




Population-Based Evolutionary Gaming for Unsupervised Person Re-identification

Yunpeng Zhai¹ · Peixi Peng^{1,3} · Mengxi Jia¹ · Shiyong Li⁴ · Weiqiang Chen⁵ · Xuesong Gao⁵ · Yonghong Tian^{1,2,3} 

Received: 19 November 2021 / Accepted: 13 September 2022 / Published online: 1 October 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Unsupervised person re-identification has achieved great success through the self-improvement of individual neural networks. However, limited by the lack of diversity of discriminant information, a single network has difficulty learning sufficient discrimination ability by itself under unsupervised conditions. To address this limit, we develop a population-based evolutionary gaming (PEG) framework in which a population of diverse neural networks are trained concurrently through selection, reproduction, mutation, and population mutual learning iteratively. Specifically, the selection of networks to preserve is modeled as a cooperative game and solved by the best-response dynamics, then the reproduction and mutation are implemented by cloning and fluctuating hyper-parameters of networks to learn more diversity, and population mutual learning improves the discrimination of networks by knowledge distillation from each other within the population. In addition, we propose a cross-reference scatter (CRS) to approximately evaluate re-ID models without labeled samples and adopt it as the criterion of network selection in PEG. CRS measures a model's performance by indirectly estimating the accuracy of its predicted pseudo-labels according to the cohesion and separation of the feature space. Extensive experiments demonstrate that (1) CRS approximately measures the performance of models without labeled samples; (2) and PEG produces new state-of-the-art accuracy for person re-identification, indicating the great potential of population-based network cooperative training for unsupervised learning. The code is released on github.com/YunpengZhai/PEG.

Keywords Evolutionary gaming · Population-based training · Unsupervised learning · Person re-identification

1 Introduction

Person re-identification (re-ID) aims to match persons in an image gallery collected from non-overlapping camera

Communicated by Wenjun Kevin Zeng.

✉ Peixi Peng
pxpeng@pku.edu.cn

✉ Yonghong Tian
yhtian@pku.edu.cn

Yunpeng Zhai
ypzhai@pku.edu.cn

Mengxi Jia
mxjia@pku.edu.cn

Shiyong Li
lishiyong@huawei.com

Weiqiang Chen
chenweiqiang@iCloud.com

Xuesong Gao
xuesong@outlook.com

networks, which has attracted increasing interest thanks to its wide applications in security and surveillance. Though supervised re-ID methods (Yang et al., 2020; Zheng et al., 2016) have achieved very decent results, they are largely dependent on sufficient data with expensive manual annotation, which also require substantial personal identity information and entail privacy issues. By contrast, unsupervised re-ID not only reduces the cost of labeling but also

¹ National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing, China

² School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen, China

³ Peng Cheng Laboratory, Shenzhen, China

⁴ AI Application Research Center, Huawei Technologies Co., Ltd, Shenzhen, China

⁵ State Key Laboratory of Digital Multimedia Technology, Hisense, Qingdao, China

protects personal privacy without checking images manually. Commonly, unsupervised re-ID can be divided into two categories: unsupervised domain adaptation (UDA) (Zhai et al., 2020; Zhong et al., 2020) and fully unsupervised re-ID (FU) (Chen et al., 2021; Lin et al., 2019) depending on whether using extra labeled data. In this study, we will mainly focus on the fully unsupervised setting which learns directly from unlabeled images and allows for more scalability in real-world deployments.

To address the challenges of unsupervised re-ID, recent efforts concentrate on training individual neural networks by means of a self-improvement strategy (Song et al., 2018; Ge et al., 2020). They attempt to learn better representations based on self-predicted pseudo-labels via clustering algorithms (Caron et al., 2018) or graph neural networks (Ye et al., 2017). However, a single model can use such a self-learning mechanism only to enhance the discrimination ability it already has and cannot tackle the incorrectly predicted pseudo-labels, which prevents it from maximizing its discrimination. Due to the lack of diversity of single models, incorrect pseudo-labels are likely to remain the same after unsupervised training such as the false positive samples where images of different persons are clustered into the same group or the false negative samples where the images of the same person are clustered into different groups, as shown in Fig. 1. Importantly, since models learn diverse discrimination with different architectures, the incorrect pseudo-labels predicted by a model may be predicted correctly by another model, marked by boxes in Fig. 1b. In this paper, we attempt to address unsupervised re-ID by multiple model training, in which the complementary information of different models can be integrated and utilized effectively to explore the various latent knowledge contained in unlabeled data (the quantitative analysis is shown in Sect. 4.4.1).

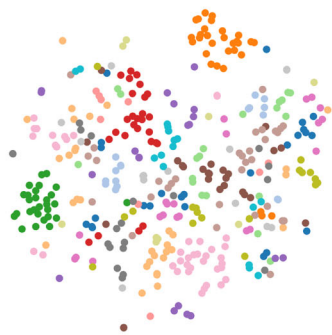
However, multiple model training still faces two challenging issues: (1) How to learn diverse discrimination with multiple different models? (2) How to select a set of better models from many diverse models for training? To tackle these issues, we propose a population-based evolutionary gaming (PEG), which selects and trains discriminative models by exploration and exploitation of their diversity. PEG trains a population of models concurrently by iterative selection, reproduction, mutation, and population mutual learning of neural networks, as shown in Fig. 2. Specifically, selection adapts the whole population to the unlabeled data by selecting and preserving the optimal subset of networks with complementary discrimination ability while abandoning other networks out of the subset. This combinatorial optimization of networks in selection is modeled as a multi-agent cooperative game and solved by the best response dynamics, in which each agent attempts to learn the best response to the other agents' action and thus leads to Nash equilibrium. Then, reproduction and mutation are performed on the selected pop-

ulation to increase its diversity by making multiple copies of each network and applying a stochastic disturbance to their hyper-parameters. Selection and reproduction jointly maintain the size of the population. Afterward, population mutual learning is conducted among networks to assemble and further explore the discrimination capacity via knowledge distillation within populations. Each network learns representations from both population-shared pseudo-labels and soft-labels predicted by other individual networks. Utilizing periodically performing selection, reproduction and mutation, population mutual learning, the evolutionary gaming process enables favorable traits and knowledge of neural networks to be transmitted through successive generations.

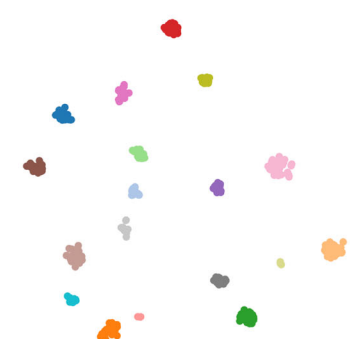
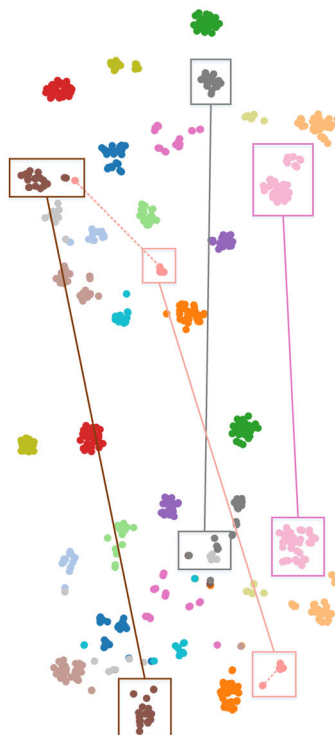
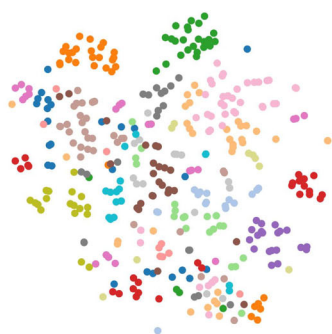
In the evolution gaming, a core issue is to define the utility function of the game, that is, the criterion of network selection in the evolution. However, the evaluation of CNN models without labeled datasets has not been well studied. Here, we propose cross-reference scatter (CRS), which can approximately evaluate the quality of networks using unlabeled samples. Generally, the pseudo-labels predicted by better networks are more accurate; however, their accuracy cannot be directly evaluated when the ground truth is unavailable. Moreover, models trained by more accurate pseudo-labels tend to achieve larger intra-cluster cohesion and inter-cluster separation in the feature space because incorrect labels will enforce models to separate samples of the same class or aggregate samples of different classes. Motivated by this phenomenon, we indirectly evaluate a network according to the feature cohesion and separation of a reference model that is trained by pseudo-labels of the evaluated network. Hence, the CRS is defined by the ratio of the inter-cluster and intra-cluster variance of features to measure both separation and cohesion. We demonstrate that the CRS approximately reflects the discrimination capacity of models without ground truth data and thus promotes the evolution gaming to learn better representations.

A preliminary version of this work has been partially published (Zhai et al., 2020), which has demonstrated the effectiveness of mutual learning among multiple networks in unsupervised conditions. Based on that version, this manuscript has made great improvements, including: (1) We propose a novel population-based evolutionary gaming (PEG) framework (Sect. 3.1). The previous algorithm works passively only on given networks, and cannot adaptively select the most suitable models from the model base. Based on the mutual learning, PEG additionally contains an iterative selection of networks via a multi-agent cooperative game preventing the weak networks to distract the overall discrimination capability (Sect. 3.1.1). (2) We propose a new cross-reference scatter (CRS) to approximately measure re-ID models without labeled data. To evaluate the model discrimination, the previous version introduced inter-/intra-cluster scatter to roughly modulate the weights of models

ResNet-50



DenseNet-169



(a) Before training

(b) Single model training

(c) Multi-model training (Ours)

Fig. 1 Feature distribution of the same samples with different methods where each color denotes a person identity. Single model training (b) uses the self-learning mechanism only to enhance the discrimination ability it already has before training (a) and still suffers from inaccurate

pseudo-labels. However, multi-model training (c) explores and exploits the complementary information among different models (marked by corresponding colored boxes) and achieves more discrimination

during mutual learning. However, it cannot be considered as the utility function of the cooperative game in PEG due to the lack of capability to accurately evaluate models. This paper improves inter-/intra-cluster scatter to cross-reference scatter by adding a cross-reference evaluation (CR) scheme (Sect. 3.1.1). (3) More qualitative and quantitative experiments are conducted to evaluate the effectiveness of the method, including but not limited to the validation and analysis of CRS, the cooperative game, and PEG.

In summary, our contribution is as follow:

- It proposes a novel population-based evolutionary gaming framework for unsupervised person re-ID which trains a diverse population of neural networks by iterative selection, reproduction, mutation and mutual learning.
- It introduces a multi-agent cooperative game for the selection of networks in the PEG, which aims to find and preserve an optimal subset of the population on unlabeled data.
- It investigates the evaluation of re-ID models using unlabeled data and proposes a cross-reference scatter which approximately measures a model’s discrimination capa-

bility by indirectly estimating its predicted pseudo-labels according to the cohesion and separation of feature space.

- Experiments show that PEG outperforms state-of-the-art methods on large-scale datasets, indicating the great potential of population-based multiple model training.

2 Related Works

2.1 Unsupervised Person Re-ID

Unsupervised person re-ID can be categorized into Unsupervised Domain Adaptation (UDA) and Fully Unsupervised Re-ID (FU). UDA methods try to train a re-ID model by unlabeled target data together with labeled source data, while FU methods attempt to train models with only unlabeled data after pre-training. Despite the different data conditions, most UDA and FU methods adopt similar learning strategies which can be summarized into two categories. A line of works are mainly based on alignment to reduce distribution shift between cameras or domains in pixel level, such as SPGAN (Deng et al., 2018), CamStyle (Zhong et

al., 2019b), HHL (Zhong et al., 2018), ECN (Zhong et al., 2019), ATNet (Liu et al., 2019), PDA-Net (Li et al., 2019), DG-Net++ (Zou et al., 2020) and GCL (Chen et al., 2021), or feature level, such as TJ-AIDL (Wang et al., 2018), DAAM (Huang et al., 2019), UCDA-CCE (Qi et al., 2019) IICS (Xuan & Zhang, 2021) and CAP (Wang et al., 2020). This line of methods sufficiently utilize the reliable information of camera or domain styles but ignore the latent relationship among unlabeled samples, which hinders them from better performance. Another line of works are based on pseudo label discovery, which rely on the iteration of pseudo-label mining and model fine-tuning (Fan et al., 2018; Song et al., 2018; Zhang et al., 2019; Jin et al., 2020; Zhao et al., 2020; Zheng et al., 2021), such as BUC (Lin et al., 2019), SSG (Fu et al., 2019; Zhai et al., 2020), HCT (Zeng et al., 2020) and SpCL (Ge et al., 2020). Recent works mainly focus on label generation, label refinery, the assistance of extra information, and optimization of representation. BUC (Lin et al., 2019) proposed a bottom-up clustering approach to generate pseudo labels. To reduce pseudo label noise, DCML (Chen et al., 2020) selected credible training samples and MMT (Ge et al., 2020) proposed a mutual learning scheme for better pseudo labels. JVTC (Li & Zhang, 2020) and CycAs (Wang et al., 2020) explore temporal information to refine visual similarity. Contrastive learning with feature memory bank has been widely used in many works to learn more robust representation (Zheng et al., 2021; Chen et al., 2021). SpCL (Ge et al., 2020) progressively generated more reliable clusters for the unified contrastive loss. Cluster Contrast (Dai et al., 2021) proposed to store feature vectors and compute contrast loss in the cluster level. Although great success has been made, this line of methods usually leverage a single model to learn the knowledge that it already has, making it hard to learn sufficient capability due to the lack of diverse discrimination. To alleviate this problem, we propose PEG based on multi-model training where diversity of discrimination can be explored and exploited by the evolution of networks.

2.2 Multiple Model Ensemble

There is a considerable number of previous works on ensembles with neural networks. Explicit ensemble methods often train a series of base-level networks and average the predictions across them as the final result, which have low efficiency during both training and testing (Hansen & Salamon, 1990; Perrone & Cooper, 1992; Krogh & Vedelsby, 1994; Dietterich, 2000; Huang et al., 2017; Lakshminarayanan et al., 2017). Recently, implicit ensemble methods are explored to tackle this problem. A typical approach (Srivastava et al., 2014; Wan et al., 2013; Huang et al., 2016; Singh et al., 2016) generally create a series of networks with shared weights during training and then implicitly ensemble them at test time. Another approach (Shen et al., 2019) focuses on label refin-

ery by distilling and transferring knowledge from a variety of trained networks to a single network for higher discrimination capability. However, these supervised methods cannot be directly used on unsupervised re-ID tasks, especially when the training set and the testing set share non-overlapping label space. On the other hand, existing methods accomplish the ensemble on all base-level networks while they ignore the problem that a very weak base-level network could drag down the overall performance when included. Commonly, “All” is not the “Best”. In this work, we propose a cooperative game in the selection phase of the framework to find and preserve the optimal combination of base-level networks using the unlabeled data and obtain progressive ensemble by an iterative population evolutionary gaming under unsupervised conditions.

2.3 Algorithmic Game Theory

Machine learning methods with multi-agent game are proposed to address various tasks, such as image generation (Goodfellow et al., 2014), attacks and defenses for deep learning (Yuan et al., 2019), playing computer games (Vinyals et al., 2019; Peng et al., 2020), etc. SVM can be considered as a game between two agents where one agent challenges the other to find the best hyper-plane after providing the most difficult points for classification. Generative adversarial networks (GANs) (Goodfellow et al., 2014) train two networks, the discriminator and the generator, against each other in order to generate images that can pass for real data. These methods are designed for non-cooperative games where agents have contrary rewards. However, in this work, the selection of networks is modeled as a multi-agent cooperative game, where rewards are global and shared by all agents. Although methods with cooperative games have been explored for reinforcement learning (Peng et al., 2020), they can not be used for such a computer vision task. Our approach consider the Best-response dynamics in cooperative game theory to solve a Nash equilibrium of model selection strategy.

