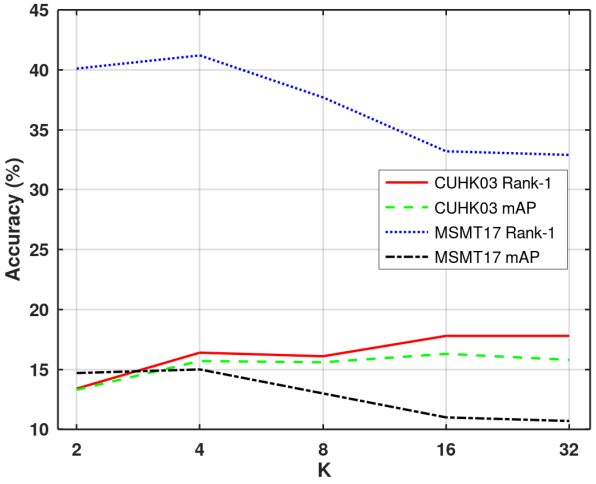
When the binary cross entropy loss is applied, in Fig. A, we show the performance with different parameter configurations of the GS sampler, trained on Market-1501. We observe that for the batch size (Fig. A (a)), generally the accuracy increases with the increasing batch size (thus increasing P), but saturates at about 64. It is understood that mini batches with larger batch size provides more comprehensive data for learning, however, at the cost of enlarged computation time, recalling that the number of iterations per epoch is fixed as C for the GS sampler. For example, with $B = 64$, the training of QAConv-GS on Market-1501 is about 0.68 hours on a single V100 GPU. However, this is about 1.32 hours with $B = 128$ for training the same epochs.

Next, we evaluate the influence of K under fixed $B = 64$, as shown in Fig. A (b). Interestingly, larger K leads to gradually better performance on the CUHK03-NP, however, it degrades the performance significantly on MSMT17. It appears that $K = 4$ is a reasonable trade-off.

Since the hard triplet loss performs better, in the following, by default we still use this loss.



(a) Effect of batch size



(b) Effect of K

Figure A. Performance with different parameter configurations of the GS sampler when the binary cross entropy loss is applied, trained on Market-1501. (a) with varying batch size under fixed $K = 4$; and (b) with varying K under fixed $B = 64$.

C. Application to Other Baselines

Furthermore, to show the generality of the proposed graph sampling method, we apply it to two other algorithms, namely OSNet [10] and TransMatcher [4].

The official code of OSNet[1] (MIT License) is used. We used its osnet_ibn_x1_0 config, with softmax+triplet loss and the PK sampler (RandomIdentitySampler) for the best performance, denoted by OSNet-IBN + PK. This combination of softmax+triplet loss and the PK sampler is also the most popular setting in person re-identification for strong baselines. Then, upon this baseline, we apply the pro-

[1] https://github.com/KaiyangZhou/deep-person-reid

posed graph sampling to replace the PK sampler, denoted by OSNet-IBN + GS. The training is performed on the MSMT17 (all), as in [10], and the learned models are evaluated on the CUHK03-NP and Market-1501 datasets. The results are shown in Table B. From the comparison it can be seen that the proposed GS sampler can also improve other strong baselines in replacing the popular PK sampler. Therefore, it is proved to be general and may also be applied to other methods.

Table B. Direct cross-dataset evaluation results (%) with different baselines trained on MSMT17 (all).

Method	CUHK03		Market	
	R1	mAP	R1	mAP
OSNet-IBN [10]	-	-	66.5	37.2
OSNet-IBN + PK	23.4	23.6	67.9	39.6
OSNet-IBN + GS	**24.5**	**24.9**	**71.3**	**42.6**

Furthermore, with a very recent method TransMatcher [4], we also compare the PK and GS samplers. The official code of TransMatcher [2] (MIT License) is used, with its default settings. The results are shown in Table C. It can also be observed that on average the proposed GS sampler performs much better than the PK sampler, verifying again the generality of GS.

