# Bipartite Mode Matching for Vision Training Set Search from a Hierarchical Data Server

**Yue Yao**[1,†], **Ruining Yang**[2], **Tom Gedeon**[3]

[1]Shandong University, China
[2]Northeastern University, United States
[3]Curtin University, Australia

## Abstract

We explore a situation in which the target domain is accessible, but real-time data annotation is not feasible. Instead, we would like to construct an alternative training set from a large-scale data server so that a competitive model can be obtained. For this problem, because the target domain usually exhibits distinct modes (*i.e.*, semantic clusters representing data distribution), if the training set does not contain these target modes, the model performance would be compromised. While prior existing works improve algorithms iteratively, our research explores the often-overlooked potential of optimizing the structure of the data server. Inspired by the hierarchical nature of web search engines, we introduce a hierarchical data server, together with a bipartite mode matching algorithm (BMM) to align source and target modes. For each target mode, we look in the server data tree for the best mode match, which might be large or small in size. Through bipartite matching, we aim for all target modes to be optimally matched with source modes in a one-on-one fashion. Compared with existing training set search algorithms, we show that the matched server modes constitute training sets that have consistently smaller domain gaps with the target domain across object re-identification (re-ID) and detection tasks. Consequently, models trained on our searched training sets have higher accuracy than those trained otherwise. BMM allows data-centric unsupervised domain adaptation (UDA) orthogonal to existing model-centric UDA methods. By combining the BMM with existing UDA methods like pseudo-labeling, further improvement is observed.

**Code** — https://github.com/yorkeyao/BMM

## Introduction

The widespread adoption of machine learning models in real-world applications rely on the availability of extensive and well-annotated training datasets (Lin et al. 2014; Shao et al. 2019; Everingham et al. 2015). However, in certain domains with dataset bias, such as autonomous driving, medical imaging, or large-scale surveillance systems, real-time data annotation is often infeasible due to high costs, time constraints, and the need for domain expertise to deal with such data bias (Torralba and Efros 2011; Fan et al. 2018; Zhong et al. 2019; Song et al. 2020; Luo, Song, and Zhang 2020; Bai et al. 2021). An alternative approach is to leverage large-scale, pre-existing data servers to search for transfer learning training sets that can still yield high-performing models. Yet, a critical challenge remains: the target domain often comprises distinct modes in filming scenes or object appearances, and if these modes are not well-represented in the training set, model performance is likely to suffer.

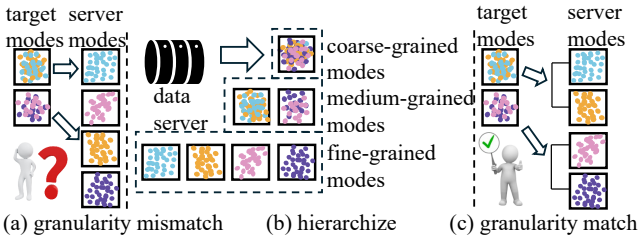


Figure 1: Motivation illustration. Our research explores the often-overlooked potential of optimizing the structure of the data server. To explain, when aligning target modes with server modes, it is often challenging (a) due to granularity mismatches. In this paper, we propose a hierarchical server design (b) that allows target modes to match source modes at varying granularities, resulting in more effective and precise alignment (c).

We are motivated by the following considerations. First, while our goal is to align server and target modes, this alignment often faces a critical challenge of granularity mismatches. As illustrated in Fig. 1, for example, if the target domain contains a mode labeled "apple", it would be inappropriate to align it only with specific modes like "real apple" or "painted apples", or with a broader category such as "fruit". A more meaningful match occurs when the semantics of the target mode and server mode align at a similar semantic level. For example, matching the target "apple" to a server mode that includes all forms of apples, whether real or depicted, is crucial for effective alignment. However, the semantics of modes are often difficult to control precisely, leading to inherent ambiguity in the alignment.

Second, existing methods in this area, albeit just a few, rely on improving algorithms that optimize data search based on distribution difference (Yao et al. 2023a; Yan,


† Corresponding author: yaoyorke@gmail.com

Acuna, and Fidler 2020; Cao et al. 2021; Tu et al. 2023) or model feedback (Ghorbani and Zou 2019). However, the potential of optimizing the structure of the data server itself is less studied. In our research, we address this gap by optimizing the architecture of data servers to better facilitate training set search. Specifically, we are inspired by web search engines, particularly the hierarchical organization used to retrieve and rank web pages efficiently. For example, Google's PageRank algorithm (Langville and Meyer 2006), constructs a hierarchical structure of websites by evaluating the importance of web pages through link analysis. This hierarchical organization is essential for effective information retrieval and ensures that the relevant content is quickly accessible.

Given these considerations, our approach introduces a hierarchical data server that organizes server data into a multi-level structure, facilitating mode matching between the source and target domains. The hierarchical structure is specifically obtained by agglomerative clustering from bottom-up to build hierarchical semantics of modes. To leverage this hierarchical architecture, we also introduce bipartite mode matching (BMM) framework. BMM involves flat clustering for the target domain to generate target modes, based on their image features extracted using a model pretrained on Imagenet (Szegedy et al. 2016). Afterwards, each target mode is linked to each server mode, with a feature-level distance between each pair of modes, specifically using the Fréchet Inception Distance (FID) (Heusel et al. 2017). Based on these connection weights, we construct a bipartite graph composed of server modes and target modes as vertices, and their corresponding FIDs as edge values. By employing minimum weight bipartite matching, we select server clusters to form an optimal training set distribution.

Experimentally, we show that the BMM, the joint use of a hierarchical data server and the bipartite matching, is superior than existing methods for training set search for object re-ID, and detection targets. We compare BMM with random selection and existing training set search techniques (Yan, Acuna, and Fidler 2020; Yao et al. 2023b) and observe that BMM leads to a consistently lower domain gap between the searched training set and target set, consequently providing the model with higher accuracy than competing methods. Additionally, we report that with further data pruning techniques, the refined training set can maintain or exceed the performance of the server data pool. Moreover, by combining the searched training set with existing unsupervised domain adaptation methods like pseudo-labelling, further improvement is observed.

