

Table 4. Generalization comparison of various training sets: those generated by random sampling, greedy sampling, and SnP, as well as the source data pool. AlicePerson is used as the target. We directly test the trained models on Market, Duke, MSMT and report the individual and the average rank-1 accuracy (%) and mAP (%).

Train. set	Market		Duke		MSMT		Average		AlicePerson	
	R1	mAP	R1	mAP	R1	mAP	R1	mAP	R1	mAP
Sour. pool	67.36	43.16	62.39	41.15	35.72	15.40	55.16	33.23	38.96	17.62
Random	6.44	1.92	38.29	18.75	19.90	6.42	21.54	9.03	23.67	9.80
Greedy	74.11	51.98	26.08	13.27	7.32	2.13	35.83	22.46	33.22	15.10
SnP	74.32	51.03	27.06	13.43	8.19	2.28	36.52	22.24	38.23	18.17

Table 5. Comparison of different source pools. A: Market + Duke + MSMT. B: A + UnrealPerson. C: B + RandPerson + PersonX. All: C + CUHK03 + RAiD + VIPeR + PKU-Reid. SnP is consistently effective for different pools. 2% IDs are selected from the source pool. Notations and evaluation metrics are the same as those in the previous table.

Search method	Pool A		Pool B		Pool C		All	
	R1	mAP	R1	mAP	R1	mAP	R1	mAP
Random	11.47	4.01	21.56	8.61	23.60	9.39	23.67	9.80
Greedy	8.04	2.81	18.66	7.63	24.26	9.56	33.22	15.10
SnP	14.17	4.47	25.44	10.16	36.32	17.22	38.23	18.17

## 8. Experimental Details

**SnP settings.** For a dataset denoted as a target domain (e.g., Market and VeRi), their unlabeled training sets are used as the search target. For image feature extraction, we use InceptionV3 [42] pretrained on ImageNet [7].

**Task model configuration.** We have two types of models: direct transfer models and pseudo-label based models. For direct transfer, we use multiple task models, including ID-discriminative embedding (IDE) [56], the part-based convolution (PCB) [41], and TransReid [17]. For IDE, we adopt the strategy from [30] which uses ResNet-50 [16], adds batch normalization and removes ReLU after the final feature layer. For PCB, we use the ResNet-50 backbone and vertically partition an image into six equal horizontal parts. For pseudo-label methods, we use UDA [37] and MMT [14]. The unlabeled target training set (e.g., AlicePerson training set) is used for pseudo-label model training. Note that when implementing these task models, we use their official implementation with default hyperparameters including learning rate and training epochs.

**Computation Resources.** We run experiments on an 8-GPU machine and an oracle cloud machine. The 8-GPU machine has 8 NVIDIA 3090 GPUs and its CPU is AMD EPYC 7343 Processor. For the oracle cloud machine, it has 2 NVIDIA A10 GPUs and its CPU is Intel Platinum 8358 Processor. For both the 3090 machine and the oracle cloud machine, we use PyTorch version 1.12.1+cu116.

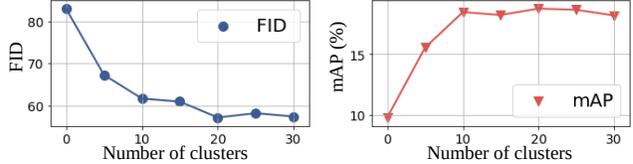


Figure 8. Impact of the number of clusters  $J$  to the domain gap (FID) between searched and target (left), and re-ID accuracy (right). AlicePerson is targeted, and the IDE [56] model is trained.

## 9. Further Analysis

**Generalization analysis of the training set generated by SnP.** In Table 4, we directly apply the re-ID model obtained from the AlicePerson-specific training set to various test sets, such as Market, Duke, and MSMT. Compared with the model trained on the source pool, our target-specific model has higher accuracy on the target test set, and lower accuracy on other domains (MSMT for example shown in Table 4). It demonstrates the generated training set is target-specific, improving re-ID accuracy on the given domain, while its generalization ability is weakened. But still, its application is promising given its target-specificity strength.

**Analysis of different source pools.** We compare various source pools when AlicePerson is the target domain. From Table 5, we observe that our method gives very stable accuracy when using different source pools. In comparison, the compared methods, *i.e.*, random sampling and greedy sampling, as much less stable. For example, random sampling performs well under Pool A which only contains real-world images and might be similar to the target domain. But it is much poorer under Pool C which has more synthetic data, meaning a larger domain gap from the target.

**Analysis of hyperparameter  $J$ .** In Fig. 8, we show the increase of the number of clusters  $J$  lowers the domain gap between the target and searched training set and generally increases the task accuracy. It reaches saturation after  $J = 20$ . In this paper, we use  $J = 50$  in all setups.

**Correlation between the number of images and training set quality.** While the correlation between the number of IDs and training set quality has been shown in Fig. 3. We show that when the number of ID is fixed, the correlation between the number of images and the rank-1 accuracy in Fig. 7. when we have a fixed number of IDs, the increase in the number of images brings a significant increase in rank-1 accuracy.