D. Application to Unsupervised Domain Adaptation

As a new scenario, we tried unsupervised domain adaptation (UDA) by replacing PK with GS in the source domain of SpCL [1]. The Rank1/mAP results for PK and GS are 86.1/70.9 and 87.3/71.5, respectively, for CUHK03-NP→Market-1501. Slight improvements can be observed. However, for the time being, it is still not yet straightforward to apply GS for pseudo labeled samples by clustering on the target domain. This may be because pseudo labels could be noisy, and GS may be aggressive in finding hard negative samples that could possibly be positive. To address this, further developments may be required in considering how to handle noisy samples, which is quite interesting.

E. Further Ablation Studies

In this paper, all experiments are with images of 384×128 as inputs. To understand the influence of image size, we also conduct experiments with images of 256×128 as inputs. The results are shown in Table D. It can be seen that the results are quite close to each other on Market-1501 and MSMT17, though results with 384×128 are clearly better than that of 256×128 on CUHK03. Note that our results with 256×128 still achieve the state of the art compared to existing methods in Table 1 of the main paper.

In the proposed GS, one example per class is sampled for the graph construction, which is efficient. Alternatively, class centers can also be considered for graph construction. In fact, class centers are used in [6] for clustering based batch sampling, and we show better performance of GS in Table 3 of the main paper. To further understand this, we use class centers to construct graphs for GS. This is denoted by QAConv-GS-Center. The comparison results are shown in Table D. It is clear that GS performs better.

There might be two problems with class centers. First, it lacks flexibility of sample relationships, since many classes may have large distribution variances. This is also discussed in [5]. Second, computing class centers requires feature extraction of all training samples, which hinders large-scale learning. The average training time increases from 1.68 hours of QAConv-GS to 2.27 hours of QAConv-GS-Center. Especially, QAConv-GS costed 3.4 hours to train MSMT17 (all), but QAConv-GS-Center costed 5.4 hours.

F. Visualization of GS

Finally, we show some examples for the nearest neighboring classes generated by the GS sampler in Fig. B. It can be observed that, the GS sampler is indeed able to find similar classes as hard examples to challenge the learning. For example, similar kind of clothes, similar colors, patterns, and accessories. These confusing examples helps a lot in learning discriminative models. Besides, it seems that in early epochs, the model tends to evaluate similarity with visual appearance, regardless of the influence of foreground and background. However, in late epochs, the model learns to remove the influence of background, and learns higher level of abstraction. For example, in the upper right group, the similarity is less affected by bicycles in background with epoch 15. In the first group of MSMT17, the similarity is less affected by trees in background with later epochs. With epoch 15 of the first group, the model learns the concept of security guards. In the upper right group, with epoch 15 the model learns the concept of girls with short skirts. In the last group of Market-1501, with epoch 15 the clothes are more consistent in style and color. In the upper right group of MSMT17, with epoch 15 in GS the model correctly retrieves red coats. In the last group of MSMT17, with epoch 15 in GS the model correctly retrieves pink coats as well.

G. Limitations

The proposed method, despite achieving very good results, may have two limitations. First, GS requires additional computation for mini batch sampling. We design two ways to reduce the computation, that is, employing GS only at the beginning of each epoch, and randomly sampling only

[2] https://github.com/ShengcaiLiao/QAConv/tree/master/projects/transmatcher

Method	Training	CUHK03-NP		Market-1501		MSMT17	
		Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
TransMatcher-PK	Market-1501	**22.9**	**21.5**	-	-	45.6	17.8
TransMatcher-GS	Market-1501	22.2	21.4	-	-	**47.3**	**18.4**
TransMatcher-PK	MSMT17	23.6	**22.9**	78.3	51.7	-	-
TransMatcher-GS	MSMT17	**23.7**	22.5	**80.1**	**52.0**	-	-
TransMatcher-PK	MSMT17 (all)	30.7	29.5	79.9	55.7	-	-
TransMatcher-GS	MSMT17 (all)	**31.9**	**30.7**	**82.6**	**58.4**	-	-
TransMatcher-PK	RandPerson	**18.0**	**16.5**	73.3	45.3	40.6	14.1
TransMatcher-GS	RandPerson	17.1	16.0	**77.3**	**49.1**	**48.3**	**17.7**

Table C. Comparison of direct cross-dataset evaluation results (%) using TransMatcher [4] with PK and GS. MSMT17 (all) means all images are used for training, regardless of subset splits.