## Related Work

**Unsupervised domain adaptation** (UDA) has been widely applied in fields like medical image analysis (Ke et al. 2021; Mormont, Geurts, and Marée 2018), language modeling (Conneau and Kiela 2018), and object detection (Chen et al. 2017; Dai et al. 2016; Girshick et al. 2014; Yao et al. 2023b), aiming to mitigate domain gaps by transferring knowledge from a labeled source domain to an unlabeled target domain. To enhance model generalization, various strategies have been explored (Torralba and Efros 2011; Perronnin, Sánchez, and Mensink 2010; Saenko et al. 2010; Deng et al. 2018; Lou et al. 2019), including learning domain-invariant representations (Bousmalis et al. 2016; Tzeng et al. 2017; Zhao et al. 2019) or performing source-to-target domain transformations (Hoffman et al. 2018; Lee et al. 2018; Yao et al. 2020; Sun et al. 2024). Moreover, domain generalization tackles this challenge from different angles, such as meta-learning (Balaji, Sankaranarayanan, and Chellappa 2018; Dou et al. 2019; Li et al. 2018; Zhang et al. 2020), representation learning (Khosla et al. 2012; Blanchard, Lee, and Scott 2011; Gan, Yang, and Gong 2016; Ghifary et al. 2016; Wang et al. 2020), and data augmentation. The method introduced in this paper differs from these model-centric approaches. Our BMM framework uses data-centric methods **orthogonal to** these model-centric ones, and can achieve higher accuracy when jointly used.

**Training set search from a data server** focuses on retrieving effective training samples from large-scale data pools (Yan, Acuna, and Fidler 2020; Settles 2009; Yao et al. 2023a; Douze et al. 2024). While some methods (Xu et al. 2019) use classifiers with multi-scale features to select subsets, others (Yan, Acuna, and Fidler 2020; Settles 2009) rely on pre-trained experts to assess domain gaps. The most relevant prior work, SnP (Yao et al. 2023a), searches and prunes data under budget constraints but is tailored for re-ID, whereas our method generalizes to broader tasks like object detection. Similarly, TL;DR (Wang et al. 2023) focuses on language processing and vehicle detection only. In contrast, we use a unified benchmark for broader evaluation across diverse tasks. In experiments, compared with these methods, we show that hierarchical structure and BMM leads to a consistently lower domain gap between the searched training set and target set, consequently yielding higher model accuracy than competing methods.

## Method

Our goal is to build a labeled dataset from the source data pool for a given unlabeled target dataset, ensuring the differences in data biases compared to the target domain are as small as possible. This approach enables a model trained using the source dataset to perform effectively on the target dataset. In order to build the desired training dataset successfully, we propose the BMM framework.

### Problem Description

We follow the search and pruning framework proposed in Yao et al. (2023a), with a special focus on the search part. The target dataset is defined as $\mathcal{D}_T = \{(\boldsymbol{x}_i, y_i)\}_{i\in[m_t]}$, with $m_t$ representing the total number of image-label combinations in the target dataset, denoted by $[m_t] = \{1, 2, \ldots, m_t\}$. This set conforms to the distribution $p_T$, *i.e.*, $\mathcal{D}_T \sim p_T$.

In order to create the training dataset $\mathcal{D}_S$, we establish a source data pool $\mathcal{S}$ comprising a variety of datasets or domains. This is expressed as $\mathcal{S} = \mathcal{D}_S^1 \bigcup \mathcal{D}_S^2 \cdots \bigcup \mathcal{D}_S^K$, where each $\mathcal{D}_S^k$ for $k \in [K]$ denotes the $k$-th source dataset. Ide-

We use "cluster" and "mode" interchangeably in this paper as modes are obtained using clustering.

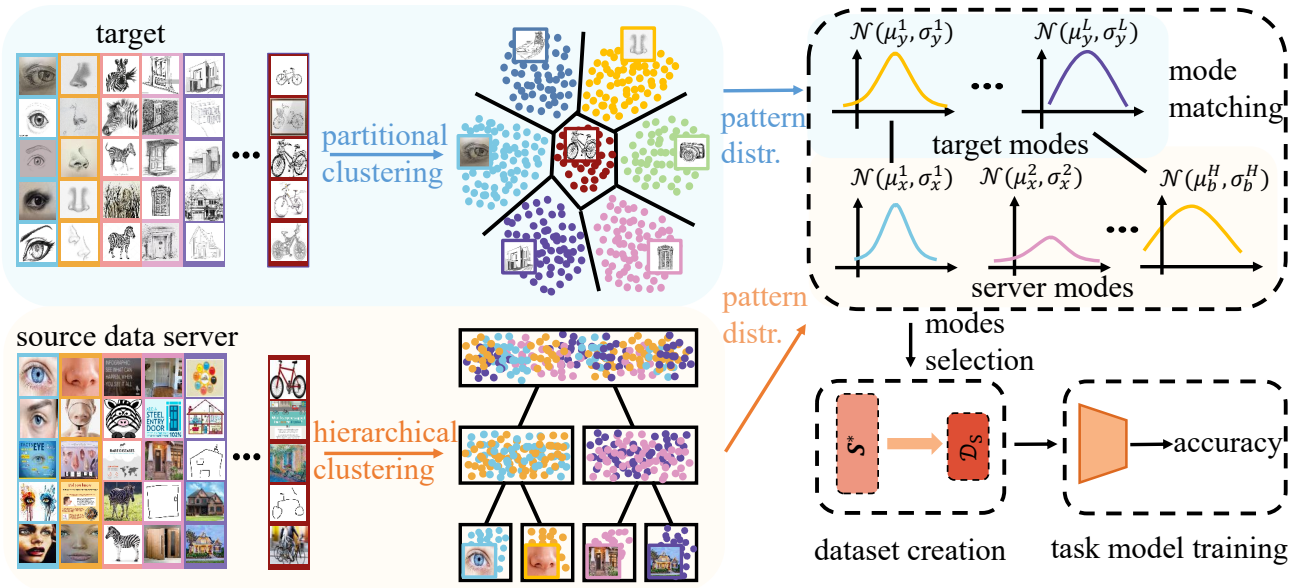


Figure 2: Workflow of BMM. **(Top left):** For a given target, we extract modes using flat clustering (Lloyd 1982; MacQueen et al. 1967). **(Bottom left):** For our data server, we extract modes using hierarchical clustering (Müllner 2011). For modes existing in both target and the data server, we align them using the bipartite graph matching algorithm, *i.e.*, Hungarian algorithm (Kuhn 1955; Munkres 1957). For source modes aligning to the target, we select them to form our searched training set. The searched training set can be further pruned and then used for model training.