2.4 Unsupervised Evaluation Metrics of Models

Metrics used in person re-ID always depend on samples with ground truth, such as mean Average Precision (mAP) and Cumulative Match Characteristic (CMC) curve, which are calculated between model prediction and the corresponding ground truth labels. However, these supervised metrics are not available during unsupervised learning when labels of data are unknown, therefore, they cannot be used as the criterion of the model selection in our PEG framework. On the other hand, several unsupervised evaluation metrics which require no data label have been designed to measure the performance of clustering algorithms as internal evaluation

metrics (Davies & Bouldin, 1979; Baker & Hubert, 1975; Hubert & Levin, 1976; Maulik & Bandyopadhyay, 2002; Halkidi et al., 2002). For example, the silhouette coefficient (Rousseeuw, 1987) estimates the average distance between each point in one cluster and points in the nearest neighboring cluster. The Dunn index (Dunn, 1973) calculates the ratio of the minimum of inter-cluster distance to the maximum of intra-cluster distance between samples. Nevertheless, these cluster validations cannot be directly used for the evaluation of re-ID models, for example, by the quality of clustering with their extracted features under the same clustering algorithm. That's because the distribution of feature clusters cannot measure the performance of models, especially in unsupervised settings. For instance, the metrics may estimate well clustering of features even when the model is poor but only is trained to overfit on its inaccurate labels. In this paper, we propose a cross-reference scatter which approximately measures a model's discrimination capability by indirectly estimating its predicted pseudo-labels. It utilizes the pseudo-labels to train a reference network for a few iterations and then observes the cohesion and separation of its feature space to estimate the discrimination of the evaluated model. This method mines the latent visual relationships between image samples and so can approximately estimate models' discrimination on unlabeled data.

2.5 Population-Based Evolutionary Training

Population-based evolution has been widely studied to solve real-valued optimization problems. For distance metric learning, a related task of re-ID, EDML (Fukui et al., 2013) and its variants (Kalintha et al., 2019; Ali et al., 2020) were proposed to optimize a linear or non-linear transformation using differential evolution. However, these approaches cannot address the training of deep neural networks in re-ID due to the large scale of learnable parameters. Our approach is inspired by and built upon another line of Population Based Training (PBT) (Jaderberg et al., 2017), which is originally proposed for optimization of hyperparameters of networks. PBT trains a population of networks and performs periodically a process of exploiting and exploring, leading to automatic learning of the best configurations. It has been proved effective for a suite of challenging problems, including Atari and StarCraft II of reinforcement learning (Vinyals et al., 2019; Jaderberg et al., 2019), training Generative Adversarial Network (GAN) (Jaderberg et al., 2017) and data augmentation (Ho et al., 2019). However, such a population-based training of networks has not been explored in unsupervised conditions, in which the criterion of network selection is difficult to determine. On the other hand, existing PBT approaches follow the principle of best individual selection, while our method selects and preserves optimal groups of networks that are more complementary. We additionally incorporate mutual

learning within the population into the framework, leading to superior performance on the unsupervised re-ID.

3 Methodology

3.1 Population-Based Evolutionary Gaming

Due to the lack of diversity of individual networks, sufficient discrimination for unsupervised person re-ID is difficult to achieve. In contrast to previous works that use a single network for self-training, we propose a PEG that concurrently trains a diverse population of neural networks through an evolutionary game. In our formulation, the population \mathcal{P} contains K networks, each of which is denoted as $\mathcal{M}(\theta, \phi)$. θ is the learnable parameters, and ϕ is its hyper-parameters including the learning rate and loss ratios. The proposed training algorithm consists of three iterative phases, namely, selection to preserve adaptive networks, reproduction and mutation to learn more diversity, and population mutual learning to assemble knowledge, as illustrated in Fig. 2. The procedure of PEG is also described in Algorithm 1.

Algorithm 1 Population-based Evolutionary Gaming

Input: Unlabeled dataset $\{\mathbf{X}\}$.

Input: Initial population \mathcal{P} of K models $\{\mathcal{M}^k\}$ with parameters $\{\theta^k\}$ and hyper-parameters $\{\phi^k\}$, $k = 1, \dots, K$.

Output: The inference model $\mathcal{M}(\theta)$.

```

1: for each generation do
2:   // Selection
3:   Select  $L$  models from the population  $\mathcal{P}$  by the cooperative game
   in Sect. 3.1.1,
    $\{\mathcal{M}^l, l = 1, \dots, L\} = \text{SELECTION}(\mathcal{P}, L)$ .
4:   Update the population  $\mathcal{P} \leftarrow \{\mathcal{M}^l, l = 1, \dots, L\}$ .
5:   // Reproduction & Mutation
6:   for each model  $\mathcal{M}^l$  do
7:     Clone  $H$  models of  $\mathcal{M}^l$ :  $\mathcal{M}_h^l(\theta_h^l, \phi_h^l) = \mathcal{M}^l(\theta^l, \phi^l)$ ,  $h =$ 
      $1, \dots, H$ .
8:     Mutate the hyper-parameters of the cloned models:
      $\phi_h^l \sim \mathbf{U}((1-r)\phi_h^l, (1+r)\phi_h^l)$ ,  $h = 1, \dots, H$ .
9:     Add the cloned model into the population,  $\mathcal{P} \leftarrow \mathcal{P} +$ 
      $\{\mathcal{M}_h^l(\theta_h^l, \phi_h^l)\}$ ,  $h = 1, \dots, H$ 
10:  end for
11:  Update the population size  $K \leftarrow L \times (H + 1)$ 
12:  // Population mutual learning
13:  Optimize parameters of models in  $\mathcal{P}$  by population mutual learning
   in Sect. 3.1.3:
    $\{\theta^k\} \leftarrow \text{PML}(\mathbf{X}, \{\theta^k\})$ ,  $k = 1, \dots, K$ .
14: end for
15: Select a model for inference:  $\mathcal{M}(\theta) = \text{SELECTION}(\mathcal{P}, 1)$ 
16: Return The inference model  $\mathcal{M}(\theta)$ .
```

3.1.1 Selection

Since poor models may drag down the performance in multiple model training, we first propose a selection phase to

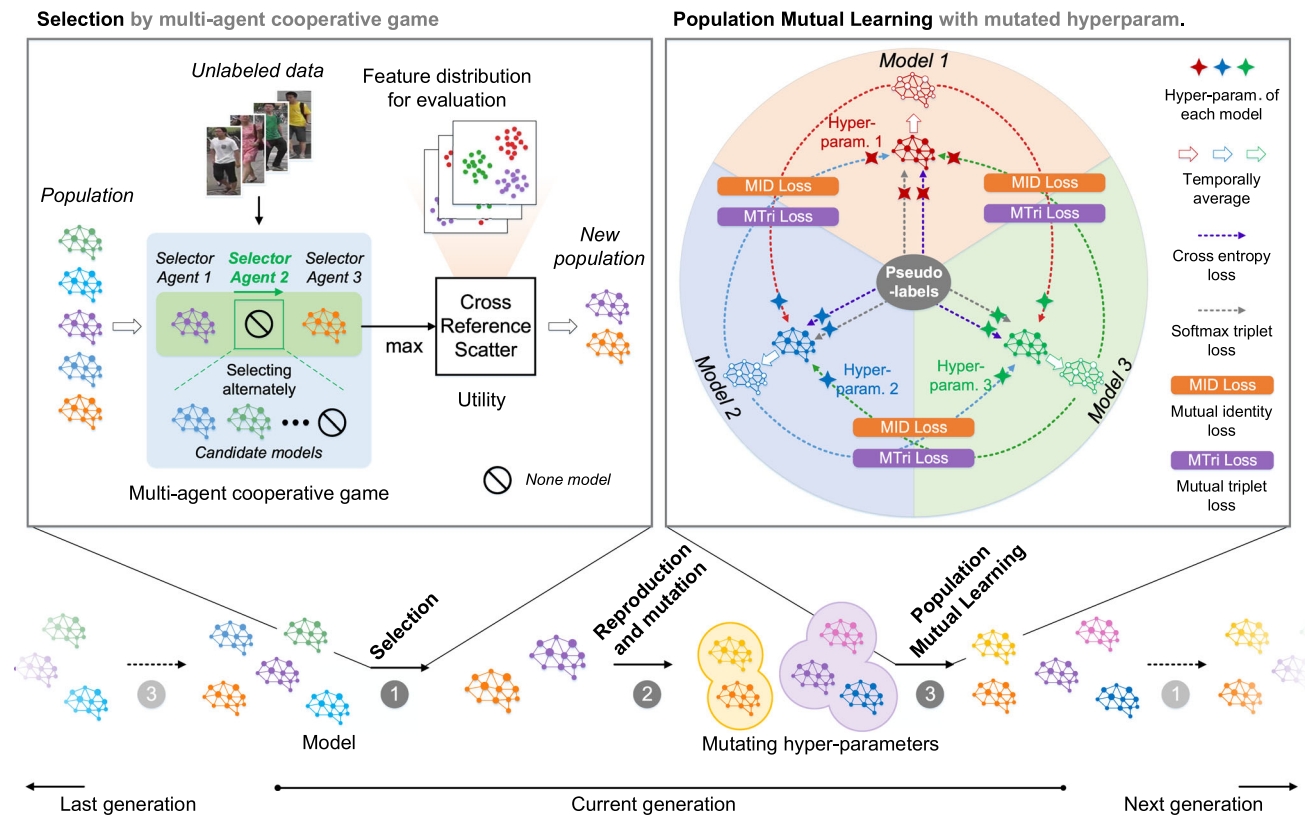


Fig. 2 Population-based evolutionary gaming framework. PEG iteratively performs selection, reproduction and mutation and population mutual learning to learn diverse and discriminative models under unsupervised conditions. In every generation, (1) selection preserves the optimal combination of models through a cooperative game with a set

of selector agents to maximize the utility function (CRS). (2) Reproduction and mutation clone models and fluctuate their hyper-parameters to explore more diversity. (3) Population mutual learning trains models with mutated hyper-parameters by knowledge distillation from each other to enhance and assemble their discrimination

preserve better models in PEG. Given a population \mathcal{P} of K neural network models $\{\mathcal{M}^1, \dots, \mathcal{M}^K\}$, selection aims to find an optimal subset of the population that is more adaptive to the given data, as shown in Fig. 2. Then, networks of the subset are preserved for later training, while other networks are abandoned to reduce the population size. The selection scheme is considered as a multi-agent cooperative game among L selector agents characterized by $(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_L, u)$, where \mathcal{A}_l is the action space of agent l ; and $u : \mathbf{A} \rightarrow \mathbb{R}$ denotes the utility function of the joint action $\mathbf{A} \in \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_L$. The action of each agent $\mathbf{a}_l \in \mathcal{A}_l$ is to select one neural network from the population \mathcal{P} , $\mathcal{A}_l = \{\mathcal{M}_1, \dots, \mathcal{M}_K\}$. The number of agents is restricted due to the limitation of computational resources. In the cooperative game, agents pursue the same goal to maximize their team utility u . To maximize the discrimination and complementarity of the preserved networks, we define the utility function u by the performance of the ensemble model. However, a model’s performance is difficult to estimate without labeled testing data. To address this problem, we design cross-reference scatter J_{cr} to evaluate the ensemble model and consider it the formula of the utility

function, $u(\mathbf{A}) = J_{cr}(\vartheta(\mathbf{A}))$, where ϑ denotes the ensemble model produced by the networks currently selected by the agents. The detailed description of the cross-reference scatter will be provided in Section 3.1.1. Since there are approximately K^L possible action combinations, a global optimal solution is impossible to derive by enumerating all the possibilities. Therefore, we turn to obtain a Nash equilibrium solution $\tilde{\mathbf{A}} = \{\tilde{\mathbf{a}}_l\}$, where each agent attempts to learn the best response to the other agents’ actions:

$$u(\mathbf{a}'_l, \tilde{\mathbf{A}}_{-l}) \leq u(\tilde{\mathbf{A}}), \tag{1}$$

where \mathbf{a}'_l is any unilateral deviation, $\tilde{\mathbf{A}}_{-l} = \{\tilde{\mathbf{a}}_{l'}\}_{l' \neq l}$ and $-l$ represents all agents except \mathbf{a}_l . We solve Eq. 1 via best-response dynamics. Each agent acts in a circular manner until it falls into a Nash equilibrium, where the action of each agent is the best response to the other agents. Below, we provide the detailed procedure of the cooperative selection game.

- (1) Initialization of agent actions: randomly initialize $\mathbf{a}_l \in \mathcal{A}_l$ for $l = 1, \dots, L$.

- (2) For agent l , solve the optimal action of agent \mathbf{a}_l in response to the actions of the other $L - 1$ agents. The objective for optimization of \mathbf{a}_l can be formulated as,

$$\mathbf{a}_l^* = \arg \max_{\hat{\mathbf{a}}_l \in \mathcal{A}_l} u(\hat{\mathbf{a}}_l, \mathbf{a}_{-l}), \tag{2}$$

where \mathbf{a}_{-l} means all agents except \mathbf{a}_l .

- (3) Then update the joint action to $(\mathbf{a}_l^*, \mathbf{a}_{-l})$, as \mathbf{a}_l^* is a best response to \mathbf{a}_{-l} .

$$\mathbf{A} \leftarrow (\mathbf{a}_l^*, \mathbf{a}_{-l}). \tag{3}$$

- (4) Repeat steps 2 to 3 for agent $l + 1$.
 (5) If the joint action \mathbf{A} has not changed in the last $L - 1$ optimization rounds, the utility falls into Nash equilibrium, where every agent implements the best response to all other agents. In this case, we stop the optimization process, preserve the selected networks for the next generation, and abandon the other networks.

Cross-reference Scatter

A core issue is to estimate a model’s performance using unlabeled data in the selection phase of the proposed evolutionary game; however, such a measurement has not been explored. In this study, we propose a cross-reference scatter (CRS) for the approximate evaluation of re-ID models using unlabeled samples. Generally, the pseudo-labels predicted by better networks are more accurate, but the specific accuracy of the labels cannot be directly evaluated without the ground truth. However, models trained by more accurate pseudo-labels tend to achieve larger intra-cluster cohesion and inter-cluster separation in the feature space because incorrect labels will enforce models to separate samples of the same class or aggregate samples of different classes, which is difficult to accomplish. Therefore, it is reasonable to indirectly evaluate a network according to the feature cohesion and separation of a reference model that is trained by pseudo-labels which are predicted with the evaluated model.