## 10. Source Pool Details

We show the composition (tree map) of the source pool for the training set search in Fig. 9. On the left of Fig. 9, for person re-ID, we summarize the details of each dataset in rectangles and show some examples. The size of each rectangle refers to the number of images in the dataset, while

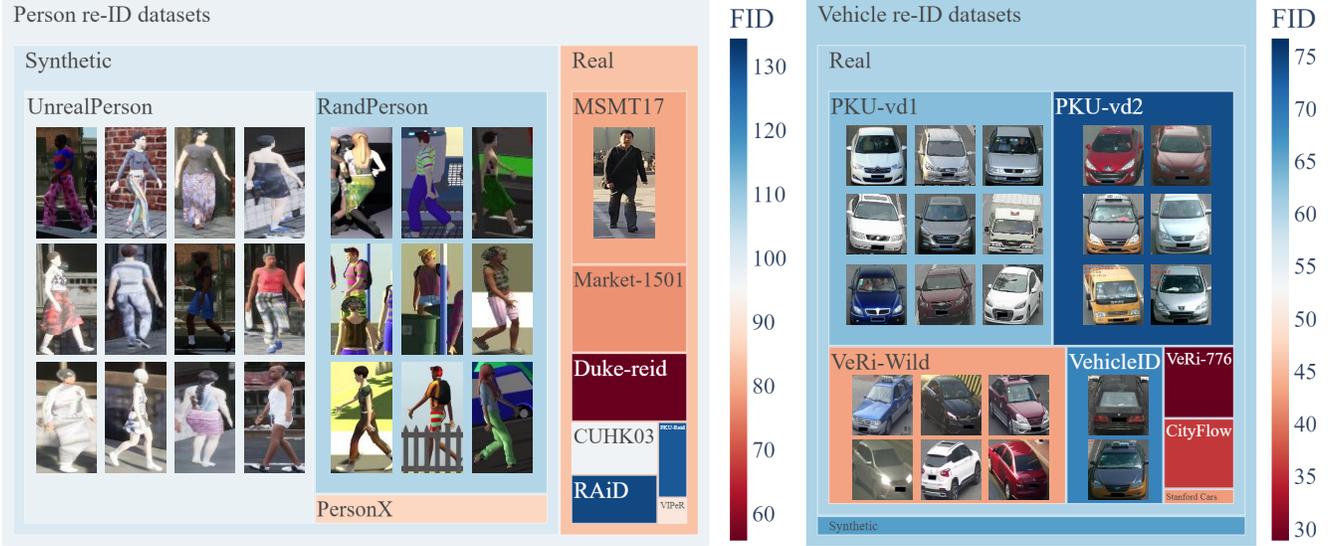


Figure 9. The composition (tree map) of the source pool for the training set search. **(Left:)** Person re-ID datasets were collected for constructing the source pool. **(Right:)** Vehicle re-ID datasets. For both person re-ID and vehicle re-ID, we use synthetic data and real data. Each innermost rectangle represents a publicly available object re-ID dataset. The relative size of rectangles denotes the relative image size of datasets. The relative color shows the domain gap (measured in FID) value to the target test set. In person re-ID, the target for calculating the domain gap is AlicePerson [1]. In vehicle re-ID, the target for calculating the domain gap is AliceVehicle [1].

the color of the rectangle indicates the domain gap of that dataset to the target set. Regarding the target set in person re-ID, we use AlicePerson [1] for calculating the FID.

We summarize the details of vehicle re-ID datasets in the right of Fig. 9. The size and color of each rectangle carry the same meanings as those on the left of Fig. 9. The AliceVehicle [1] is used as the target set for calculating FID.

The datasets we use for the source pool are standard benchmarks, which are publicly available. We list their open-source as follows. For person re-ID:

**Market** [57] [https://zheng-lab.cecs.anu.edu.au/Project/project\\_reid.html](https://zheng-lab.cecs.anu.edu.au/Project/project_reid.html);

**Duke** [58] [https://exposing.ai/duke\\_mtmc/](https://exposing.ai/duke_mtmc/);

**MSMT17** [48] <http://www.pkumc.com/publications/msmt17.html>;

**CUHK03** [21] [https://www.ee.cuhk.edu.hk/~xgwang/CUHK\\_identification.html](https://www.ee.cuhk.edu.hk/~xgwang/CUHK_identification.html);

**RAiD** [6] <https://cs-people.bu.edu/dasabir/raid.php>;

**PersonX** [40] <https://github.com/sxzrt/Instructions-of-the-PersonX-dataset>;

**UnrealPerson** [55] <https://github.com/FlyHighest/UnrealPerson>;

**RandPerson** [46] <https://github.com/VideoObjectSearch/RandPerson>;

**PKU-Reid** [31] <https://github.com/charliememory/PKU-Reid-Dataset>;

**VIPeR** [4] <https://vision.soe.ucsc.edu/node/178>.

For vehicle re-ID, we also list their open-source:

**VeRi** [24] <https://github.com/JDAI-CV/VeRidataset>;

**CityFlow** [44] <https://www.aicitychallenge.org/>;

**VehicleID** [23] <https://www.pkuml.org/resources/pku-vehicleid.html>;

**VeRi-Wild** [27] <https://github.com/PKU-IMRE/VERI-Wild>;

**VehicleX** [53] <https://github.com/yorkeyao/VehicleX>;

**Stanford Cars** [20] [http://ai.stanford.edu/~jkrause/cars/car\\_dataset.html](http://ai.stanford.edu/~jkrause/cars/car_dataset.html);

**PKU-vd1** [50] <https://www.pkuml.org/resources/pku-vds.html>;

**PKU-vd2** [50] <https://www.pkuml.org/resources/pku-vds.html>.