Algorithm 1: Bipartite Mode Matching for Training Set Search

**Input:** data server $\mathcal{S}$, target set $\mathcal{D}_T$, and number of source clusters $J$ and the number of target clusters $L$.
**Begin:**
Balanced Cluster $(\mathcal{S}, J) \longrightarrow \{\mathbf{C}^1, \cdots, \mathbf{C}^J\}$
$\{\mathbf{C}^1, \cdots, \mathbf{C}^J\} \longrightarrow \{\mathbf{S}^1, \cdots, \mathbf{S}^H\}$ ▷ Hier. clustering
Cluster $(\mathcal{D}_T, L) \longrightarrow \{\mathbf{T}^1, \cdots, \mathbf{T}^L\}$
$V = \{\mathbf{S}^1, \cdots, \mathbf{S}^H, \mathbf{T}^1, \cdots, \mathbf{T}^L\}$, $E = \emptyset$ ▷ Graph init.
**for** $x$ in 1 to $H$ **do**
  **for** $y$ in 1 to $L$ **do**
    $edge(\mathbf{S}^x, \mathbf{T}^y).value = \text{FID}(\mathbf{S}^x, \mathbf{T}^y)$
    $E = E \cup edge(\mathbf{S}^x, \mathbf{T}^y)$
$\sigma^*$ = Hungarian $(G(V, E))$ ▷ bipartite graph matching
$\mathbf{S}^* = \emptyset$
**for** $y$ in 1 to $L$ **do**
  **if** $\mathbf{S}^{\sigma^*(y)} \notin \mathbf{S}^*$ **then**
    $\mathbf{S}^* = \mathbf{S}^* \cup \mathbf{S}^{\sigma^*(y)}$
**return** $\mathbf{S}^*$

ally, we hope to create a subset $\mathbf{S}^*$ from the data server $\mathcal{S}$ through a sampling process. Consider $h_{\mathbf{S}}$ as the model that is trained on any given dataset $\mathbf{S}$. The risk $h_{\mathbf{S}}$ of prediction on the test sample $\boldsymbol{x}$ which has the ground truth label $y$ is calculated as $\ell(h_{\mathbf{S}}(\boldsymbol{x}), y)$. $\mathbf{S}^*$ is constructed with the aim of guaranteeing that the model $h_{\mathbf{S}^*}$ demonstrates the least risk on $\mathcal{D}_T$, *i.e.*,

$$\mathbf{S}^* = \arg\min_{\mathbf{S} \in 2^{\mathcal{S}}} \mathbb{E}_{\boldsymbol{x}, y \sim p_T}[\ell(h_{\mathbf{S}}(\boldsymbol{x}), y)]. \quad (1)$$

As analysed in Yao et al. (2023a), it is common for our target dataset $\mathcal{D}_T$ to have dataset bias (modes). In this case, if the training set cannot reflect them, the presence of such a difference introduces a hurdle to the efficacy of training, as models may not generalize effectively to real-world scenarios. In this paper, shown in Algorithm 1, we propose the BMM, to search for a training set adapted to target bias.

## Hierarchical Data Server

We are motivated by the notion that dataset bias can be represented by clustered modes (Tu et al. 2023). Given our data server, image features are extracted to create a sample space, and we then conduct clustering within this feature space.

Specifically, we utilize a feature extractor $\boldsymbol{F}(\cdot)$ which converts an input image into a d-dimensional feature represented as $f \in \mathbb{R}^d$. Such a feature extractor, when pre-trained on Imagenet (Szegedy et al. 2016), is able to extract essential semantic features of $\mathcal{F}$ through efficiently condensing the image features into a refined representation. Formally, with $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i \in [m_s]}$ representing the aggregate of $K$ datasets or domains and $m_s$ indicating the overall count of images after merging, we extract the features of all these images, donated as $\mathcal{F}_s = \{\boldsymbol{F}(\boldsymbol{x}_i)\}_{i \in [m_s]}$. From these extracted features, a mode structure is established. Following this, we proposed to use balanced k-means clustering (Lloyd 1982; MacQueen et al. 1967) to divide the feature space $\mathbb{R}^d$ into J clusters, *i.e.*, $\{\mathbf{C}^1, \cdots, \mathbf{C}^J\}$. Each cluster center among the J clusters is named as a "mode", since each center usually represents a distinct semantic content. Fig. 2 shows representative images for each mode. Thus formally, we aim to

minimize the Sum of Squared Errors (SSE):

$$SSE = \sum_{k=1}^{J} \sum_{x_i \in C_k} \|x_i - \theta_k\|^2 \tag{2}$$

which has the constraint that

$$|C_k| = \frac{m_t}{K}, \quad \forall k \tag{3}$$

where $\theta_k$ means a cluster center and $SSE$ is the final loss. Subsequently, we merge clusters step by step according to the bottom-up hierarchical structure, *i.e.*, agglomerative clustering (Müllner 2011). Specifically, we start with each element in $\{\mathbf{C}^1, \cdots, \mathbf{C}^J\}$, and pairs of closest clusters are merged as we move up the hierarchy. The process continues until all images are grouped into a single cluster, resulting in a dendrogram, *i.e.*, a tree-like diagram illustrating the formation of clusters at each step. For all clusters with different sizes found during this process, we view them as distinct modes existing in the data server. By this process, we get $H$ modes, where finally we get $\{\mathbf{S}^1, \cdots, \mathbf{S}^H\}$. Please note that $H$ is not a hyperparameter but a resulting number after the hierarchical merging of $J$ clusters. For example, in the case where the hierarchical structure is a full binary tree, the number of all nodes (modes) $H$ is approximately twice the number of all leaf nodes $J$, following $H = 2J - 1$.

A key of our method is the usage of balanced K-means rather than normal K-means. In K-means, mode sizes in the same hierarchical level may vary, as K-Means does not have constraints on this. That is, each mode at a given level does not necessarily contain the same number of images, which can lead to suboptimal mode matching in the following step when the mode size distribution follows a long-tail pattern. We replace K-Means with constrained K-Means, ensuring consistent mode sizes in a same level. Tab. 1 shows that this simple yet task-aware refinement improves performance, highlighting the potential of further optimizing hierarchical structure construction.

### Target-server Mode Matching

For each distinct mode existing in the target domain, our goal is to search for its corresponding similar modes in the source data pool. This can be solved by the algorithm designed for the assignment problem (Kuhn 1955).

We extract image features and perform flat clustering on $\mathcal{D}_T$ to get $L$ clusters $\{\mathbf{T}^1, \cdots, \mathbf{T}^L\}$. The flat clustering can help to find distinct modes in $\mathcal{D}_T$, and ensure all modes have enough dissimilarity with each other.