First, we introduce an inter-/intra-cluster scatter (ICS) to estimate the separation and cohesion of clusters in the feature space. Although existing metrics such as DBI (Davies & Bouldin, 1979), SC (Rousseeuw, 1987) have been studied to estimate clustering, they usually pay more attention to the hard edge samples of clusters while ignoring the overall distribution, and thus are not applicable to measure re-ID models. Inspired by the objective of linear discriminant analysis, that is, to maximize the ratio of the between-class variance and the within-class variance, the inter-/intra-cluster scatter is defined as the ratio of the inter-cluster variance and intra-cluster variance in the clustered feature space. Given the set of images represented by feature vectors $\mathbf{f}(X|\Theta)$, where Θ denotes the parameters of the feature extractor network,

Algorithm 2 Cross Reference Scatter (CRS)

Input: Unlabeled dataset $\{\mathbf{X}\}$.
Input: Evaluated model Θ .
Input: Reference model θ^{ref} .
Output: CRS of the evaluated model: $J_{cr}(\Theta)$.
 1: Extract features on $\{\mathbf{X}\}$ by the evaluated model Θ : $\mathbf{f}(\mathbf{X}|\Theta)$.
 2: Generate pseudo-labels $\tilde{Y}(\Theta)$ of \mathbf{X} by clustering samples using $\mathbf{f}(\mathbf{X}|\Theta)$.
 3: Train the reference model θ^{ref} with $\{\mathbf{X}, \tilde{Y}(\Theta)\}$ for a fixed number of iterations by optimizing Eq. 12, 14.
 4: Calculate ICS of the reference model θ^{ref} on $\{\mathbf{X}\}$: $J(\mathbf{X}|\theta^{ref})$ by Eq. 7.
 5: $J_{cr}(\Theta) = J(\mathbf{X}|\theta^{ref})$.
 6: **Return** CRS of the evaluated model: $J_{cr}(\Theta)$.

we cluster all samples into M groups as \mathcal{C} . We measure the cohesion of each cluster by the variance of features assigned to it. The intra-cluster scatter of cluster \mathcal{C}_i is defined as

$$S_{intra}^i(X|\Theta) = \sum_{x \in \mathcal{C}_i} [\mathbf{f}(x|\Theta) - \mu_i]^T [\mathbf{f}(x|\Theta) - \mu_i], \tag{4}$$

where $\mu_i = \sum_{x \in \mathcal{C}_i} \mathbf{f}(x|\Theta)/n_i$ is the centroid of cluster \mathcal{C}_i (with n_i samples). Then, the intra-cluster scatter of all clusters is computed as

$$S_{intra}(X|\Theta) = \sum_{i=1}^M S_{intra}^i(X|\Theta). \tag{5}$$

To measure the separation of feature clusters, the inter-cluster scatter is defined as the variance of cluster centroids

$$S_{inter}(X|\Theta) = \sum_{i=1}^M n_i [\mu_i - \mu]^T [\mu_i - \mu], \tag{6}$$

where $\mu = \sum_{n=1}^N \mathbf{f}(x_n|\Theta)/N$ is the center of the entire dataset. Considering both the separation and cohesion of feature clusters, the inter-/intra-cluster scatter $J(X|\Theta)$ is defined as the ratio of the inter-cluster scatter and intra-cluster scatter

$$J(X|\Theta) = S_{inter}(X|\Theta)/S_{intra}(X|\Theta). \tag{7}$$

$J(X|\Theta)$ increases when the inter-cluster scatter is larger and the intra-cluster scatter is smaller, which entails larger separation and cohesion within feature clusters.

Utilizing inter-/intra-cluster scatter (ICS), we attempt to evaluate a model by indirectly estimating its predicted pseudo-labels in a cross-reference (CR) manner. Given a network model with parameter Θ for evaluation, we first implement the model to extract the convolutional features of all samples $\mathbf{f}(X|\Theta)$. Then, minibatch k-means clustering is performed on $\mathbf{f}(X|\Theta)$ to classify all samples into M different clusters. After clustering, the produced cluster IDs are used

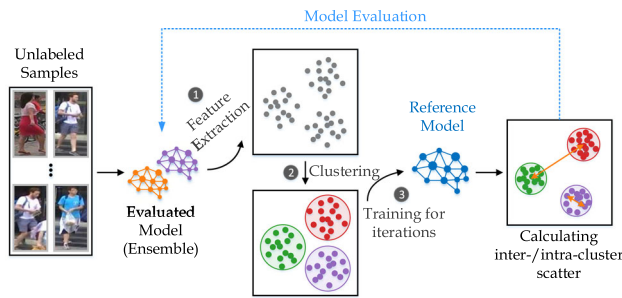


Fig. 3 Illustration of evaluation scheme of the proposed cross-reference scatter (CRS), which approximately measures a model’s discrimination capability by the inter-/intra-cluster scatter of a reference model that is briefly trained using pseudo-labels predicted by the evaluated model

as pseudo-labels $\tilde{Y}(\Theta)$ for samples X . To estimate the accuracy of the predicted pseudo-labels $\tilde{Y}(\Theta)$, we adopt them as supervision to train a reference model with parameters θ^{ref} by optimizing the cross-entropy loss with label smoothing and the softmax triplet loss for a certain number of iterations, and then measure the separation and cohesion of the reference model by computing the inter-/intra-cluster scatter $J(X|\theta^{ref})$ as cross reference scatter (CRS) $J_{cr}(\Theta)$. The value of CRS is then used for model evaluation, where a larger CRS indicates better discrimination capability of the evaluated model Θ . The evaluation scheme is illustrated in Fig. 3 and the detailed process is shown in Algorithm 2.

Importantly, we use the **k-means** clustering because the fair comparison of CRS among models requires the same cluster number during clustering. Specifically, since CRS is defined by the ratio of intra-cluster variance and inter-cluster variance, it is relative to the cluster numbers. For example, a larger cluster number may lead to a larger CRS value due to the smaller intra-cluster variances. And the cluster numbers by other clustering algorithms, i.e. **DBSCAN**, with different evaluated feature models are likely to be different, making it unfair to compare their CRS for model selection.

For fast and fair evaluation, we adopt a slight network, OSNet, with the same initial parameters as the reference model to evaluate different models. The number of training iterations of it is set to a small value from 500 to 1000 according to the number of samples.

3.1.2 Reproduction and Mutation

Reproduction and mutation provide more diversity within the population by reproducing networks and mutating their hyper-parameters, including the learning rate and loss ratios after selection. In the reproduction and mutation phase, each network reproduces multiple descendants, one of which preserves the original hyper-parameters while the others apply a stochastic disturbance to their hyper-parameters to attempt to learn different information and increase the diversity of the population. Specifically, the mutated hyper-parameters ϕ' are

Algorithm 3 Population Mutual Learning (PML)

Input: Unlabeled dataset $\{X\}$. Population \mathcal{P} of K models parameterized by $\{\theta^k\}$, $k = 1, \dots, K$.

Output: Updated neural network parameters $\{\theta^k\}$.

- 1: **for** each epoch **do**
- 2: Extract ensemble features on $\{X\}$ by combinatorial model $\vartheta(\mathcal{P})$: $\mathbf{f}(X|\vartheta(\mathcal{P})) = [\mathbf{f}(X|\theta^1); \dots; \mathbf{f}(X|\theta^K)]$.
- 3: Generate pseudo-labels \tilde{Y} of X by clustering samples using $\mathbf{f}(X|\vartheta(\mathcal{P}))$.
- 4: **for** each iteration t , mini-batch $\mathcal{B} \subset X$ **do**
- 5: Randomly sample S networks $\{\theta^{k_s}\} \subset \{\theta^k\}$, each indexed by k_s , $s = 1, \dots, S$.
- 6: Calculate soft-labels from temporally average model of each sampled network with $\{\theta_T^{k_s}\}$: $\mathbf{p}(x_{i \in \mathcal{B}}|\theta_T^{k_s})$, $\mathcal{P}_{i \in \mathcal{B}}(\theta_T^{k_s})$
- 7: Calculate output of each current model with $\{\theta^{k_s}\}$: $\mathbf{p}(x_{i \in \mathcal{B}}|\theta^{k_s})$, $\mathcal{P}_{i \in \mathcal{B}}(\theta^{k_s})$.
- 8: Update parameters $\{\theta^{k_s}\}$ by optimizing Eq. 16.
- 9: Update temporally average model weights $\{\theta_T^{k_s}\}$ following Eq. 8.
- 10: **end for**
- 11: **end for**
- 12: **Return** Networks parameters $\{\theta^k\}$, $k = 1, \dots, K$.

sampled from a uniform distribution $\mathbf{U}((1-r)\phi, (1+r)\phi)$ that fluctuates within r of the original value. The steps 5–11 in Algorithm 1 summarize the process of reproduction and mutation. Note that the mutation does not immediately change the weight parameters of neural networks. Changes occur to them when networks are trained by their mutated hyper-parameters in mutual learning.

3.1.3 Population Mutual Learning

After mutation, population mutual learning is performed among networks in the population \mathcal{P} to access and assemble diverse discrimination capability using unlabeled data in an iteratively collaborative way, as shown in Fig. 2. Each network accomplishes learning from the whole population by means of its own hyper-parameters acquired from mutation. The learning scheme consists of a clustering-based pseudo-label prediction procedure and a mutual feature learning procedure. In each iterative epoch, pseudo-labels are first predicted for all samples via clustering and then utilized to fine-tune the networks of the population. In this phase, networks learn representations of images in two ways: from the shared pseudo-labels predicted by the whole population via clustering and from the output of other networks as soft labels via knowledge distillation. The procedure of this population mutual learning is described in Algorithm 3.

Pseudo-label prediction. Pseudo-labels are predicted at the beginning of each iterative epoch. In order to predict reliable pseudo-labels, the framework utilizes all networks in the population $\{\mathcal{M}^1, \dots, \mathcal{M}^K\}$ jointly as a combinatorial model $\vartheta(\mathcal{P})$ to extract features for sample clustering. The clustering-based pseudo-label prediction procedure consists

of three steps in total: (1) First, ensemble features of unlabeled samples $\{\mathbf{X}\}$ are obtained by concatenating features that are individually extracted by all networks, $\mathbf{f}(\mathbf{X}|\vartheta(\mathcal{P})) = [\mathbf{f}(\mathbf{X}|\theta^1); \dots; \mathbf{f}(\mathbf{X}|\theta^k)]$; (2) Then, DBSCAN (Ester et al., 1996) clustering is performed on $\mathbf{f}(\mathbf{X}|\vartheta(\mathcal{P}))$ to classify all unlabeled samples into M different clusters. (3) The produced cluster IDs are used as pseudo-labels \tilde{Y} for the training samples \mathbf{X} . The steps 2 and 3 in **Algorithm 3** summarize this clustering process.

Different from CRS, we use DBSCAN here for model learning to generate more accurate pseudo-labels, since DBSCAN has been proven effective and efficient for a lot of clustering-based unsupervised person re-identification (Ge et al., 2020; Chen et al., 2021). Compared with the k-means cluster algorithm, DBSCAN mines sample relations more accurately according to their density without setting the number of clusters and then helps learn more discriminative models.

Mutual feature learning Utilizing the predicted pseudo-labels, the framework aims to organize networks within the population to learn from each other and enhance themselves in a mutual learning manner, as shown in Fig. 2. In each training iteration, the same batch of images with different random augmentations is first fed to all the networks in the population parameterized by $\{\theta^k\}$ to predict the classification confidence predictions $\{\mathbf{p}(x_n|\theta^k)\}$ and feature representations $\{\mathbf{f}(x_n|\theta^k)\}$. The classification confidence predictions are computed by a linear transformation of the feature representations followed by a softmax function. To transfer knowledge from one network to others, the outputs of each network serve as soft labels for training other networks. However, directly using the current predictions as soft labels to train each model decreases the independence of the model outputs, which might result in error amplification. To avoid this issue, the temporally averaged model (Tarvainien & Valpola, 2017) of each network, which preserves more original knowledge, is used to generate reliable soft pseudo-labels for supervision. The parameters of the temporally averaged model of network θ^k at current iteration t are denoted as Θ_t^k , which is updated as

$$\Theta_t^k = \alpha \Theta_{t-1}^k + (1 - \alpha)\theta^k, \tag{8}$$

where $\alpha \in [0, 1]$ is the scale factor, and the initial temporal average parameters are $\Theta_0^k = \theta^k$. For each network \mathcal{M}^k , three loss functions are computed as optimization objectives: mutual identity loss, mutual triplet loss and voting loss. The mutual identity loss (Zhang et al., 2018) of models learned by a certain network \mathcal{M}^e is defined as the cross entropy between the ID prediction of the student network \mathcal{M}^k and the teacher

network \mathcal{M}^e

$$\mathcal{L}_{mid}^{k \leftarrow e} = -\frac{1}{N} \sum_{n=1}^N \mathbf{p}(x_n|\Theta^e)^T \log \mathbf{p}(x_n|\theta^k). \tag{9}$$

The mutual triplet loss (Ge et al., 2020) of models learned by a certain network \mathcal{M}^e is defined as the binary cross entropy

$$\mathcal{L}_{mtri}^{k \leftarrow e} = -\frac{1}{N} \sum_{n=1}^N \left[\mathcal{P}_n(\Theta^e) \log \mathcal{P}_n(\theta^k) + (1 - \mathcal{P}_n(\Theta^e)) \log(1 - \mathcal{P}_n(\theta^k)) \right], \tag{10}$$

where $\mathcal{P}_n(\theta^k)$ denotes the softmax of the feature distance between negative sample pairs

$$\mathcal{P}_n(\theta^k) = \frac{e^{\|\mathbf{f}(x_n|\theta^k) - \mathbf{f}(x_{n-}|\theta^k)\|}}{e^{\|\mathbf{f}(x_n|\theta^k) - \mathbf{f}(x_{n+}|\theta^k)\|} + e^{\|\mathbf{f}(x_n|\theta^k) - \mathbf{f}(x_{n-}|\theta^k)\|}}, \tag{11}$$

where x_{n+} denotes the hardest positive sample of anchor x_n according to the pseudo-labels \tilde{Y} and x_{n-} denotes the hardest negative sample. $\|\cdot - \cdot\|$ denotes L_2 distance.

To learn stable and discriminative knowledge from the pseudo-labels obtained by clustering, we introduce voting loss, which consists of the classification loss and triplet loss. For each model \mathcal{M}^k , the classification loss is defined as the cross entropy with label smoothing (Szegedy et al., 2016)

$$\mathcal{L}_{id}^k = -\frac{1}{N} \sum_{n=1}^N \tilde{\mathbf{q}}^T \log \mathbf{p}(x_n|\theta^k), \tag{12}$$

where $\tilde{\mathbf{q}}$ is the smoothing label according to pseudo-labels \tilde{Y} . Each element is calculated by

$$\tilde{q}_j = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{M} & j = \tilde{y}_n \\ \frac{\varepsilon}{M} & j \neq \tilde{y}_n \end{cases}, \tag{13}$$

The softmax triplet loss is defined as:

$$\mathcal{L}_{tri}^k = -\frac{1}{N} \sum_{n=1}^N \log \frac{e^{\|\mathbf{f}(x_n|\theta^k) - \mathbf{f}(x_{n-}|\theta^k)\|}}{e^{\|\mathbf{f}(x_n|\theta^k) - \mathbf{f}(x_{n+}|\theta^k)\|} + e^{\|\mathbf{f}(x_n|\theta^k) - \mathbf{f}(x_{n-}|\theta^k)\|}} \tag{14}$$

where x_{n+} denotes the hardest positive sample of anchor x_n according to the pseudo-labels and x_{n-} denotes the hardest negative sample. The voting loss is defined by summarizing the classification loss and the triplet loss

$$\mathcal{L}_{vot}^k = w_{id} \mathcal{L}_{id}^k + w_{tri} \mathcal{L}_{tri}^k, \tag{15}$$

where w_{id} and w_{tri} are the loss ratios. For each model \mathcal{M}^k , the overall optimized objective is defined by

$$\mathcal{L}^k = \frac{1}{K-1} (w_{mid} \sum_{e \neq k}^K \mathcal{L}_{mid}^{k \leftarrow e} + w_{mtri} \sum_{e \neq k}^K \mathcal{L}_{mtri}^{k \leftarrow e}) + \mathcal{L}_{tot}^k. \quad (16)$$

Each model is trained by its own hyper-parameters $\phi = \{\varepsilon, w_{id}, w_{tri}, w_{mid}, w_{mtri}\}$ to explore different information. In addition, direct descendants of the same networks do not learn from each other in the mutual learning phase since they acquire similar knowledge. Note that training of a large-size population requires unaffordable computational resources. To address this problem, we use a random sampling strategy of networks. Specifically, for each batch of data, mutual learning is performed only on a randomly sampled subset of networks, as shown in step 5 in **Algorithm 3**.