Method	Training	CUHK03-NP		Market-1501		MSMT17	
		Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
QAConv-GS	Market-1501	**19.1**	**18.1**	-	-	**45.9**	**17.2**
QAConv-GS (256×128)	Market-1501	16.9	17.2	-	-	45.4	17.1
QAConv-GS-Center	Market-1501	15.4	14.7	-	-	45.0	15.7
QAConv-GS	MSMT17	**20.9**	**20.6**	**79.1**	49.5	-	-
QAConv-GS (256×128)	MSMT17	18.6	19.8	77.9	**49.6**	-	-
QAConv-GS-Center	MSMT17	15.3	16.1	73.9	41.5	-	-
QAConv-GS	MSMT17 (all)	**27.6**	**28.0**	**82.4**	**56.9**	-	-
QAConv-GS (256×128)	MSMT17 (all)	24.3	25.6	81.5	55.3	-	-
QAConv-GS-Center	MSMT17 (all)	25.2	24.6	78.6	51.2	-	-
QAConv-GS	RandPerson	**18.4**	**16.1**	76.7	46.7	**45.1**	15.5
QAConv-GS (256×128)	RandPerson	16.2	14.4	74.7	45.5	45.0	**15.8**
QAConv-GS-Center	RandPerson	17.4	15.4	**76.8**	**47.0**	44.3	15.2

Table D. Comparison of the direct cross-dataset evaluation results (%) for different variants of QAConv-GS. QAConv-GS-Center is based on selecting class centers for graph construction for GS. MSMT17 (all) means all images are used for training, regardless of subset splits.

one sample per class for the distance computation and graph construction. As a result, the additional running time introduced by GS is still acceptable, as reported in Section 5.4.1 of the main paper. Besides, note that with GS the number of training epochs is generally reduced. For example, with GS the proposed method usually requires less than 20 epochs for training, while existing methods typically require 60 epochs or more to train. Therefore, GS deserves the additional computational costs. However, in our experiments, the maximal number of classes is only 8,000. GS may still have a big limitation with millions of identities, which need further investigation.

Second, as discussed, GS provides challenging examples for training, and so the default hard triplet loss only works well with $K = 2$. Otherwise, the training is too difficult to converge. Nevertheless, as discussed in Section B, this limitation can be solved by employing the binary cross entropy loss as an alternative, though with inferior performance.

H. Social Impacts

Person re-identification is a technique to automatically search persons from a large amount of videos. It has potential social values in some practical applications, such as person image retrieval of suspects, character recognition in movies [2], and so on. For example, it is very useful to reduce large amount of human labors and greatly advance the effort in criminal investigation. Accordingly, person re-identification methods are actively studied. The person re-identification technique, however, may also be used by company for a surveillance of employees, or by malls for tracking of daily visitors. Therefore, it requires effective legislation to avoid abuse of this technique. This paper focuses on foundational research; it is not tied to particular applications, let alone deployments.

Besides, the research and developments of such technique are often with datasets collected from surveillance videos that may contain personally identifiable information. To address this, a positive action is to remove such informa-