The assignment problem can be instantiated as a complete undirected bipartite graph, where each edge is assigned a non-negative cost value. This is designed to precisely determine the cost associated with each link, thereby effectively addressing the assignment problem. In training set search, the undirected bipartite graph is represented as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, with the vertex set $\boldsymbol{V}$ comprising the union of two non-intersecting sets, $\boldsymbol{X} = \{\mathbf{S}^1, \cdots, \mathbf{S}^H\}$ (*i.e.*, the sources modes) and $\boldsymbol{Y} = \{\mathbf{T}^1, \cdots, \mathbf{T}^L\}$ (*i.e.*, the target modes), such that $H > L$; In practice, we usually set the number of $H$ to two orders larger than $L$. The set E of edges contains every two-element subset $\{\mathbf{S}^x, \mathbf{T}^y\}$, with $\mathbf{S}^x \in \boldsymbol{X}$ and $\mathbf{T}^y \in \boldsymbol{Y}$. The cost associated with edge $e = \{\mathbf{S}^x, \mathbf{T}^y\}$, namely $c(\mathbf{S}^x, \mathbf{T}^y)$, is determined using Fréchet Inception Distance (FID) (Heusel et al. 2017) between them, which is defined as:

$$\text{FID}(x, y) = \|\boldsymbol{\mu}_x - \boldsymbol{\mu}_y\|_2^2 + \text{Tr}(\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y - 2(\boldsymbol{\Sigma}_x \boldsymbol{\Sigma}_y)^{\frac{1}{2}}). \tag{4}$$

In Eq. 4, $\boldsymbol{\mu}_x \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}_x \in \mathbb{R}^{d \times d}$ represent the mean and covariance matrix of the image descriptors for cluster $\mathbf{S}^x$, respectively. The mean and covariance matrix for cluster $\mathbf{T}^y$ are denoted by $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$. The function $\text{Tr}(.)$ denotes the trace of a square matrix, and $d$ refers to the dimension of the image descriptors. Given $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, to find a bipartite matching between these two sets $\boldsymbol{X}$ and $\boldsymbol{Y}$, we search for a permutation of L elements $\sigma \in \mathfrak{S}_L$ with the lowest overall cost of all matching:

$$\sigma^* = \arg\min_{\sigma \in \mathfrak{S}_L} \sum_{i}^{L} \text{FID}(\mathbf{T}^i, \mathbf{S}^{\sigma(i)}), \tag{5}$$

where $\text{FID}(\mathbf{T}^i, \mathbf{S}^{\sigma(i)})$ refers to the matching cost between the original prediction and the perturbed prediction corresponding to index $i$ in $\{\mathbf{T}^1, \cdots, \mathbf{T}^L\}$ and index $\sigma(i)$ in $\{\mathbf{S}^1, \cdots, \mathbf{S}^H\}$. This optimal pairing is efficiently calculated using the Hungarian algorithm following previous work (Kuhn 1955; Munkres 1957).

After this process, we can get our searched training set $\mathbf{S}^* = \{\mathbf{S}^{\sigma(i)}\}_{i=1}^{L}$, means the combination all matched source clusters. Note that this process is also likely to have repeat selection of data. Thus, during the combination process, we delete those which previously appeared. After searching, we can use dataset pruning techniques to further reduce the training set size to meet our requirement (Yao et al. 2023a), resulting in $\mathcal{D}_S$ for model training.

## Experiment

### Server and Target Datasets

For a dataset denoted as a target domain, their unlabeled training sets are used as the target. Otherwise indicated, we use direct transfer models in this paper, where we train models on searched training sets and test on targets directly. Unless otherwise specified, for object re-ID, we use ID-discriminative embedding (IDE) (Zheng et al. 2016). For detection, we use RetinaNet (Lin et al. 2017).

**Object re-ID.** We conducted experiments using both the person re-ID dataset and the vehicle re-ID dataset separately, reusing settings from (Yao et al. 2023a).

**Vehicle Detection.** The data server is composed of seven datasets, including ADE20K (Zhou et al. 2019), COCO (Lin et al. 2014), BDD (Yu et al. 2020), CityScapes (Cordts et al. 2016), DETRAC (Wen et al. 2020; Lyu et al. 2017), Kitti (Geiger, Lenz, and Urtasun 2012) and VOC (Everingham et al. 2015). The data server has 176,491 images in total. We first use Exdark (Loh and Chan 2019), which is captured only in low-light environments and has 7,363 images, as our target. We also use the Region100 benchmark as our target, which was used as the benchmark dataset in the 2nd

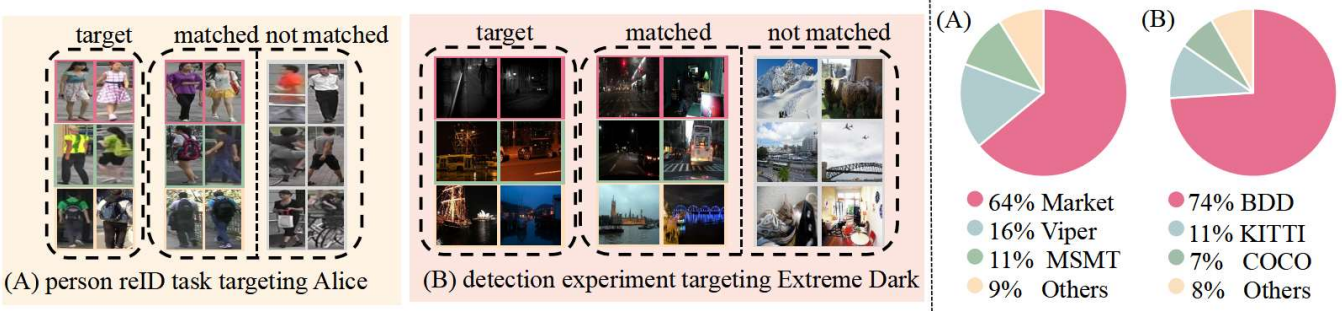


Figure 3: Mode matching examples (**Left**) and composition of the searched training set (**Right**). (**Left**): We show matched data server modes (images) resembling target modes (images). Four sub-figures (A) and (B) show mode matching examples for person re-ID, and vehicle detection respectively. For example, (B) shows a successful mode matching a dark environment. It is such successful mode matching that ensures the searched training set has a similar distribution to the target, and thus high training efficacy. (**Right:**) The pie charts on the right illustrate the proportions of the searched training set.