3.2 Analysis of Escaping Capacity

Here we analyze the escaping capacity from the local optimum of our approach. The optimization of our approach can be formulated into two interactive phases. The first one is to optimize the label assignment of samples according to feature models

$$\arg \min_{\tilde{Y}} f_a(\tilde{Y}, \Theta, X), \quad (17)$$

where \tilde{Y} is the assigned labels and f_a is the objective loss function of the phase which is determined by the used clustering algorithm. Θ and X denote the model parameters and input samples, respectively. The second phase is to optimize the parameters of feature models according to the label assignment

$$\arg \min_{\Theta} f_m(\Theta, \tilde{Y}, X|\phi). \quad (18)$$

$f_m(\cdot|\phi)$ is loss function to train the models Θ as Eq. 16, and ϕ is the hyper-parameters. The two optimization phases interact as a two-agent game and the *local optimum* occurs when the game halts at a Nash equilibrium:

$$\begin{aligned} & \exists (\tilde{Y}^*, \Theta^*), \\ & s.t. \\ & \tilde{Y}^* = \arg \min_{\tilde{Y}} f_a(\tilde{Y}, \Theta^*, X), \\ & \Theta^* = \arg \min_{\Theta} f_m(\Theta, \tilde{Y}^*, X|\phi). \end{aligned} \quad (19)$$

In our approach, function $f_m(\cdot|\phi)$ will be changed when the hyper-parameters ϕ are mutated to new values ϕ' , leading to the shift of the local optimal model parameters Θ^* . Therefore the Nash equilibrium between \tilde{Y} and Θ will be broken, and

the local optimum at (\tilde{Y}^*, Θ^*) will not exist exactly since the mutation changes the condition of the Nash equilibrium.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate the proposed method on three large-scale person re-identification benchmarks including Market-1501 (Zheng et al., 2015), DukeMTMC-reID (Ristani et al., 2016; Zheng et al., 2017) and MSMT17 (Wei et al., 2018).

Market-1501 This dataset contains 32,668 images of 1,501 identities from 6 disjoint cameras, among which 12,936 images from 751 identities form a training set, 19,732 images from 750 identities (plus a number of distractors) form a gallery set, and 3,368 images from 750 identities form a query set.

DukeMTMC-reID This dataset is a subset of the DukeMTMC. It consists of 16,522 training images, 2228 query images, and 17,661 gallery images of 1812 identities captured using 8 cameras. Of the 1812 identities, 1404 appear in at least two cameras and the rest (distractors) appear in a single camera.

MSMT17 contains 126,441 images of 4,101 IDs captured from a 15-camera network. The training set has 32,621 images of 1,041 identities, and the testing set has 93,820 images of 3,060 identities. During inference, 11,659 images are selected as query and the other 82,161 images are used as gallery from the testing set.

Evaluation Metrics: We use the Cumulative Matching Characteristic (CMC) curve and mean average precision (mAP) for performance evaluations and comparisons.

4.2 Implementation Details

Model settings. We adopt eight models with architectures of similar-weight parameters to initialize the population, including DenseNet-121 (Huang et al., 2017), DenseNet-169, IBN-DenseNet-121 (Pan et al., 2018), IBN-DenseNet-169, Inception-v3 (Szegedy et al., 2016), ResNet-50, IBN-ResNet-50a and IBN-ResNet-50b. All model are pretrained using ImageNet (Deng et al., 2009). In every model, the convolutional feature output by the last pooling layer is used for image representation.

PEG settings. The maximum size of networks in the selection phase L is set to 3 for experiments. A lightweight OSNet (Zhou et al., 2019) is used as the reference model of CRS for faster training. In addition, we conduct minibatch k-means clustering for CRS, and the number of clusters M is set to 500 for Market-1501 and DukeMTMC-reID following MMT (Ge et al., 2020). In the reproduction and mutation phase, each network reproduces 3 networks with mutation

factor $r = 0.5$. The whole population evolves for 3 generations in total. Our method is trained on 4 GPUs under PyTorch framework. During testing, we use only one network which is selected by CRS in the population for feature representations.

Training settings. In mutual learning, we calculate k-reciprocal Jaccard distance (Zhong et al., 2017) for clustering, where k_1, k_2 are set to 6 and 30, respectively. We set the minimum cluster samples to 4 and a distance threshold to 0.6 for DBSCAN (Ester et al., 1996). During training, the input image is resized to 256×128 , and traditional image augmentation is performed via random flipping and random erasing. For each class from the training set, a mini-batch of 256 is sampled with $P = 16$ randomly selected classes and $K = 16$ randomly sampled images for computing the hard batch triplet loss. We use the Adam (Kingma & Ba, 2014) with weight decay 0.0005 to optimize parameters. In population mutual learning, the learning rate is fixed to 0.00035 for the overall 15 epochs. In each epoch, the temporal ensemble momentum α in Eq. 8 is set to 0.999.

4.3 Comparison with State-of-the-Arts

We compare PEG with state-of-the-art person re-ID methods in Tables 1 and 2 on Market-1501, DukeMTMC-reID and MSMT17 datasets, respectively. The performance of our full approach is reported as *PEG(Full)*. In addition, we also evaluate PEG using the same backbone of ResNet50 as most of the other methods since backbones are important for feature learning. However, the backbones of models are automatically selected in our selection phase. To guarantee a model with ResNet50 is preserved in the population, we limit PEG to choose at least one ResNet50 network at every time of selection. The results tested by ResNet50 are reported as *PEG/ResNet50*.

Previous unsupervised methods can be categorized into unsupervised domain adaptation (UDA) and fully unsupervised (FU) methods. State-of-the-art UDA methods are first listed and compared in Tables 1 and 2, including MMCL (Wang & Zhang, 2020), JVTC (Li & Zhang, 2020), DG-Net++ (Zou et al., 2020), ECN++ (Zhong et al., 2020), AD-Cluster (Zhai et al., 2020), MMT (Ge et al., 2020), DCML (Chen et al., 2020), MEB-Net (Zhai et al., 2020), MetaCam-DSCE (Yang et al., 2021), SpCL (Ge et al., 2020) and GLT (Zheng et al., 2021). All these methods usually rely on an annotated source domain to provide basic discrimination and transfer it to the target domain. Without any identity annotation from source domains, our proposed PEG outperforms all of them on Market-1501, DukeMTMC-reID datasets, and most of them on MSMT17 dataset except SpCL. The results indicate the better capacity of PEG to explore the information of the unlabeled data by exploiting the diversity of multiple models. On the other hand, although other

approaches have also been proposed to utilize multiple models, such as MMT and MEB-Net, our PEG still surpasses them by exploring and exploiting the diversity of multiple models through evolutionary gaming. With mutation to provide more diverse discrimination, it automatically finds and preserves the optimal combination of networks from the population in every generation and thus achieves better performance in the end.

State-of-the-art fully unsupervised methods are then listed and compared in Tables 1 and 2 including BUC (Lin et al., 2019), SSL (Lin et al., 2020), JVTC (Li & Zhang, 2020), MMCL (Wang & Zhang, 2020), MPRD (Ji et al., 2021), HCT (Zeng et al., 2020), CycAs (Wang et al., 2020), GCL (Chen et al., 2021), UGA (Wu et al., 2019), IICS (Xuan & Zhang, 2021), IN unsup. (Fu et al., 2021), SpCL (Ge et al., 2020), OPLG (Zheng et al., 2021), CAP (Wang et al., 2020), ICE (Chen et al., 2021) and ClusterContrast (Dai et al., 2021). Especially, ICE (aware) denotes using extra camera information, and ICE (agnostic) denotes not using it. The compared approaches mainly rely on the pseudo-label discovery of single networks. Among them, methods tagged by “*” denote that elaborate extra temporal information is additionally used to improve the discrimination, such as CycAs and UGA, while our approach only considers person appearance similarity. The performance of these methods is provided just for reference since it is not our point to explore the extra temporal information, and our method does not use any of them. The fully unsupervised methods are separated into two groups, including linear classifier based methods and memory bank based methods:

(1) Comparison with linear classifier based methods

For the linear classifier based methods, our approach with ResNet50 achieves better performance than most of them only except CycAs and UGA on the MSMT17 dataset, as shown in Tables 1, and 2. Different from the other two datasets, CycAs and UGA with extra temporal information achieves better performance on MSMT17 because images in the dataset are more diverse and harder to cluster accurately, making the elaborate extra temporal information particularly important. Nevertheless, these methods still suffer from the lack of diversity in single model training, which prevented them from maximizing their discrimination under unsupervised conditions. The superior performance of PEG can be attributed to the multiple model training, which improves the networks’ discriminative capability by mutual learning among diverse networks. And it can also be attributed to the selection of PEG, which preserves the more discriminative models in every generation and achieves the better performance of them. In addition, our full approach PEG(Full) further improves the re-ID performance by automatically selecting better architectures.

(2) Comparison with memory bank based methods

Memory banks (Ge et al., 2020) were employed in many

Table 1 Comparison with person re-identification state-of-the-art methods on Market-1501 and DukeMTMC-reID datasets

Methods	Market-1501				DukeMTMC-reID			
	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10
<i>Unsupervised domain adaptation</i>								
MMCL (Wang & Zhang, 2020)	60.4	84.4	92.8	95.0	51.4	72.4	82.9	85.0
JVTC (Li & Zhang, 2020)	61.1	83.8	93.0	95.2	56.2	75.0	85.1	88.2
DG-Net++ (Zou et al., 2020)	61.7	82.1	90.2	92.7	63.8	78.9	87.8	90.4
ECN++ (Zhong et al., 2020)	63.8	84.1	92.8	95.4	54.4	74.0	83.7	87.4
AD-Cluster (Zhai et al., 2020)	68.3	86.7	94.4	96.5	54.1	72.6	82.5	85.5
MMT (Ge et al., 2020)	71.2	87.7	94.9	96.9	65.1	78.0	88.8	92.5
DCML (Chen et al., 2020)	72.6	87.9	95.0	96.7	63.3	79.1	87.2	89.4
MEB-Net (Zhai et al., 2020)	76.0	89.9	96.0	97.5	66.1	79.6	88.3	92.2
MetaCam-DSCE (Yang et al., 2021)	76.5	90.1	–	–	65.0	79.5	–	–
SpCL (Ge et al., 2020)	77.5	89.7	96.1	97.6	–	–	–	–
GLT (Zheng et al., 2021)	79.5	92.2	96.5	97.8	69.2	82.0	90.2	92.8
<i>Fully unsupervised—linear classifier based</i>								
LOMO (Liao et al., 2015)	8.0	27.2	41.6	49.1	4.8	12.3	21.3	26.6
Bow (Zheng et al., 2015)	14.8	35.8	52.4	60.3	8.3	17.1	28.8	34.9
UMDL (Peng et al., 2016)	12.4	34.5	52.6	59.6	7.3	18.5	31.4	37.6
BUC (Lin et al., 2019)	29.6	61.9	73.5	78.2	22.1	40.4	52.5	58.2
SSL (Lin et al., 2020)	37.8	71.7	83.8	87.4	28.6	52.5	63.5	68.9
JVTC (Li & Zhang, 2020)	41.8	72.9	84.2	88.7	42.2	67.6	78.0	81.6
MMCL (Wang & Zhang, 2020)	45.5	80.3	89.4	92.3	40.2	65.2	75.9	80.0
MPRD (Ji et al., 2021)	51.1	83.0	91.3	93.6	43.7	67.4	78.7	81.8
HCT (Zeng et al., 2020)	56.4	80.0	91.6	95.2	50.7	69.6	83.4	87.4
*CycAs (Wang et al., 2020)	64.8	84.8	–	–	60.1	77.9	–	–
GCL (Chen et al., 2021)	66.8	87.3	93.5	95.5	62.8	82.9	87.1	88.5
*UGA (Wu et al., 2019)	70.3	87.2	–	–	53.3	75.0	–	–
IICS (Xuan & Zhang, 2021)	72.1	88.8	95.3	96.9	59.1	76.9	86.1	89.8
IN unsup. (Fu et al., 2021)	72.4	87.8	–	–	64.9	80.3	–	–
PEG/ResNet50	82.8	92.8	97.5	98.7	70.4	82.2	90.8	93.6
PEG(Full)	84.3	93.7	97.8	98.5	71.9	83.8	91.2	93.5
<i>Fully unsupervised—memory bank based</i>								
SpCL (Ge et al., 2020)	73.1	88.1	95.1	97.0	65.3	81.2	90.3	92.2
OPLG (Zheng et al., 2021)	78.1	91.1	96.4	97.7	65.6	79.8	88.6	91.6
ICE(agnostic) (Chen et al., 2021)	79.5	92.0	97.0	98.1	67.2	81.3	90.1	93.0
ClusterContrast (Dai et al., 2021)	82.6	93.0	97.0	98.1	72.8	85.7	92.0	93.5
PEG+CCL/ResNet50	83.3	93.4	97.3	98.4	74.4	84.6	92.1	94.0
PEG+CCL(Full)	87.1	94.6	98.0	98.8	76.8	86.4	93.1	95.0
CAP (Wang et al., 2020)	79.2	91.4	96.3	97.7	67.3	81.1	89.3	91.8
ICE(aware) (Chen et al., 2021)	82.3	93.8	97.6	98.4	69.9	83.3	91.5	94.1
PEG+ICE/ResNet50	83.3	94.1	97.8	98.5	71.0	84.4	92.0	94.3
PEG+ICE(Full)	84.5	94.3	98.0	98.5	72.8	85.3	92.5	94.3

“*”Denotes the methods using extra temporal information. *PEG (Full)* denotes the overall performance of our approach. For a fair comparison, *PEG/ResNet50* is tested with the same ResNet50 backbone as most compared methods. *PEG+CCL* and *PEG+ICE* denote training with ClusterContrast (Dai et al., 2021) and ICE (Chen et al., 2021) as baselines, respectively. The performance of our approach is highlighted with bold fonts

recent methods (Dai et al., 2021) to replace the linear classifier before the softmax cross-entropy loss function to improve unsupervised re-ID performance. Specifically, memory bank based methods can be further categorized into two groups including i) ClusterContrast, SpCL, and OPLG that learn general memories for all cameras, and ii) ICE and CAP that design specific memories for each camera. Since our research mainly focuses on the problem of training multiple models, it is independent of these methods for training single models. And they are not contradictory with our main contribution and are compatible with our method. To verify this, we additionally report our performance on the two typical stronger baselines of ClusterContrast as *PEG+CCL*, and ICE as *PEG+ICE* respectively in Tables 1 and 2.

For the camera-general memory bank based methods, *PEG+CCL/ResNet50* surpasses most of other state-of-the-art methods only except ClusterContrast for Rank-1 accuracy on DukeMTMC-reID. For ClusterContrast, the relatively poor Rank-1 on DukeMTMC-reID dataset shows its weakness for some hard negative samples which were mistakenly identified as the same persons, because the soft mutual losses in mutual learning lack the certainty of labels and may not learn strong capability to separate hard negative samples. However, robust improvement of PEG is mainly shown by other metrics, especially the higher mAP on all benchmarks, indicating that PEG deals better with those hard positive samples, which is more important for the practical application of security. Furthermore, our full approach of PEG + CCL (Full) produces a new state-of-the-art performance on Market1501 and DukeMTMC-reID. The better performance can be attributed to the fact that the diverse population provides more reliable supervision for each other. The improved results also demonstrate that our evolution gaming approach is easily combined with different loss functions and can be further improved by more effective losses.

For the camera-specific memory bank based methods, *PEG+ICE/ResNet50* outperforms all the compared methods on the three datasets and produces a new state-of-the-art performance on MSMT17 dataset. The superior performance to the PEG+CCL on MSMT17 can be attributed to that camera-specific memories alleviate the strong camera variance in the dataset which has 15 cameras. However, camera-specific memories are complementary with our PEG framework and can be further improved for better performance.

4.4 Ablation Study

4.4.1 Evaluation of Components

Detailed ablation studies are performed to evaluate the components of PEG as shown in Table 3.