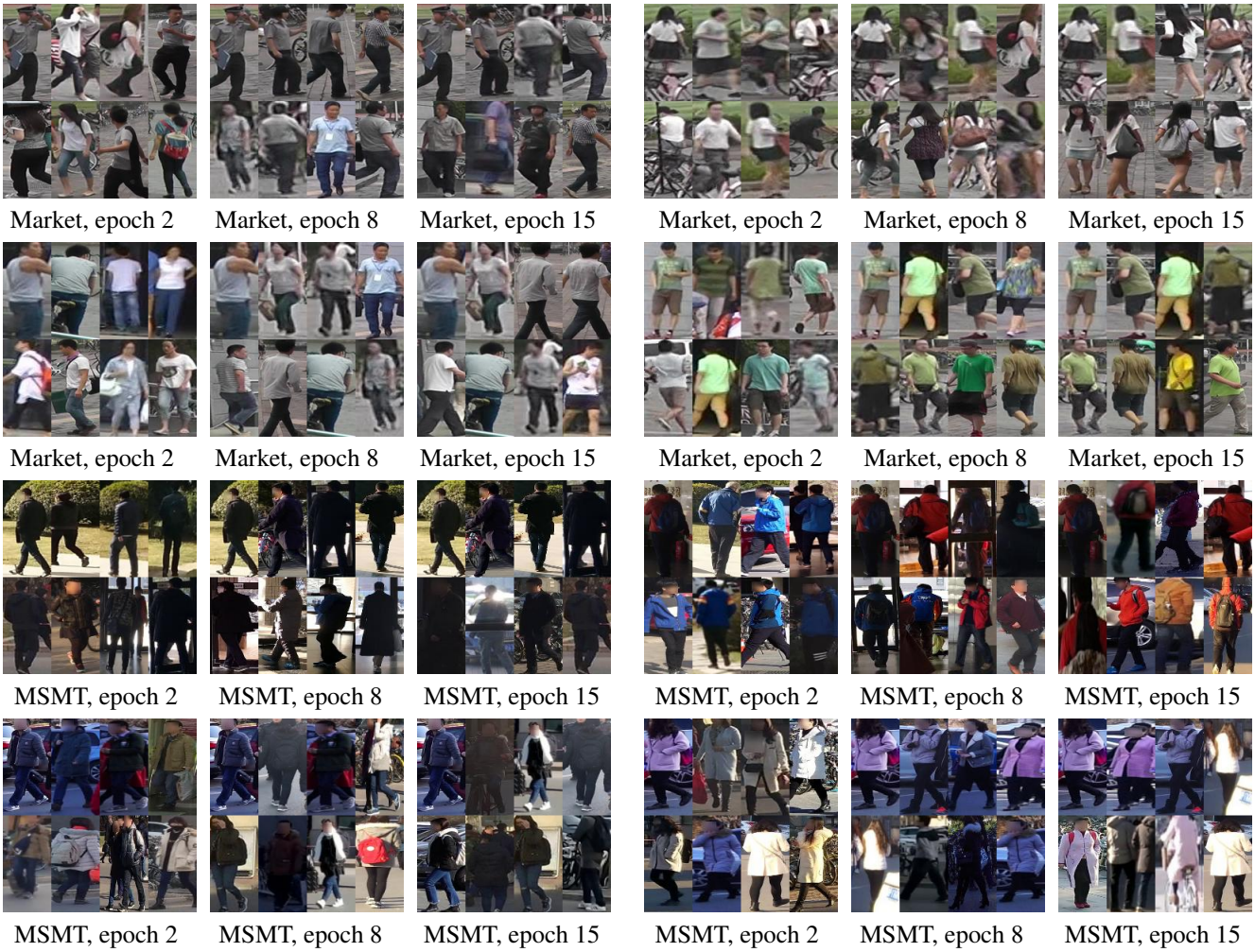


Figure B. Eight groups of examples for the nearest neighboring classes generated by the GS sampler. The first two rows are from the training on Market-1501, while the last two rows are from that of MSMT17. In each group, three sets of images are shown, corresponding to epoch 2, 8, and 15. In each set, the upper left image is the center class, and other images are the top-7 nearest neighboring classes to the center class.

tion, as done in MSMT17v2 [8] with facial areas masked. More promisingly, a better way recently demonstrated is to use synthesized data, as done in RandPerson [7].

References

[1] Yixiao Ge, Feng Zhu, Dapeng Chen, Rui Zhao, et al. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. *Advances in Neural Information Processing Systems*, 33:11309–11321, 2020. 3

[2] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *The European Conference on Computer Vision (ECCV)*, 2020. 4

[3] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014. 1

[4] Shengcai Liao and Ling Shao. TransMatcher: Deep Image Matching Through Transformers for Generalizable Person Re-identification. In *Advances in Neural Information Processing Systems*, 2021. 2, 3, 4

[5] Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. Stochastic class-based hard example mining for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7251–7259, 2019. 3

[6] Chong Wang, Xue Zhang, and Xipeng Lan. How to train triplet networks with 100k identities? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1907–1915, 2017. 3

[7] Yanan Wang, Shengcai Liao, and Ling Shao. Surpassing Real-World Source Training Data: Random 3D Characters

for Generalizable Person Re-Identification. In *28th ACM International Conference on Multimedia (ACMMM)*, 2020. 5

[8] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018. 5

[9] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. Deep metric learning for person re-identification. In *International Conference on Pattern Recognition*, pages 34–39, Dec. 2014. 1

[10] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2, 3