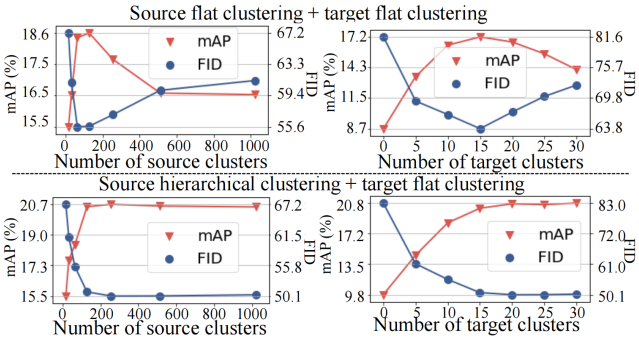


Figure 4: The comparison of source server flat clustering and source hierarchical clustering. We show the impact of the number of source and target clusters on model performance. The target cluster number is fixed to 20 when varying source cluster numbers, and the source cluster number is fixed to 128 when varying target cluster numbers. Source flat clustering requires hyperparameter tuning for the "sweet point" (shown in **top left**) and **top right**, while hierarchical clustering does not (shown in **bottom left**). Furthermore, **bottom right** shows $L > 20$ and **bottomleft** shows $> 200$ source clusters (*i.e.*, depth > 8) suffice for high accuracy.

CVPR DataCV Challenge. The Region100 benchmark consists of footage captured by static cameras from 100 regions in the real world. For videos from each different region, the first 70% is used for model training, while the remaining 30% is designated for validation and testing.

**Evaluation protocol.** For object re-ID, we report mAP and CMC ("rank-1" and "rank-5"). In vehicle detection, mAP, mAP@50, and mAP@75 evaluate overall and threshold-specific detection accuracy at IoU of 0.5 and 0.75.

## Results and Discussion

**Working mechanism of BMM.** Shown in Fig. 3, in the process of BMM, we aim to align each distinct mode found in a target dataset with a corresponding one in the data server. This alignment ensures that both datasets contain similar modes, effectively reducing the bias difference that is the domain gap between the source data pool and target domain. As proposed by Yao et al. (2023a), the closer the distribution of the source dataset is to that of the target, the more effective the training will be. Therefore, BMM facilitates the creation of a high-quality, target-specific training set by closely matching its distribution with the target. In many cases, it is very likely that we may not find a perfect semantic match for each mode in the server. However, our algorithm *still strives to find the closest possible match* due to the minimization nature of the optimization objective shown in Eq. 5. Thus, in these situations, we cab still reduce the domain gap and improve performance.

| Method | FID↓ | mAP↑ |
|---|---|---|
| Baseline | 51.93 | 26.08 |
| Balanced | **50.48** | **28.16** |

Table 1: Comparison to the balanced K-means. Settings are the same as Tab. 3 with 5% pruning rate targeting AlicePerson.

| Method | FID↓ | SSIM↑ | mAP↑ |
|---|---|---|---|
| DM w. dup. | **51.07** | 15.85 | 20.14 |
| DM w./o. dup. | 60.84 | **21.07** | 22.07 |
| BMM | 51.93 | 20.45 | **26.08** |

Table 2: Comparison between our method and direct match (DM). Settings are the same as Tab. 3 with 5% pruning rate targeting AlicePerson.

**Benefits of using hierarchical clustering on data server.** As mentioned in Fig. 1, we use hierarchical clustering rather than flat clustering to minimize the impact of mode granularity (*e.g.*, the semantic granularity of each cluster found by clustering) during matching processes. Our objective is to enable each target mode to match with the most similar modes in semantics from the data server, irrespective of source server mode size, which can be controlled by the hyperparameter: number of source clusters.

As shown in Fig. 4, if we use flat clustering on the source, we will need to carefully adjust both the number of source clusters and the number of target clusters to ensure competent matching between them, thereby reaching the "sweet point" as shown in Fig. 4 upper part. Such a "sweet point" shows that at such a specific source and target mode size, source modes and target modes can be relatively competently matched, thus resulting in a competent training set searched. However, finding such a "sweet point" requires significant effort in hyperparameter tuning and is not practical in implementation.

In comparison, as shown in Fig. 4 bottom, as long as the

| Training data | | | Target domains in person re-ID | | | | | | Target domains in vehicle re-ID | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | AlicePerson | | | Market | | | AliceVehicle | | | VeRi | | |
| | | | FID↓ | R1↑ | mAP↑ | FID↓ | R1↑ | mAP↑ | FID↓ | R1↑ | mAP↑ | FID↓ | R1↑ | mAP↑ |
| data server | | | 81.67 | 38.96 | 17.62 | 37.53 | 55.55 | 30.62 | 43.95 | 30.47 | 14.64 | 24.39 | 55.90 | 25.03 |
| Searched via BMM | | | 51.08 | 50.63 | 27.09 | 27.68 | 60.57 | 36.05 | 21.27 | 47.84 | 27.24 | 15.32 | 77.19 | 42.18 |
| Pruning | 5% IDs | Random | 81.41 | 33.16 | 14.49 | 39.65 | 47.39 | 23.97 | 44.52 | 36.36 | 14.17 | 25.27 | 70.38 | 30.44 |
| | | NDS (Yan, Acuna, and Fidler 2020) | 61.01 | 44.63 | 22.81 | 31.63 | 49.17 | 24.77 | 32.48 | 41.32 | 17.77 | 26.06 | 71.23 | 32.53 |
| | | SnP (Yao et al. 2023a) | 60.64 | 47.26 | 25.45 | 30.37 | 51.96 | 26.56 | 23.92 | 44.58 | 21.79 | 18.09 | 72.05 | 36.01 |
| | | TL;DR (Wang et al. 2023) | 62.98 | 43.08 | 21.95 | 32.08 | 48.56 | 23.04 | 33.25 | 40.98 | 17.57 | 25.98 | 71.23 | 32.53 |
| | | CCDR (Chang et al. 2024) | 60.52 | 48.47 | 25.04 | 31.07 | 50.87 | 27.08 | 24.04 | 43.68 | 21.05 | 17.89 | 71.58 | 35.85 |
| | | BMM | **51.93** | **49.28** | **26.08** | **27.05** | **53.03** | **28.39** | **21.96** | **45.08** | **23.84** | **15.98** | **72.69** | **38.55** |
| | 20% IDs | Random | 79.33 | 38.10 | 17.79 | 38.63 | 53.15 | 28.39 | 43.90 | 40.89 | 18.13 | 24.43 | 68.71 | 34.10 |
| | | NDS (Yan, Acuna, and Fidler 2020) | 63.15 | 46.74 | 22.65 | 32.42 | 53.53 | 28.19 | 24.15 | 44.58 | 22.82 | 18.74 | 71.04 | 38.07 |
| | | SnP (Yao et al. 2023a) | 61.87 | 47.20 | 25.36 | 30.58 | 57.14 | 33.09 | 23.47 | 46.07 | 25.24 | 17.93 | 73.48 | 40.75 |
| | | TL;DR (Wang et al. 2023) | 65.12 | 45.48 | 21.20 | 33.87 | 51.08 | 26.48 | 25.08 | 43.05 | 21.08 | 19.08 | 70.65 | 37.90 |
| | | CCDR (Chang et al. 2024) | 61.02 | 47.98 | 25.08 | 31.05 | 56.85 | 33.69 | 23.78 | 46.07 | 25.85 | 17.30 | 73.78 | 41.07 |
| | | BMM | **51.53** | **49.68** | **26.97** | **27.54** | **60.49** | **35.08** | **21.64** | **47.34** | **26.18** | **15.72** | **75.36** | **42.05** |