Effectiveness of multiple model training Multiple model training usually achieves better performance than single

Table 2 Comparison with person re-identification state-of-the-art methods on MSMT17 dataset

Methods	MSMT17			
	mAP	R-1	R-5	R-10
<i>Unsupervised domain adaptation</i>				
ECN (Zhong et al., 2020)	10.2	30.2	41.5	46.8
MMT (Ge et al., 2020)	24.0	50.1	63.5	69.3
SpCL (Ge et al., 2020)	26.8	53.7	65.0	69.8
<i>Fully unsupervised—linear classifier based</i>				
MMCL (Wang & Zhang, 2020)	11.2	35.4	44.8	49.8
TAUDL (Li et al., 2018)	12.5	28.4	–	–
UTAL (Li et al., 2019a)	13.1	31.4	–	–
IICS (Xuan & Zhang, 2021)	18.6	45.7	57.7	62.8
*UGA (Wu et al., 2019)	21.7	49.5	–	–
*CycAs (Wang et al., 2020)	26.7	50.1	–	–
PEG/ResNet50	24.5	48.4	61.5	67.5
PEG(Full)	30.9	57.9	69.7	74.5
<i>Fully unsupervised—memory bank based</i>				
SpCL (Ge et al., 2020)	19.1	42.3	55.6	61.2
ICE(agnostic) (Chen et al., 2021)	29.8	59.0	71.7	77.0
ClusterContrast (Dai et al., 2021)	27.6	56.0	66.8	71.5
PEG+CCL/ResNet50	33.4	61.3	73.4	77.8
PEG+CCL(Full)	41.8	69.1	79.5	82.9
CAP (Wang et al., 2020)	36.9	67.4	78.0	81.4
ICE(aware) (Chen et al., 2021)	38.9	70.2	80.5	84.4
PEG+ICE/ResNet50	42.1	72.0	82.0	85.4
PEG+ICE(Full)	44.9	73.9	83.2	86.3

“*”Denotes the methods using extra temporal information. *PEG (Full)* denotes the overall performance of our approach. For a fair comparison, *PEG/ResNet50* is tested with the same ResNet50 backbone as most compared methods. *PEG+CCL* and *PEG+ICE* denote training with ClusterContrast (Dai et al., 2021) and ICE (Chen et al., 2021) as baselines, respectively. The performance of our approach is highlighted with bold fonts

model training because of the complementary discrimination of different models. In this section, we first introduce a baseline multi-model ensemble without mutual learning for comparison that only uses voting loss in Eq. 15 to train networks jointly, denoted as *Multi-model*. With eight networks used for ensemble, pseudo-labels are predicted by concatenating the features outputted from all networks and then used to supervise the training of each network individually by optimizing the voting loss. We also report the result of the *single model baseline* using the best architecture, ResNet50-IBNa. As shown in Table 3, *Multi-model* outperforms the single model training by large margins, indicating that more accurate pseudo-labels can be predicted using multiple models.

Effectiveness of population mutual learning Population mutual learning conducts knowledge distillation among all base models for the better ensemble. Compared with the baseline ensemble as *Multi-model*, models achieve better performance with mutual learning among themselves, as *Multi-*

Table 3 Ablation studies of PEG using eight initial networks under unsupervised conditions

Methods	Market-1501				DukeMTMC-reID			
	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10
Single model baseline	69.6	84.9	92.9	94.9	55.9	72.8	81.9	85.4
Multi-model	76.4	89.7	95.6	97.1	63.4	79.0	87.0	90.0
Multi-model + PML	78.5	90.4	96.1	97.6	66.1	80.3	88.3	91.2
Multi-model + PML + Sel.	79.2	90.9	96.3	97.5	66.9	80.8	88.8	91.6
Multi-model + PML + Sel. + Rep.&Mut.	84.3	93.7	97.8	98.5	71.9	83.8	91.2	93.5
Single architecture + PEG	80.1	90.9	96.3	97.4	69.4	82.1	90.3	92.9

Single model baseline denotes the best performance of single model training using self-improvement mechanism. *Multi-model* means that all models are used for pseudo label prediction and every model is then trained individually. *PML* denotes population mutual learning. *Sel.*, *Rep.* and *Mut.* denote selection, reproduction and mutation in PEG framework, respectively. *Single architecture + PEG* denotes the population is initialized with a single model

model + PML in Table 3. For example, the mAP is improved by 2.1% and 2.7% on Market-1501 and DukeMTMC-reID, respectively. The improvement can be attributed to that models additionally learn the distribution predicted by other models which contain more discriminative information.

In addition, more detailed ablation studies are performed to evaluate the components of mutual learning as shown in Table 5. In this experiment, three networks (DenseNet-121-IBNa, DenseNet-169-IBNa, and ResNet-50-IBNa) are trained concurrently. We first validate the temporally average model by removing it, denoted as *PML w/o Θ_T* . For this experiment, we directly use the prediction of the current networks parameterized by θ_T instead of the temporally average networks with parameters Θ_T as soft labels. As Table 5 shows, distinct drops are observed, indicating that networks tend to degenerate to be homogeneous without using temporally average models, which substantially decreases the learning capability. Then we evaluate the mutual loss in Sect. 3.1.3 from two aspects: the mutual identity loss and the mutual triplet loss. The former is denoted as *PML w/o \mathcal{L}_{mid}* . Results show that mAP drops from 79.2 to 77.2% on Market-1501 dataset and from 66.9 to 65.1% on DukeMTMC-reID dataset. Similar drops can also be observed when studying the mutual triplet loss, which is denoted as *PML w/o \mathcal{L}_{mtri}* . The effectiveness of the mutual learning, including both two mutual losses, can be largely attributed to that it enhances the discrimination capability of all networks. Overall, the performance of the mutual learning ensemble largely outperforms the baseline ensemble. We also compare the mutual learning ensemble with two supervised upper bounds, which are trained using ground truths. The *Single Model* denotes evaluation using the best single model, and the *Ensemble Feature* denotes evaluation using feature ensemble among multiple networks. Our mutual learning ensemble is relatively close to them with evaluation using a single model.

Effectiveness of selection Selection phase in PEG finds and preserves an optimal subset of base networks for better multi-model training. The experiment with mutual learning and selection is denoted as *Multi-model + PML + Sel.*

in Table 3. For this experiment, the selection is performed to preserve a combination of 3 networks from all 8 networks using the cooperative game in Sect. 3.1.1, then the preserved models are trained by mutual learning. Experimental results show that the selection phase improves the performance of *Multi-model + PML* even using fewer models. The superior performance indicates that some models may be redundant and cannot provide more discrimination but require more computation during training. However, the selection effectively preserves better models with the proposed cooperative gaming while abandoning weak models that could even degrade the overall discrimination capability of the whole ensemble. Without those weaker models, models will achieve better discrimination from more reliable and efficient mutual learning.

Effectiveness of reproduction and mutation Reproduction and mutation drive the PEG framework to train more diverse models by mutating their hyper-parameters, which is the key to the exploration of model diversity in our evolution process. With this component, PEG achieves the best performance in Table 3 as *Multi-model + Sel. + Rep. & Mut.*. The effectiveness of reproduction and mutation can be attributed to the exploration of training more diverse models with selection preserving the better ones of them after mutual learning. Beneficial from the iteration of reproduction, mutation, and selection, PEG keeps exploring and exploiting diverse and discriminative capacity for better re-ID models. In addition, multiple network ensemble with different architectures is also used to exploit their diversity. To further validate the effectiveness for the diversity of reproduction and mutation, we evaluate PEG with only a single model to initialize the population with reproduction and mutation, as *Single architecture + PEG* in Table 3. Compared with the single model baseline, the experiment improves the accuracy by large margins, and it also outperforms the multi-model without reproduction and mutation. The results demonstrate that reproduction and mutation are more important for exploring diversity. Moreover, our full approach of PEG with both multiple architectures and reproduction&mutation performs the

Table 4 Ablation studies of PEG on a stronger baseline of cluster contrast loss (CCL)

Methods	Market-1501				DukeMTMC-reID			
	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10
CCL-Single	83.3	92.6	96.8	97.9	74.4	86.2	92.1	93.9
CCL-Multi	79.4	91.2	96.8	97.8	68.9	81.6	89.9	91.7
CCL-Multi + Sel.	84.3	92.9	97.1	98.2	74.9	86.3	92.4	94.1
CCL-Multi + PEG	87.1	94.6	98.0	98.8	76.8	86.4	93.1	95.0

CCL-Single denotes the baseline using the model of IBN-ResNet50. *CCL-Multi* means that all models are used for pseudo label prediction and every model is then trained individually. *Sel.* denotes selection, and *PEG* denote our full method

Table 5 Ablation studies of components of population mutual learning (PML) on selected models without reproduction and mutation

Methods	Market-1501				DukeMTMC-reID			
	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10
<i>Supervised upper bound</i>								
Single Model	84.4	94.1	97.9	98.8	71.2	85.5	92.5	94.3
Ensemble Feature	86.4	94.9	98.0	98.9	77.7	88.9	94.5	95.6
Baseline Ensemble (Only \mathcal{L}_{vor})	76.6	89.1	95.7	97.2	63.2	77.2	87.5	91.0
PML w/o Θ_T	73.3	87.9	95.3	97.1	62.6	77.5	87.0	90.1
PML w/o \mathcal{L}_{mid}	77.2	90.2	95.9	97.4	65.1	79.2	88.5	91.2
PML w/o \mathcal{L}_{mtri}	77.0	89.6	95.8	97.4	65.3	79.3	88.8	91.2
PML	79.2	90.9	96.3	97.5	66.9	80.8	88.8	91.6

Supervised Upper Bound—Deep models trained using the labelled training images. *Single Model*—evaluation using the best single model. *Ensemble Feature*—evaluation using feature ensemble among multiple networks. *Baseline Ensemble* - Models jointly trained by shared pseudo-labels but without mutual learning. \mathcal{L}_{vor} (Eq. 15), Θ_T (Eq. 8), \mathcal{L}_{mid} (Eq. 9) and \mathcal{L}_{mtri} (Eq. 10) are described in Sect. 3.1.3

best, demonstrating that the diversities from the two components are complementary.

Generalization Analysis To validate the generalization of our approach with different baseline training methods, ablation studies on a stronger baseline of cluster contrast loss are evaluated as shown in Table 4. Compared with the single model, it performs not good when directly using multiple models for pseudo-label prediction, denoted as *CCL-Multi*. The distinct degradation of performance indicates that the weak models make a negative impact on such a stronger baseline. The models converge quickly to the inaccurate pseudo label partially predicted by the weak models and can no longer be improved. However, the performance of the multi-model is largely improved using the selection before training. It demonstrates that the selection still preserves better models effectively and abandons the weak models which are harmful to the ensemble. Furthermore, *CCL-Multi + PEG* produces the best performance on both datasets, validating the effectiveness of the mutation and reproduction. The superior results show that our PEG is effective and generalizable for different baseline methods.

4.4.2 Evaluation of Cross-Reference Scatter

In this section, we first validate the basic motivation of the cross-reference scatter, the phenomenon that more accurate labels lead to larger intra-cluster cohesion and inter-cluster separation in the trained feature space. We use inter-/intra-

cluster scatters (ICS) to measure the separation as well as the cohesion over the feature space of models. As shown in Fig. 4, a larger ICS means larger inter-cluster separation and intra-cluster cohesion. We evaluate the ICS of models trained by labels with different accuracy. Specifically, the label accuracy is controlled by replacing a part of the ground truth with randomly incorrect labels. The results shown in Fig. 5 indicate a positive correlation between the ICS and the label accuracy, which confirms the basic hypothesis of CRS.

We also evaluate the cross-reference scatter with different metrics for clustering algorithm to compare the performance of re-ID models without ground truth. All comparison models with different architectures were first pre-trained in DukeMTMC-reID dataset and then evaluated in Market-1501 by the metrics. To demonstrate whether a metric can show the relative performance between models, we evaluate the correlation between the metric scores and the re-ID performance for all metrics, as shown in Fig. 6. Since CRS measures models at the start of every generation in PEG but aims to select the models that perform better after training, the metric scores were calculated before training and the re-ID performance is evaluated with mAP after the model has been unsupervised trained on the unlabeled data from Market1501, which represents more latent performance of models. We first compare ICS with two metrics for clustering algorithm including Davies–Bouldin Index (DBI) and Silhouette Coefficient (SC), as shown in the first line of Fig. 6. All three metrics are calculated directly on the feature space of the eval-

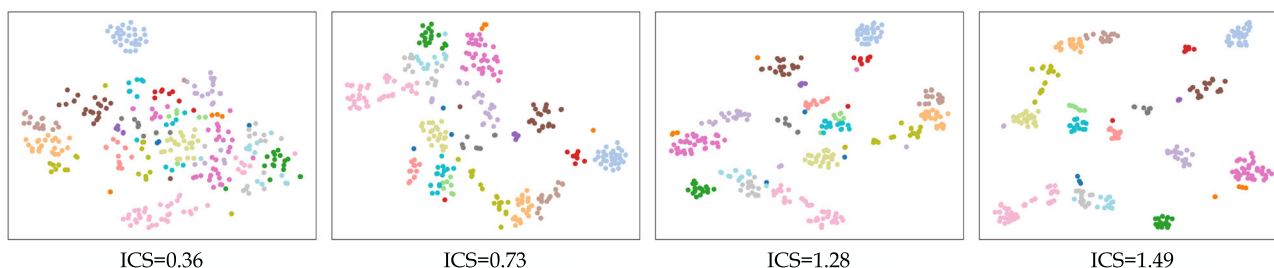


Fig. 4 Illustration of feature distribution with different Inter-/intra-cluster scatters (ICS). A larger ICS means larger cohesion within feature clusters and larger separation across feature clusters. Best viewed in color

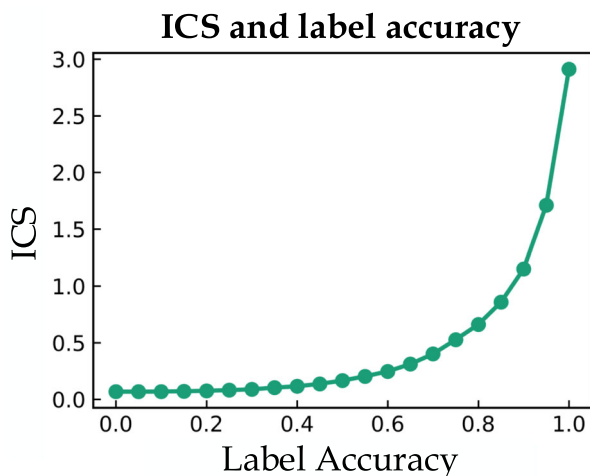


Fig. 5 The positive correlation between inter-/intra-cluster scatter (ICS) and label accuracy indicates that more accurate labels usually lead to larger ICS, which means larger cohesion within feature clusters and larger separation across feature clusters during model training

uated models by performing k-means clustering. For each metric, we used Spearman's Rank Correlation (ρ) (Spearman, 1961) and Kendall's Rank Correlation (τ) (Kendall, 1938) to measure the correlation between the metric scores and the re-ID performance. However, we clearly see the poor correlation of the metrics with the re-ID performance according to the small ρ and τ , indicating that the distribution of features before training can not show the real performance of models. Then we evaluate the three metrics using our proposed cross-reference (CR) evaluation where metrics are calculated on the feature distribution of a reference model trained by predicted labels. As illustrated in the second line of Fig. 6, correlations are consistently improved by CR, which validates its effectiveness. Importantly, our CRS (ICS+CR) performs the highest correlation with $\rho = 0.93$ and $\tau = 0.86$ among all six compared metrics. Besides, we also present the rankings of models under different metrics in Fig. 7. Compared with the ground truth ranking result in the last column, CRS achieves a similar ranking of models while other metrics fail to rank them well. The superior performance of CRS can be attributed to two reasons. One reason is the cross-reference evaluation that measures the accuracy of predicted labels can

better reflect models' performance, and another reason is the ICS which better measures the convergence degree of the reference model. Specifically, both DBI and SC focus on the distribution of the difficult edge samples of clusters while they ignore the overall distribution and thus cannot measure well the degree of model convergence.

We also evaluate CRS with different clustering algorithms such as DBSCAN. In our work, DBSCAN is adopted in model learning to generate more accurate pseudo-labels like many recent unsupervised re-ID works. However, it is not applicable for CRS because the fair comparison of CRS among models requires the same cluster number during clustering, while DBSCAN cannot guarantee that. Specifically, CRS defined by the ratio of intra-/inter-cluster variance is relative to the cluster numbers. And the cluster numbers by DBSCAN with different evaluated feature models are likely to be different, making it unfair to compare their CRS for model selection. In our work, we use kmeans with the same cluster number k for all evaluated models. To validate its effectiveness, we evaluate the correlation between CRS and model performance using different clustering algorithms, as shown in Fig. 8. Compared with kmeans ($M = 500$), DBSCAN achieves a much lower Spearman's Rank Correlation (ρ) (Spearman, 1961) and Kendall's Rank Correlation (τ) (Kendall, 1938) between the metric and mAP values. The results show that CRS with DBSCAN fails to measure the models, and KMeans does it better. Therefore, we use kmeans with the same cluster number for CRS.

Furthermore, we study the number of clusters M for k-means in CRS, which is hard to fix in the real world. We first compare the correlation between CRS and re-ID performance with different values of M , as shown in Fig. 8. CRS shows stronger correlations when M is set to 500 or 700, which is close to the number of person IDs in the datasets. The good performance of this cluster number is consistent with other kmeans based methods like MMT (Ge et al., 2020). When M is larger, the correlation will be weaker. But the CRS still basically reflects the performance of re-ID models, indicating it is robust to the cluster number. On the other hand, we also evaluate the performance of our full method with different M on both Market1501 and DukeMTMC-reID datasets. As

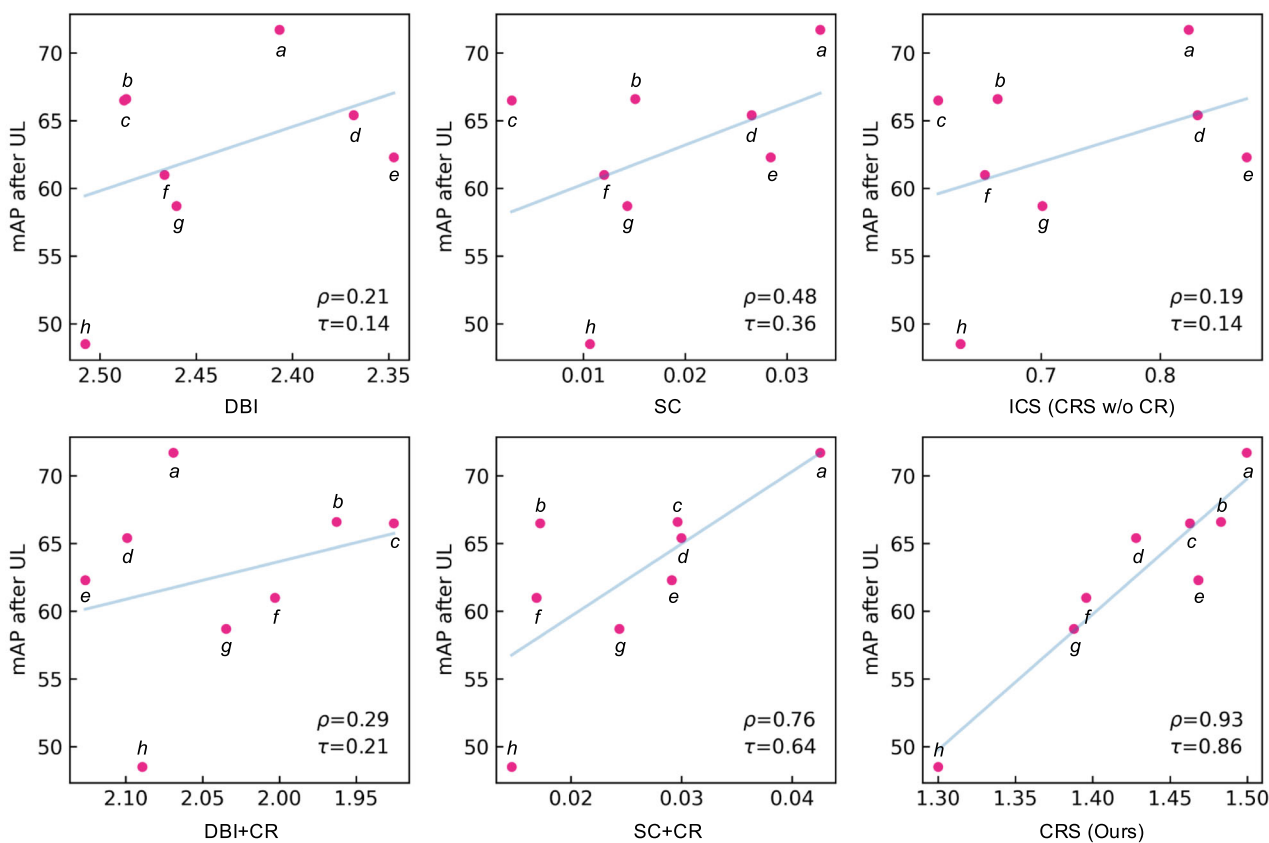


Fig. 6 Comparison with different unsupervised measures on the correlation between the measures and re-ID performance (mAP after unsupervised learning) over different models (point a–h) on Market-1501 dataset. For each measure, we use Spearman’s Rank Correlation (ρ) (Spearman, 1961) and Kendall’s Rank Correlation (τ) (Kendall,

1938) to measure the correlation between the metric and mAP values. A higher absolute value of ρ (or τ) indicates a stronger correlation. Our proposed CRS shows a stronger correlation, indicating that it better reflects the performance of re-ID models

shown in Table 6, the re-ID performance is generally consistent with the correlations of CRS. PEG performs best when M is set to 700, where CRS also achieves the highest ρ and τ , making selection able to select better models.

To validate CRS for very weak evaluated models which predict mostly wrong pseudo labels, we estimate CRS at different noise levels. Although its predicted labels are partially wrong for each evaluated model, we add extra noises by disrupting the label order of a particular portion of samples. As shown in Fig. 9, CRS maintains a stronger correlation between its values and model performance with the increase of the noise ratio, indicating its robustness for the wrong labels. When the noise level is too high such as 0.8, the correlation visibly deteriorated. However, higher CRS can still roughly reflect the better models.

In addition, we evaluate CRS with different architectures of the reference model. Four models are compared with fewer parameters to more parameters including OSNet, DenseNet-121, ResNet-50, and ResNet-101. For fair, all reference models are trained for 500 iterations during the evaluation

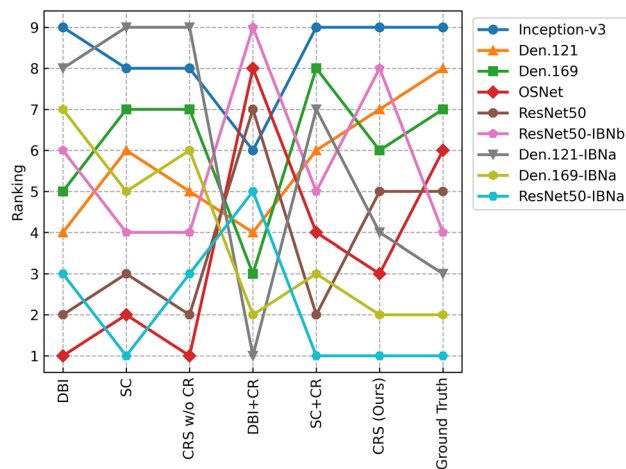


Fig. 7 Rankings of 9 models under existing clustering measures and the proposed metric “CRS”. The ground truth ranks models by the mAP after unsupervised learning

of CRS. As shown in Table 7, we observe that models easy to converge such as OSNet and ResNet-50 show better mea-

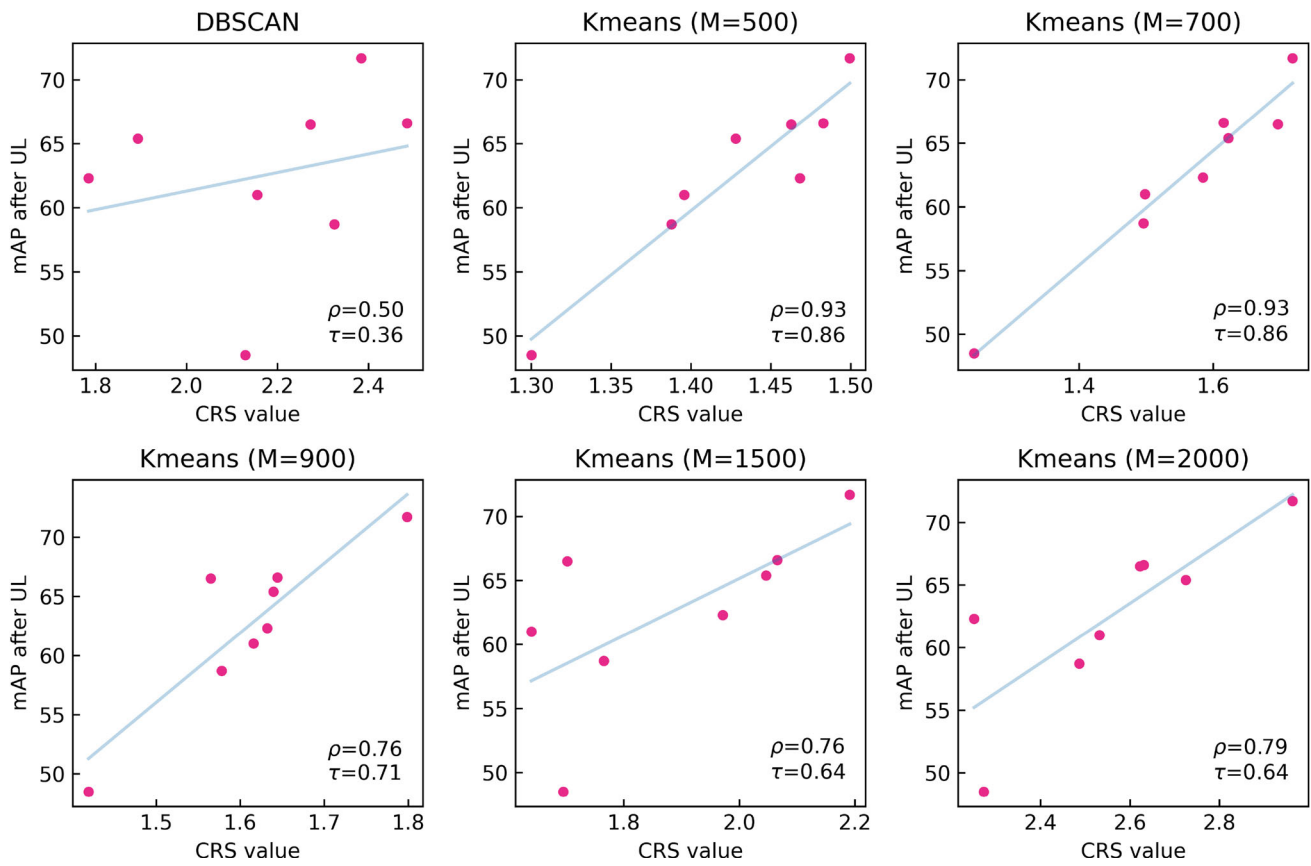


Fig. 8 Comparison of CRS with different cluster settings on the correlation between the measures and re-ID performance (mAP after unsupervised learning) over different models on Market-1501 dataset

Table 6 Comparison of re-ID performance using different cluster numbers M for kmeans clustering in CRS

Methods	Market-1501				DukeMTMC-reID			
	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10
PEG ($M = 500$)	84.3	93.7	97.8	98.5	71.9	83.8	91.2	93.5
PEG ($M = 700$)	85.0	94.1	97.8	98.8	73.3	84.8	92.0	94.1
PEG ($M = 900$)	83.8	93.1	97.4	98.4	72.3	83.7	91.3	93.8
PEG ($M = 1500$)	83.6	93.1	97.2	98.5	71.5	84.1	91.6	93.6
PEG ($M = 2000$)	83.5	93.7	97.8	98.6	71.3	83.6	91.2	93.4

Table 7 Comparison with different architecture for the reference model in CRS

Reference model	ρ	τ	Param.	Time/iter.
DenseNet-121	0.74	0.57	6.95M	0.99s
ResNet-50	0.86	0.71	23.51M	1.11s
ResNet-101	0.40	0.36	42.50M	1.87s
OSNet	0.93	0.86	1.91M	0.98s

All models are trained for 500 iterations during the evaluation of CRS. Models easy to converge such as OSNet and ResNet-50 show better measurement

surement for higher correlation ρ and τ , while models hard to converge, like DenseNet-121 and ResNet-101, don't perform well. Specifically, DenseNet using a dynamic architecture and ResNet-101 have deep layers and amounts of parameters, therefore they both require much more time to train. Since

only a few training iterations are performed in CRS, the two architectures cannot show a sufficiently differentiable difference in feature distribution when evaluating different models. Moreover, we evaluate PEG for CRS with other light-weight networks as the reference model, including MobileNet and ResNet18. Different from OSNet specially designed for re-ID, the other two architectures are designed for general purpose. Table 8 shows the re-ID performance of PEG with different reference models for CRS. Our method achieves comparable re-ID performance consistently on Market-1501 and DukeMTMC-reID datasets. The results indicate that our CRS metric is model-general for reference models, which is not limited to certain architectures. In this work, we adopt the OSNet as the reference model of CRS in all other experiments for less time-consuming and more accurate measurement.

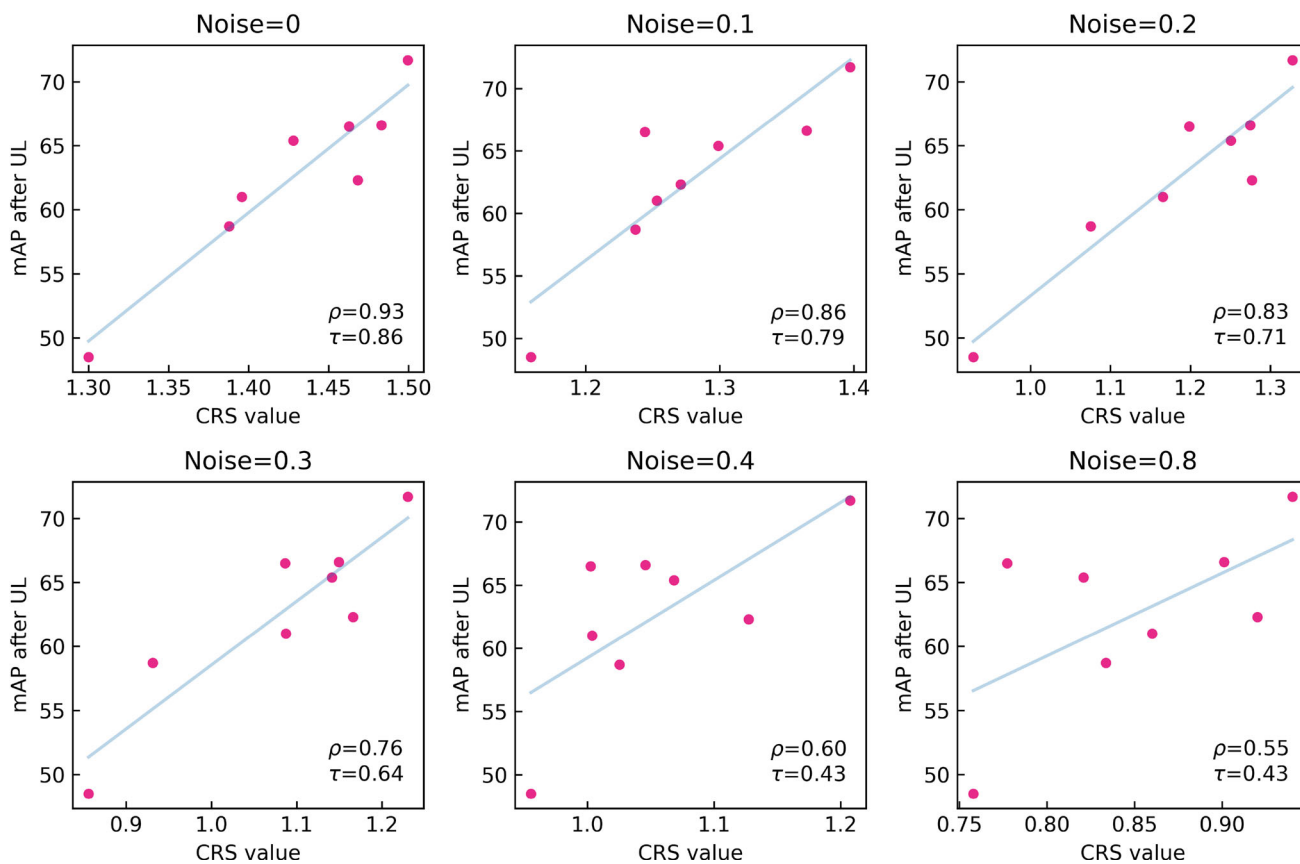


Fig. 9 Comparison of CRS at different noisy levels of pseudo-labels on the correlation between the measures and re-ID performance (mAP after unsupervised learning) over different models on Market-1501 dataset