Table 3: Comparing different methods in training data search: random sampling, greedy sampling, greedy search, and proposed BMM, in **object re-ID tasks**. We set the pruning rate as $5\%$ and $20\%$ of the total source IDs. We use four targets: AlicePerson, Market, AliceVehicle and VeRi. The task model is IDE (Zheng et al. 2016). FID, rank-1 (%), and mAP (%) are reported.

| Training data | | | ExDark | | | | Region 100 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | FID↓ | mAP↑ | mAP@50↑ | mAP@75↑ | FID↓ | mAP↑ | mAP@50↑ | mAP@75↑ |
| data server | | | 104.98 | 40.43 | 79.56 | 36.71 | 251.47 | 19.65 | 41.38 | 17.25 |
| Searched via BMM | | | 56.95 | 45.36 | 82.71 | 42.31 | 140.82 | 23.52 | 50.21 | 18.25 |
| Pruning | 5% Imgs | Random | 105.74 | 23.50 | 56.13 | 14.49 | 251.90 | 11.38 | 25.30 | 9.14 |
| | | NDS (Yan, Acuna, and Fidler 2020) | 64.35 | 30.34 | 65.34 | 22.18 | 165.31 | 12.62 | 25.34 | 9.00 |
| | | SnP (Yao et al. 2023a) | 59.78 | 32.15 | 67.18 | 25.69 | 153.82 | 15.07 | 34.65 | 11.34 |
| | | TL;DR (Wang et al. 2023) | 66.25 | 28.32 | 65.11 | 22.07 | 162.34 | 12.87 | 25.08 | 9.08 |
| | | CCDR (Chang et al. 2024) | 57.98 | 32.96 | 67.67 | 25.03 | 142.48 | 22.25 | 44.49 | **19.68** |
| | | BMM | **56.34** | **34.83** | **76.57** | **25.83** | **140.07** | **23.08** | **46.34** | 18.08 |

Table 4: Comparing different methods in training data search: random sampling, greedy sampling, greedy search, and proposed BMM, in the **vehicle detection task**. We set the pruning ratio as $5\%$. The task model is RetinaNet (Lin et al. 2017). Two targets are used: Exdark and Region100. We report FID, mAP (%), mAP@50 (%) and mAP@75 (%) respectively.

number of source clusters is not considerably small, performance will not be compromised. The reason is that if we use hierarchical clustering on the source, we prepare hierarchical modes with different levels of semantics (mode sizes) in our data server in preparation for being matched to the target. Thus source mode size will not be an influential factor of performance anymore.

Further, compared with hierarchical clustering on the source, flat clustering results in higher FID and lower mAP, showing poor matching between target and source modes.

**Necessity of using flat clustering on target data.** In the BMM framework, our goal is to match each distinct target mode with a corresponding source mode. Using flat clustering allows us to efficiently identify distinct modes, ensuring minimal correlation between them. If hierarchical clustering were applied to the target data, higher-level modes would strongly correlate with lower-level ones, complicating the task of selecting appropriate modes from redundant ones for constructing the data server. Thus, flat clustering is necessary for effective mode matching.

**The rationale for using Hungarian matching instead of a simpler greedy direct matching.** Compared with direct matching (DM), the Hungarian matching algorithm prevents multiple target modes from matching the **same** source mode, which, though rare, occurs in practice (*e.g.*, 4 out of 20 cases when targeting AlicerPerson). As shown in Tab. 2, direct match allowing duplicate matches keeps FID low but reduces data diversity (measured in SSIM averaged across datasets), harming performance. Simply deleting these duplicates leaves some target modes unmatched, degrading FID and model performance. Bipartite graph matching ensures one-to-one mode assignment, balancing diversity and domain gap, thus improving results.

**Computing time complexity analysis.** Considering $L$ target clusters and $J$ source clusters obtained, and $J >> L$. The overall time complexity of our BMM framework is $\mathcal{O}(J^3)$, including $\mathcal{O}(L^2)$ for target flat clustering, $\mathcal{O}(J^2 log J)$ for source hierarchical clustering, and for matching. Please note that the hierarchical process is analogous to an offline training phase and only needs to be done once. For different target domains, we do not need to reconstruct the hierarchical data server each time. Our mode-matching algorithm, which is applied for each target domain, has a comparatively lower time complexity ($\mathcal{O}(logJ * J * L)$). This ensures that the computational cost remains manageable during deployment, as the expensive hierarchical structure construction is a one-time process that only happens in preprocessing.

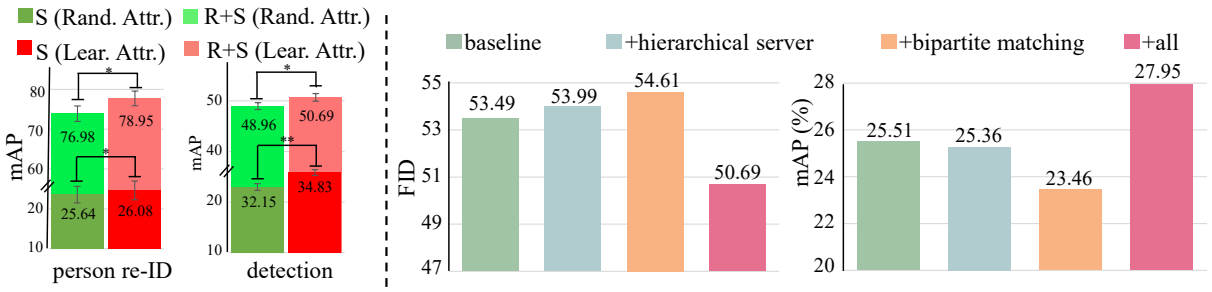


Figure 5: **Left:** Joint usage of the training set search and existing UDA (*i.e.*, pseudo-labelling) methods. MMT (Ge, Chen, and Li 2020) and AT (Li et al. 2022) are used for re-ID and detection respectively. **Right:** Ablation study of the hierarchical server and BMM framework (target Aliceperson). Starting from the baseline, we gradually add the hierarchical design and the mode matching module, observing consistent performance improvements.

**BMM framework *vs.* random sample and existing training set search algorithms.** We have this analysis in Table 3 and Table 4. To make a fair comparison and eliminate the influence of dataset size, we searched for subsets of equal size via uniform sampling. For the comparison method, SnP was originally designed for re-ID only, we extended their method to detection (Yao et al. 2023a). TL;DR was originally designed for language only, we extended their method to vision tasks and excluded TL;DR's data generation, and evaluated only its data search component (Wang et al. 2023). CCDR was designed for object detection, we extended it to person re-identification (Chang et al. 2024). Experiments on different tasks have shown a significant reduction in the domain gap between the searched dataset compared to the source server and the target domain, and an increase in the accuracy of the model.

For person re-ID on AlicePerson with a 5% pruning ratio, random sampling yields an FID of 81.41, rank-1 accuracy of 33.16%, and mAP of 14.49%, whereas BMM reduces the FID by 29 and improves rank-1 accuracy and mAP by 16% and 12%, respectively. Similar gains are observed on vehicle re-ID. Compared with NDS (Yan, Acuna, and Fidler 2020), SnP (Yao et al. 2023a), TL;DR (Wang et al. 2023), and CCDR (Chang et al. 2024), BMM achieves the best domain alignment and detection performance, reaching an FID of 56.26, mAP of 42.23%, and mAP@50 of 81.69%. The improvement over the greedy search in SnP is statistically significant and stable (Fig. 5). Pseudo labels from UDA further enhance performance.

**Joint usage of the training set search and existing UDA methods yields higher accuracy.** The training set search methods are orthogonal to existing UDA methods, *e.g.*, pseudo labels obtained by the latter can further augment our searched training set to achieve higher accuracy. As shown in Fig. 5 left, our method, when jointly used together with pseudo-labeling methods Mutual Mean-Teaching (MMT) (Ge, Chen, and Li 2020) and Adaptive Teacher (AT) (Li et al. 2022), yields much higher accuracy for re-ID (targeting Market), and detection, respectively. For example, the accuracy of using MMT increases from 26.08% to 78.95% compared to only using the searched training set for direct transfer. When joint training with MMT, our method increases from 76.98% to 78.95% compared to greedy search.

**Ablation study.** We conduct an ablation study of the proposed modules, as shown in Fig. 5 right. Our baseline uses flat clustering in the data server and a greedy search strategy to select clusters (Yao et al. 2023a). Replacing flat clustering with hierarchical clustering alone does not lead to noticeable improvement. Similarly, using the proposed mode-matching module in place of greedy search, while keeping flat clustering, also yields no performance gain. These results are expected—our system is designed for hierarchical clustering and mode matching to work in tandem. Only when both modules are used together do we observe significant improvements over the baseline, confirming the effectiveness of our joint design. The composition of searched training sets is shown in Fig. 3.

**Analysis of hyperparameters.** Fig. 4 (bottom) shows the relationship between the number of clusters in the source ($J$) and target ($L$) domain clustering, mAP, and FID metrics. As $J$ and $L$ increase, the domain gap (FID) and mAP improve. In the source domain, accuracy stabilizes at $J = 128$. In the target domain, accuracy plateaus at $L = 10$, and the FID reaches its lowest value of 9.8 at $L = 20$, with mAP at 83.0%. As long as the cluster numbers are not too small, performance remains stable.

## Conclusion

In this paper, we focus on the training set search problem from a data server, for object re-ID and detection tasks, with a specific focus on the structure of the source data server. We propose a hierarchical data server and BMM framework, to make the searched training set have a similar distribution with the target, such as styles and class distributions. We show that the matched source modes constitute training sets that have consistently better mode matching and smaller domain gap with the target domain. Experiments show that the BMM outperforms existing training set search methods. Furthermore, we analyze various components in the BMM system and find them to be stable under various data servers and targets, and hyperparameters.

## Acknowledgments

This work was supported in part by the Key Research and Development Program of Shandong Province China (2025CXGC010901), the Shandong Province Overseas Young Talents Program, the ARC Discovery Project (DP210102801), Oracle Cloud credits, and related resources provided by Oracle for Research.

## References

Bai, Y.; Jiao, J.; Ce, W.; Liu, J.; Lou, Y.; Feng, X.; and Duan, L.-Y. 2021. Person30k: A dual-meta generalization network for person re-identification. In *CVPR*, 2123–2132.

Balaji, Y.; Sankaranarayanan, S.; and Chellappa, R. 2018. Metareg: Towards domain generalization using meta-regularization. *NeurIPS*, 31.

Blanchard, G.; Lee, G.; and Scott, C. 2011. Generalizing from several related classification tasks to a new unlabeled sample. *NeurIPS*, 24.

Bousmalis, K.; Trigeorgis, G.; Silberman, N.; Krishnan, D.; and Erhan, D. 2016. Domain separation networks. *NeurIPS*, 29.

Cao, T.; Doubov, S. A.; Acuna, D.; and Fidler, S. 2021. Scalable Neural Data Server: A Data Recommender for Transfer Learning. *NeurIPS*, 34: 8984–8997.

Chang, C.; Long, K.; Li, Z.; and Rai, H. 2024. Classifier Guided Cluster Density Reduction for Dataset Selection. In *CVPR*, 7338–7347.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4): 834–848.

Conneau, A.; and Kiela, D. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 3213–3223.

Dai, J.; Li, Y.; He, K.; and Sun, J. 2016. R-fcn: Object detection via region-based fully convolutional networks. *NeurIPS*, 29.

Deng, W.; Zheng, L.; Ye, Q.; Kang, G.; Yang, Y.; and Jiao, J. 2018. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *CVPR*.

Dou, Q.; Coelho de Castro, D.; Kamnitsas, K.; and Glocker, B. 2019. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, volume 32.

Douze, M.; Guzhva, A.; Deng, C.; Johnson, J.; Szilvasy, G.; Mazaré, P.-E.; Lomeli, M.; Hosseini, L.; and Jégou, H. 2024. The Faiss library.

Everingham, M.; Eslami, S. A.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. *IJCV*, 111: 98–136.

Fan, H.; Zheng, L.; Yan, C.; and Yang, Y. 2018. Unsupervised person re-identification: Clustering and fine-tuning. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 14(4): 1–18.

Gan, C.; Yang, T.; and Gong, B. 2016. Learning attributes equals multi-source domain generalization. In *CVPR*, 87–97.