Table 8 Comparison of re-ID performance of PEG using different light-weight networks as reference models in CRS

Reference Models	Market-1501				DukeMTMC-reID			
	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10
OSNet	84.3	93.7	97.8	98.5	71.9	83.8	91.2	93.5
MobileNet	83.9	93.1	97.8	98.6	72.0	83.9	91.2	93.3
ResNet18	84.8	93.8	97.7	98.7	72.1	83.4	91.8	93.8

Table 9 Comparison with different selection strategies for initial network architectures

Selection strategies	mAP	R-1	R-5	R-10
Deepest	77.9	89.9	96.1	97.1
Most heavyweight	76.7	89.3	96.1	97.5
Cooperative game (ours)	79.2	90.9	96.3	97.5

The results are tested after once selection and mutual learning without mutation on Market-1501 dataset

Table 10 Comparison between individual selection and group selection in PEG

Selection strategies	mAP	R-1	R-5	R-10
Individual selection	82.3	92.7	97.1	98.2
Group selection	83.6	93.3	97.3	98.3

Individual selection selects networks with better individual performance (CRS) while group selection selects the network combination with better overall performance (CRS)

4.4.3 Analysis of Selection

Comparison with different model selection strategies For the method of selection of networks in PEG, we first compare our cooperative gaming (using the best-response dynamics according to CRS) with different selection strategies of network architectures, for example, using some of the deepest or weight-heaviest networks since deeper or weight-heavier networks generally achieve better performance. Considering that these strategies can not select networks from ones with the same architecture, in this experiment, we perform the selection from networks with different architectures only once and then train them by mutual learning before testing. The experiment results shown in Table 9 indicate that our approach selects better models which achieve higher performance through mutual learning. The better selection can be attributed to that the CRS approximately measures the

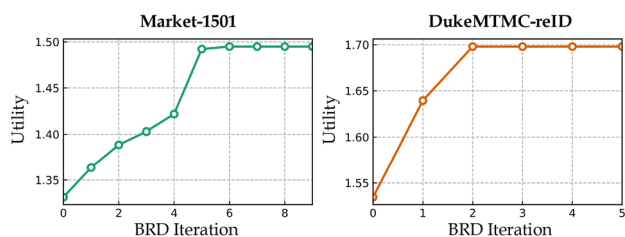


Fig. 10 Illustration of curves of utility outcome (calculated by CRS) over the best response dynamics (BRD) iterations in the first selection phase during population-based training. The utility outcome increases strictly and eventually halts at a Nash equilibrium

discriminative capability of models by efficiently using the unlabeled data.

Moreover, we compare our group selection with the individual selection in PEG. For individual selection, we evaluated the CRS of every single network and accordingly preserved the best L networks. While for group selection, we use the cooperative game to find and preserve the group of L networks with the highest overall CRS. Through the iterative evolutionary game, the group selection performs better, as shown in Table 10. The superior performance indicates that networks preserved by the group selection are more complementary and it helps to achieve a better population in later evolutionary training.

Convergence analysis of cooperative selection gaming.

We now discuss the convergence of the cooperative gaming of selection. Note that in every iteration of best response dynamics in Eq. 2, the outcome of the utility function strictly increases. Thus, no cycles are possible. Since the game is finite by assumption, it eventually ends, necessarily at a Nash equilibrium. The convergence of the cooperative game is illustrated in Fig. 10, where each game eventually halts at a Nash equilibrium.

4.4.4 Parameter Analysis

Analysis of agent number L in the selection. The agent number L in the cooperative game of selection determines the maximal size of the selected subset of networks. Here we evaluate the performance of our method and computational cost of the selection over different values of L , as shown in Fig. 11. Usually, a smaller L will lead to a lack of diversity of the population since only a small number of networks can be preserved during selection. However, L should not be very large because it will waste much more computational resources for solving the best-response dynamics. On the other hand, a larger L also means the larger size of the population in the next generation, which will cost more time for mutual learning among the networks. Taken together, a L of 3 is proper in our experiments, which achieves good performance without consuming too many computational resources.

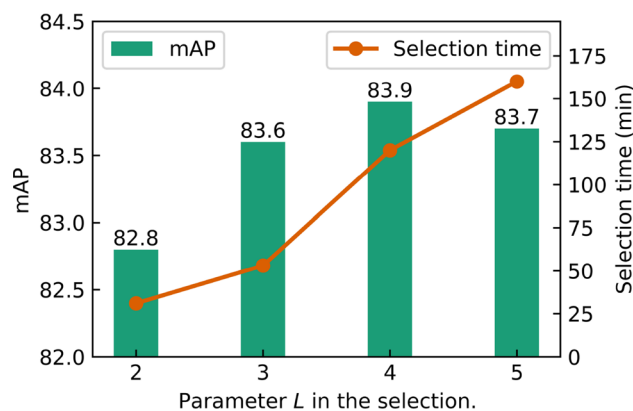


Fig. 11 Comparison with different agent numbers L in the cooperative selection game. A larger L leads to better performance but higher time consumption. The mutation factor r is set to 0.2 for stability

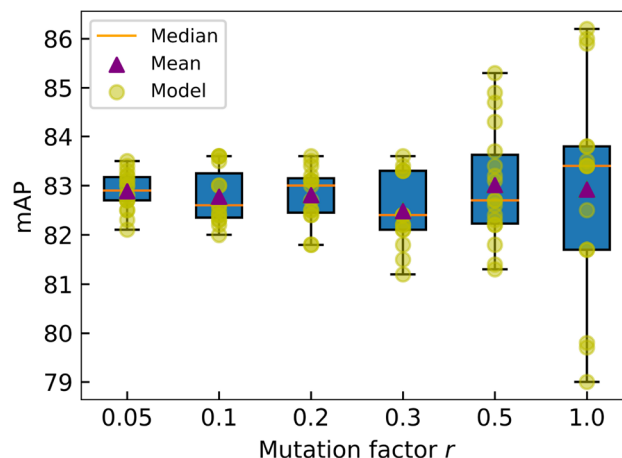


Fig. 12 Performance of networks in populations over different mutation factor r . Each box represents a population and points denote models. A larger r leads to better performance but larger variance of networks within populations

Analysis of mutation factor r The mutation factor r in Sect. 3.1.2 will affect the diversity of populations and so the evolutionary training processes. We studied this parameter by setting it to different values and checking the mAP performance of all networks in the populations. Fig. 12 shows experimental results on Market-1501, where each circle point denote a single model. Using a larger r usually leads to a higher diversity within populations, which further leads to a higher possibility of achieving better performance. Specifically, a larger r results in a higher upper bound (maximized performance) and a similar average value. Notably, the average values do not represent the final performance. Although PEG aims to train a population of diverse networks, only one network is selected automatically according to Cross Reference Scatter for inference at the end of training, which is probably to be the better one. Therefore, the final performance of our method doesn't depend on the average values of the population but depends on the performance of the selected model. On the other hand, a population with a higher upper

Generation	1	2	3	4	5
mAP	66.0	83.8	84.9	85.3	84.4
R-1	81.8	93.0	93.5	94.2	93.1

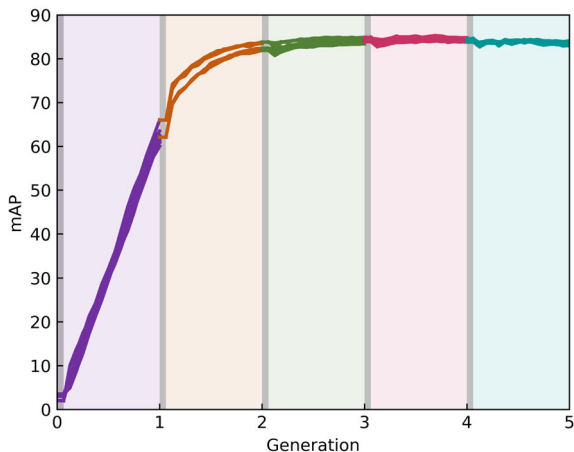


Fig. 13 Illustration of the model performance in every generation for 5 generation evolution

bound is more likely to select a better model. For example, when r is set to 0.05 the best model in the population achieves 83.6% mAP, but when r is set to 0.5, there are 1/4 models in the population that achieve mAP higher than 83.6%, which may be selected for superior performance. Importantly, the final model is selected according to Cross Reference Scatter which is to estimate model performance by unlabeled training data. Experimental results in Sect. 4.4.2 demonstrated that better models are likely to have higher CRS values to be selected. And when $r = 0.5$ it provides more better models as candidates for the final selection. However, a larger r will also bring larger variance and instability of network performance within populations because it may reproduce very weak networks that drag down the overall discrimination capability of the whole population by mutual learning. Given all of that, we set r to 0.5 for both performance and stability.

Analysis of the number of generations To analyze the number of generations, we provide the model performance in every generation for 5 generation evolution, as shown in Fig. 13. The performance of models is boosted rapidly in the first and second generation, and the boost slows down gradually as the generation increases. After three generations of evolution, the performance is nearly convergent, and models achieve stable results.

4.4.5 Multiple Models versus Heavyweight Models

Heavyweight networks are more likely to learn discriminative representations than lightweight models since they have deeper architectures and more parameters. However, models with heavyweights require more time and computa-

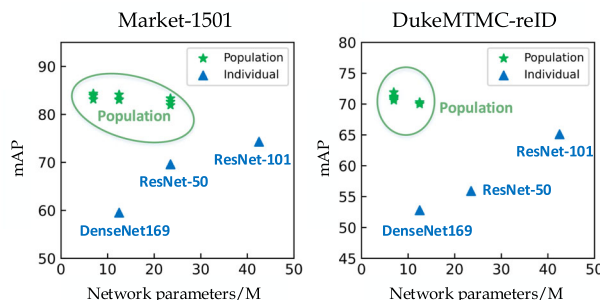


Fig. 14 Comparison between the lightweight networks within the population and heavyweight networks trained individually on two datasets

tional resources during both the training and inference stages, making them infeasible in practice. Our experiments show that through PEG, lightweight models can surpass heavyweight models with IBN that are individually trained under unsupervised conditions. Take the market-1501 dataset as an example. After the evolutionary game of the population of lightweight networks, all member networks of the population achieve better performance than heavyweight networks, such as ResNet-101, as shown in Fig. 14. Specifically, one of the networks achieves a much higher mAP of 84.3% than ResNet-101 with only 1/3 of the parameters. The superior results can be attributed to two aspects. One is the mutation and selection that sufficiently explore and exploit the population. Mutation makes networks learn diverse knowledge, and selection maintains the optimal model groups and abandons the others. The models in the selected and preserved groups are complementary, so they produce more accurate and robust pseudo-labels for the next training phase and learn more discriminative features. The second reason is the mutual learning performed among all networks in the whole population. Since the models preserved by mutation and selection are diverse and complementary, each contains only a small part of the knowledge of the whole population. Through mutual learning, the knowledge of the population is assembled into each network by distillation, which equips the models with more discriminative capability. The PEG method explores the potential of lightweight networks and searches for the approximate global optimal solution and thus outperforms the heavyweight models.

4.5 Computational Cost

To evaluate the efficiency and effectiveness of the computational cost, we evaluate a series of large single models for reference as shown in Table 11. Experiments are conducted on four V100 GPUs. From the perspective of training, our method requires comparable computational cost with the large single models (such as ResNets420) while achieving significant performance improvement. And from the perspec-

Table 11 Comparison of computational cost between single models and PEG with different population sizes

Methods	Performance		Training cost			Testing cost	
	mAP	R-1	Param. /M	Complexity /GMac	Time /h	Param. /M	Complexity /GMac
IBN-DenseNet169	57.4	75.2	12.49	2.23	3.5	12.49	2.23
IBN-ResNet50	69.6	84.9	23.51	4.08	3.9	23.51	4.08
IBN-ResNext101	72.2	87.4	42.13	6.54	4.8	42.13	6.54
ResNet200	73.1	88.7	62.65	10.02	8.9	62.65	10.02
ResNeXts420	73.1	86.5	189.84	20.62	19.3	189.84	20.62
PEG (small)	84.1	93.0	72.00	12.62	10.7	12.49	2.23
PEG (large)	84.3	93.7	128.85	24.57	21.1	12.49	2.23

PEG (small) selects only two models during selection, and every model reproduces to 2 times. And PEG (large) follows the original settings according to the implementation details

tive of testing, our method requires as less computational cost as the small single models (such as IBN-DenseNet169) and surpasses them by large margins. More detailed descriptions are listed below.

(1) The improvement by increasing parameters is limited for single models on re-ID performance, and our PEG largely surpasses the best single model with comparable computations, demonstrating the cost meaningful. Specifically, we evaluate five architectures from lightweight to heavyweight including IBN-DenseNet169, IBN-ResNet50, IBN-ResNext101, ResNet200 and ResNeXts420. As parameters increase, single models usually achieve better performance, whereas they require more computational complexity and time for training. However, the improvement is limited when the parameters are very large, i.e., ResNeXts420 cannot surpass ResNet200 even though more than two times of parameters and training time are used. Compared with single models, PEG improves the accuracy by large margins. Although population-based training demands more cost, the cost is worth and affordable. Importantly, PEG provides further improvement that cannot be achieved by simply increasing model parameters.

To reduce the cost, we provide two implementation versions of PEG: small and large, with different sizes of the population. Specifically, PEG (small) maintains a lightweight population for efficient training, which selects only two models during selection and every model reproduces to 2 times. And PEG (large) follows the original settings according to the implementation details. Significantly, the provided PEG (small) achieves comparable performance with PEG (large) and requires only half the training cost. It also outperforms the best single architecture ResNet200 for more than 10% of mAP while costing comparable computing resources, which is more affordable and efficient than the large version. We suggest the version of PEG (small) for application in resource-limited environments.

(2) PEG requires less computational cost for testing, making it more applicable and valuable in practice. As is shown in Table 11, the computational complexity during

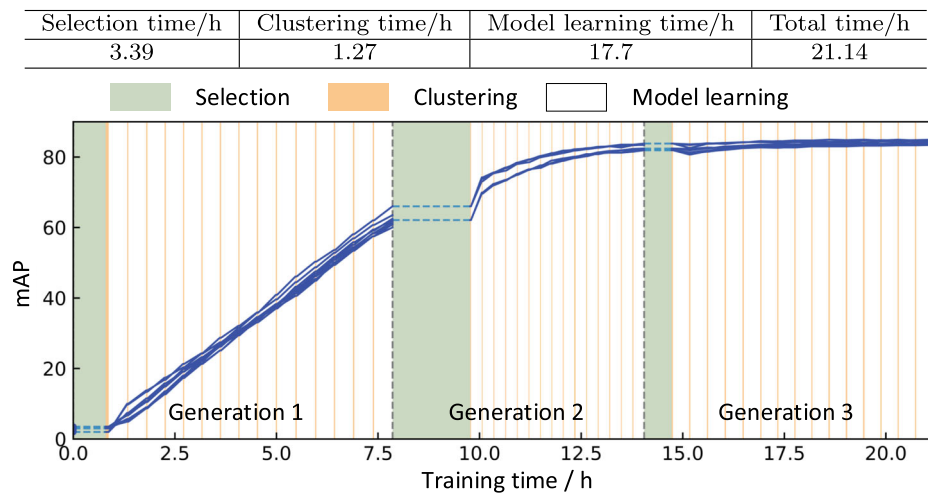
testing of PEG is largely less than the large single models, even nearly 1/5 of the best one, ResNet200. It only requires as less computational cost as small single models such as IBN-DenseNet169 while surpassing its performance by large margins. It is because only one network in the population is selected in the end for evaluation. Since the training procedure is only conducted once, while the test will be continuously repeated in the actual re-ID system, PEG with less testing cost is applicable and valuable in practice.

To further analyze the training time of every procedure in our approach, we illustrated the training process over time in Fig. 15. Among the total training time, model learning accounts for the largest proportion. Model learning is performed by data loading, feedforward of all networks, backward of losses, and updating of parameters. This part of time is relative to the number and depth of networks. For example, the time of model learning in Generation 2 is shorter than in the other generations because there are only four networks in the population. The time of the selection stage is different in the three generations. On the one hand, it is affected by the number of candidate networks. On the other hand, it is affected by the convergence of the best-response dynamics. Moreover, clustering costs the least time, only for the extraction of features and execution of clustering algorithms.

5 Conclusion

The paper proposed a population-based evolutionary gaming which trains concurrently a population of networks for unsupervised person re-ID. We demonstrate that the population can evolve and achieve progressive discrimination through iterative selection to preserve adaptive networks, reproduction and mutation to provide more diversity, and mutual learning to assemble knowledge. Moreover, our proposed cross-reference scatter can approximately estimate the performance of networks using unlabeled data and thus is utilized as the utility of cooperative game in the selection phase.

Fig. 15 Illustration of time consumption of every procedure in our approach. Time of selection, clustering and model learning are represented by green, yellow and white, respectively. Blue curves represent the performance of every model in the population



Our approach not only produces a new state-of-the-art accuracy on multiple benchmarks but also provided a fresh insight for population-based multi-network training.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s11263-022-01693-7>.

Acknowledgements This work is partially supported by grants from the Key-Area Research and Development Program of Guangdong Province under contact No. 2019B010153002, and grants from the National Natural Science Foundation of China under contract No. 61825101 and No. 62088102. The computing resources of Pengcheng Cloudbrain are used in this research.

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

- Ali, B., Moriyama, K., Kalintha, W., Numao, M., & Fukui, K. I. (2020). Reinforcement learning based metric filtering for evolutionary distance metric learning. *Intelligent Data Analysis*, 24(6), 1345–1364.
- Baker, F. B., & Hubert, L. J. (1975). Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association*, 70(349), 31–38.
- Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*(pp. 132–149).
- Chen, G., Lu, Y., Lu, J., & Zhou, J. (2020). Deep credible metric learning for unsupervised domain adaptation person re-identification. In *Computer vision—ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16* (pp. 643–659). Springer
- Chen, H., Lagadec, B., & Bremond, F. (2021a). Ice: Inter-instance contrastive encoding for unsupervised person re-identification. [arXiv:2103.16364](https://arxiv.org/abs/2103.16364)
- Chen, H., Wang, Y., Lagadec, B., Dantcheva, A., & Bremond, F. (2021b). Joint generative and contrastive learning for unsupervised person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2004–2013).
- Dai, Z., Wang, G., Zhu, S., Yuan, W., & Tan, P. (2021). Cluster contrast for unsupervised person re-identification. [arXiv:2103.11568](https://arxiv.org/abs/2103.11568)
- Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, 1(2), 224–227. <https://doi.org/10.1109/TPAMI.1979.4766909>.
- Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*.
- Deng, W., Zheng, L., Ye, Q., Kang, G., Yang, Y., & Jiao, J. (2018). Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *IEEE CVPR*.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (pp. 1–15). Springer.
- Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.
- Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD* (pp. 226–231).
- Fan, H., Zheng, L., Yan, C., & Yang, Y. (2018). Unsupervised person re-identification: Clustering and fine-tuning. *TOMCCAP*, 14(4), 83:1–83:18.
- Fu, Y., Wei, Y., Wang, G., Zhou, Y., Shi, H., & Huang, T. S. (2019). Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 6112–6121).
- Fu, D., Chen, D., Bao, J., Yang, H., Yuan, L., Zhang, L., Li, H., & Chen, D. (2021). Unsupervised pre-training for person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14750–14759).
- Fukui, K., Ono, S., Megano, T., & Numao, M. (2013). Evolutionary distance metric learning approach to semi-supervised clustering with neighbor relations. In *2013 IEEE 25th international conference on tools with artificial intelligence* (pp. 398–403). IEEE.
- Ge, Y., Chen, D., & Li, H. (2020a). Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. [arXiv:2001.01526](https://arxiv.org/abs/2001.01526)
- Ge, Y., Zhu, F., Chen, D., Zhao, R., & Li, H. (2020b). Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. [arXiv:2006.02713](https://arxiv.org/abs/2006.02713)

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).
- Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002). Clustering validity checking methods: Part ii. *ACM Sigmod Record*, 31(3), 19–27.
- Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), 993–1001.
- Ho, D., Liang, E., Chen, X., Stoica, I., & Abbeel, P. (2019). Population based augmentation: Efficient learning of augmentation policy schedules. In *International conference on machine learning, PMLR* (pp. 2731–2741).
- Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K.Q. (2016). Deep networks with stochastic depth. In *European conference on computer vision (ECCV)* (pp. 646–661). Springer.
- Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., & Weinberger, K. Q. (2017a). Snapshot ensembles: Train 1, get m for free. [arXiv:1704.00109](https://arxiv.org/abs/1704.00109)
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q. (2017b). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4700–4708).
- Huang, Y., Peng, P., Jin, Y., Xing, J., Lang, C., & Feng, S. (2019). Domain adaptive attention model for unsupervised cross-domain person re-identification. [arXiv:1905.10529](https://arxiv.org/abs/1905.10529)
- Hubert, L. J., & Levin, J. R. (1976). A general statistical framework for assessing categorical clustering in free recall. *Psychological Bulletin*, 83(6), 1072.
- Jaderberg, M., Dalibard, V., Osindero, S., Czarniecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., & Simonyan, K., et al. (2017). Population based training of neural networks. [arXiv:1711.09846](https://arxiv.org/abs/1711.09846)
- Jaderberg, M., Czarniecki, W. M., Dunning, I., Marris, L., Lever, G., Castaneda, A. G., et al. (2019). Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443), 859–865.
- Ji, H., Wang, L., Zhou, S., Tang, W., Zheng, N., & Hua, G. (2021). Meta pairwise relationship distillation for unsupervised person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3661–3670).
- Jin, X., Lan, C., Zeng, W., & Chen, Z. (2020). Global distance-distributions separation for unsupervised person re-identification. [arXiv:2006.00752](https://arxiv.org/abs/2006.00752)
- Kalintha, W., Ono, S., Numao, M., & Ki, F. (2019). Kernelized evolutionary distance metric learning for semi-supervised clustering. *Intelligent Data Analysis*, 23(6), 1271–1297.
- Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2), 81–93.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- Krogh, A., & Vedelsby, J. (1994). Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, 7, 231–238.
- Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems* (pp. 6402–6413).
- Li, J., & Zhang, S. (2020). Joint visual and temporal consistency for unsupervised domain adaptive person re-identification. In *European conference on computer vision* (pp. 483–499). Springer.
- Li, M., Zhu, X., & Gong, S. (2018). Unsupervised person re-identification by deep learning tracklet association. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 737–753).
- Li, M., Zhu, X., & Gong, S. (2019a). Unsupervised tracklet person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(7), 1770–1782.
- Li, Y. J., Lin, C. S., Lin, Y. B., & Wang, Y. C. F. (2019b). Cross-dataset person re-identification via unsupervised pose disentanglement and adaptation. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 7919–7929).
- Liao, S., Hu, Y., Zhu, X., & Li, S. Z. (2015). Person re-identification by local maximal occurrence representation and metric learning. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.
- Lin, Y., Dong, X., Zheng, L., Yan, Y., & Yang, Y. (2019). A bottom-up clustering approach to unsupervised person re-identification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 8738–8745.
- Lin, Y., Xie, L., Wu, Y., Yan, C., & Tian, Q. (2020). Unsupervised person re-identification via softened similarity learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3390–3399).
- Liu, J., Zha, Z.J., Chen, D., Hong, R., & Wang, M. (2019). Adaptive transfer network for cross-domain person re-identification. In *IEEE CVPR*.
- Maulik, U., & Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12), 1650–1654.
- Pan, X., Luo, P., Shi, J., & Tang, X. (2018). Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 464–479).
- Peng, P., Xiang, T., Wang, Y., Pontil, M., Gong, S., Huang, T., & Tian, Y. (2016). Unsupervised cross-dataset transfer learning for person re-identification. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.
- Peng, P., Xing, J., & Cao, L. (2020). Hybrid learning for multi-agent cooperation with sub-optimal demonstrations. In *IJCAI* (pp. 3037–3043).
- Perrone, M. P., & Cooper, L. N. (1992). When networks disagree: Ensemble methods for hybrid neural networks. Technical report: Brown Univ Providence RI Inst for Brain and Neural Systems.
- Qi, L., Wang, L., Huo, J., Zhou, L., Shi, Y., & Gao, Y. (2019). A novel unsupervised camera-aware domain adaptation framework for person re-identification. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 8080–8089).
- Ristani, E., Solera, F., Zou, R. S., Cucchiara, R., & Tomasi, C. (2016). Performance measures and a data set for multi-target, multi-camera tracking. In *IEEE ECCV workshops*.
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65.
- Shen, Z., He, Z., & Xue, X. (2019). Meal: Multi-model ensemble via adversarial learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 4886–4893.
- Singh, S., Hoiem, D., & Forsyth, D. (2016). Swapout: Learning an ensemble of deep architectures. In *Advances in neural information processing systems* (pp. 28–36).
- Song, L., Wang, C., Zhang, L., Du, B., Zhang, Q., Huang, C., & Wang, X. (2018). Unsupervised domain adaptive re-identification: Theory and practice. [CoRR abs/1807.11334](https://arxiv.org/abs/1807.11334)
- Spearman, C. (1961). The proof and measurement of association between two things.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.

- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2818–2826).
- Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems* (pp. 1195–1204).
- Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782), 350–354.
- Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., & Fergus, R. (2013). Regularization of neural networks using dropconnect. In *International conference on machine learning* (pp. 1058–1066).
- Wang, D., & Zhang, S. (2020). Unsupervised person re-identification via multi-label classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10981–10990).
- Wang, J., Zhu, X., Gong, S., & Li, W. (2018). Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *IEEE CVPR*.
- Wang, M., Lai, B., Huang, J., Gong, X., & Hua, X. S. (2020a). Camera-aware proxies for unsupervised person re-identification. [arXiv:2012.10674](https://arxiv.org/abs/2012.10674)
- Wang, Z., Zhang, J., Zheng, L., Liu, Y., Sun, Y., Li, Y., & Wang, S. (2020b). Cycas: Self-supervised cycle association for learning re-identifiable descriptions. In *Computer vision-ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16* (pp. 72–88). Springer.
- Wei, L., Zhang, S., Gao, W., & Tian, Q. (2018). Person transfer gan to bridge domain gap for person re-identification. In *IEEE CVPR*.
- Wu, J., Yang, Y., Liu, H., Liao, S., Lei, Z., & Li, S.Z. (2019). Unsupervised graph association for person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8321–8330).
- Xuan, S., & Zhang, S. (2021). Intra-inter camera similarity for unsupervised person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11926–11935).
- Yang, F., Yan, K., Lu, S., Jia, H., Xie, D., Yu, Z., Guo, X., Huang, F., & Gao, W. (2020). Part-aware progressive unsupervised domain adaptation for person re-identification. *IEEE Transactions on Multimedia*.
- Yang, F., Zhong, Z., Luo, Z., Cai, Y., Lin, Y., Li, S., & Sebe, N. (2021). Joint noise-tolerant learning and meta camera shift adaptation for unsupervised person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4855–4864).
- Ye, M., Ma, A.J., Zheng, L., Li, J., & Yuen, P.C. (2017). Dynamic label graph matching for unsupervised video re-identification. In *Proceedings of the IEEE international conference on computer vision* (pp. 5142–5150).
- Yuan, X., He, P., Zhu, Q., & Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9), 2805–2824.
- Zeng, K., Ning, M., Wang, Y., & Guo, Y. (2020). Hierarchical clustering with hard-batch triplet loss for person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13657–13665).
- Zhai, Y., Lu, S., Ye, Q., Shan, X., Chen, J., Ji, R., & Tian, Y. (2020a). Ad-cluster: Augmented discriminative clustering for domain adaptive person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9021–9030).
- Zhai, Y., Lu, S., Ye, Q., Shan, X., Chen, J., Ji, R., & Tian, Y. (2020b). Ad-cluster: Augmented discriminative clustering for domain adaptive person re-identification. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.
- Zhai, Y., Ye, Q., Lu, S., Jia, M., Ji, R., & Tian, Y. (2020c). Multiple expert brainstorming for domain adaptive person re-identification. [arXiv:2007.01546](https://arxiv.org/abs/2007.01546)
- Zhang, X., Cao, J., Shen, C., & You, M. (2019). Self-training with progressive augmentation for unsupervised cross-domain person re-identification. In *Proceedings of the IEEE international conference on computer vision (ICCV)*. (pp. 8222–8231).
- Zhang, Y., Xiang, T., Hospedales, T. M., & Lu, H. (2018). Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4320–4328).
- Zhao, F., Liao, S., Xie, G. S., Zhao, J., Zhang, K., & Shao, L. (2020). Unsupervised domain adaptation with noise resistible mutual-training for person re-identification. *European conference on computer vision (ECCV)* (pp. 1–18). Glasgow.
- Zheng, K., Liu, W., He, L., Mei, T., Luo, J., & Zha, Z. J. (2021a). Group-aware label transfer for domain adaptive person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5310–5319).
- Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., & Tian, Q. (2015). Scalable person re-identification: A benchmark. In *The IEEE international conference on computer vision (ICCV)*.
- Zheng, L., Yang, Y., & Hauptmann, A. G. (2016). Person re-identification: Past, present and future. [arXiv:1610.02984](https://arxiv.org/abs/1610.02984)
- Zheng, Y., Tang, S., Teng, G., Ge, Y., Liu, K., Qin, J., Qi, D., & Chen, D. (2021b). Online pseudo label generation by hierarchical cluster dynamics for adaptive person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8371–8381).
- Zheng, Z., Zheng, L., & Yang, Y. (2017). Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *IEEE ICCV*.
- Zhong, Z., Zheng, L., Cao, D., & Li, S. (2017). Re-ranking person re-identification with k-reciprocal encoding. In *IEEE CVPR*.
- Zhong, Z., Zheng, L., Li, S., & Yang, Y. (2018). Generalizing a person retrieval model hetero- and homo-generously. In *ECCV* (pp. 176–192).
- Zhong, Z., Zheng, L., Luo, Z., Li, S., & Yang, Y. (2019a). Invariance matters: Exemplar memory for domain adaptive person re-identification. In *IEEE CVPR*.
- Zhong, Z., Zheng, L., Zheng, Z., Li, S., & Yang, Y. (2019b). Camstyle: A novel data augmentation method for person re-identification. *IEEE TIP*, 28(3), 1176–1190.
- Zhong, Z., Zheng, L., Luo, Z., Li, S., & Yang, Y. (2020). Learning to adapt invariance in memory for person re-identification. In *IEEE transactions on pattern analysis and machine intelligence*.
- Zhou, K., Yang, Y., Cavallaro, A., & Xiang, T. (2019). Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3702–3712).
- Zou, Y., Yang, X., Yu, Z., Kumar, B.V., & Kautz, J. (2020). Joint disentangling and adaptation for cross-domain person re-identification. In *Computer vision-ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, Part II 16* (pp. 87–104). Springer.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.