Ge, Y.; Chen, D.; and Li, H. 2020. Mutual Mean-Teaching: Pseudo Label Refinery for Unsupervised Domain Adaptation on Person Re-identification. In *ICLR*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

Ghifary, M.; Balduzzi, D.; Kleijn, W. B.; and Zhang, M. 2016. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7): 1414–1430.

Ghorbani, A.; and Zou, J. 2019. Data shapley: Equitable valuation of data for machine learning. In *ICML*, 2242–2251. PMLR.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 580–587.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 1989–1998. Pmlr.

Ke, A.; Ellsworth, W.; Banerjee, O.; Ng, A. Y.; and Rajpurkar, P. 2021. CheXtransfer: performance and parameter efficiency of ImageNet models for chest X-Ray interpretation. In *CHIL*, 116–124.

Khosla, A.; Zhou, T.; Malisiewicz, T.; Efros, A. A.; and Torralba, A. 2012. Undoing the damage of dataset bias. In *Computer Vision–ECCV 2012: 12th ECCV, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12*, 158–171. Springer.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83–97.

Langville, A. N.; and Meyer, C. D. 2006. *Google's PageRank and beyond: The science of search engine rankings*. Princeton university press.

Lee, H.-Y.; Tseng, H.-Y.; Huang, J.-B.; Singh, M.; and Yang, M.-H. 2018. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, 35–51.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. 2018. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, volume 32.

Li, Y.-J.; Dai, X.; Ma, C.-Y.; Liu, Y.-C.; Chen, K.; Wu, B.; He, Z.; Kitani, K.; and Vajda, P. 2022. Cross-domain adaptive teacher for object detection. In *CVPR*, 7581–7590.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *ICCV*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2): 129–137.

Loh, Y. P.; and Chan, C. S. 2019. Getting to Know Low-light Images with The Exclusively Dark Dataset. *CVIU2*, 178: 30–42.

Lou, Y.; Bai, Y.; Liu, J.; Wang, S.; and Duan, L.-Y. 2019. Embedding adversarial learning for vehicle re-identification. *TIP*, 28(8): 3794–3807.

Luo, C.; Song, C.; and Zhang, Z. 2020. Generalizing person re-identification by camera-aware invariance learning and cross-domain mixup. In *ECCV*, 224–241. Springer.

Lyu, S.; Chang, M.-C.; Du, D.; Wen, L.; Qi, H.; Li, Y.; Wei, Y.; Ke, L.; Hu, T.; Del Coco, M.; et al. 2017. UA-DETRAC 2017: Report of AVSS2017 & IWT4S challenge on advanced traffic monitoring. In *AVSS*, 1–7. IEEE.

MacQueen, J.; et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.

Mormont, R.; Geurts, P.; and Marée, R. 2018. Comparison of deep transfer learning strategies for digital pathology. In *CVPR workshops*, 2262–2271.

Müllner, D. 2011. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*.

Munkres, J. 1957. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1): 32–38.

Perronnin, F.; Sánchez, J.; and Mensink, T. 2010. Improving the fisher kernel for large-scale image classification. In *ECCV*.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *ECCV*.

Settles, B. 2009. Active learning literature survey.

Shao, S.; Li, Z.; Zhang, T.; Peng, C.; Yu, G.; Zhang, X.; Li, J.; and Sun, J. 2019. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 8430–8439.

Song, L.; Xu, Y.; Zhang, L.; Du, B.; Zhang, Q.; and Wang, X. 2020. Learning from Synthetic Images via Active Pseudo-Labeling. *TIP*.

Sun, X.; Yao, Y.; Wang, S.; Li, H.; and Zheng, L. 2024. Alice Benchmarks: Connecting Real World Re-Identification with the Synthetic. In *ICLR*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *CVPR*.

Torralba, A.; and Efros, A. A. 2011. Unbiased look at dataset bias. In *ECCV*.

Tu, W.; Deng, W.; Gedeon, T.; and Zheng, L. 2023. A Bag-of-Prototypes Representation for Dataset-Level Applications. In *CVPR*, 2881–2892.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *CVPR*, 7167–7176.

Wang, A. J.; Lin, K. Q.; Zhang, D. J.; Lei, S. W.; and Shou, M. Z. 2023. Too large; data reduction for vision-language pre-training. In *ICCV*, 3147–3157.

Wang, G.; Han, H.; Shan, S.; and Chen, X. 2020. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *CVPR*, 6678–6687.

Wen, L.; Du, D.; Cai, Z.; Lei, Z.; Chang, M.-C.; Qi, H.; Lim, J.; Yang, M.-H.; and Lyu, S. 2020. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *CVIU2*, 193: 102907.

Xu, Y.; Wang, Y.; Chen, H.; Han, K.; Xu, C.; Tao, D.; and Xu, C. 2019. Positive-unlabeled compression on the cloud. *NeurIPS*, 32.

Yan, X.; Acuna, D.; and Fidler, S. 2020. Neural data server: A large-scale search engine for transfer learning data. In *CVPR*, 3893–3902.

Yao, Y.; Lei, H.; Gedeon, T.; and Zheng, L. 2023a. Large-scale Training Data Search for Object Re-identification. In *CVPR*, 15568–15578.

Yao, Y.; Zheng, L.; Yang, X.; Naphade, M.; and Gedeon, T. 2020. Simulating Content Consistent Vehicle Datasets with Attribute Descent. In *ECCV*.

Yao, Y.; Zheng, L.; Yang, X.; Napthade, M.; and Gedeon, T. 2023b. Attribute Descent: Simulating Object-Centric Datasets on the Content Level and Beyond. *TPAMI*.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2636–2645.

Zhang, M. M.; Marklund, H.; Dhawan, N.; Gupta, A.; Levine, S.; and Finn, C. 2020. Adaptive risk minimization: A meta-learning approach for tackling group shift.

Zhao, H.; Des Combes, R. T.; Zhang, K.; and Gordon, G. 2019. On learning invariant representations for domain adaptation. In *ICML*, 7523–7532. PMLR.

Zheng, L.; Bie, Z.; Sun, Y.; Wang, J.; Su, C.; Wang, S.; and Tian, Q. 2016. Mars: A video benchmark for large-scale person re-identification. In *ECCV*.

Zhong, Z.; Zheng, L.; Luo, Z.; Li, S.; and Yang, Y. 2019. Invariance Matters: Exemplar Memory for Domain Adaptive Person Re-identification. In *CVPR*.

Zhou, B.; Zhao, H.; Puig, X.; Xiao, T.; Fidler, S.; Barriuso, A.; and Torralba, A. 2019. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127: 302–321.