Cross-domain Open-world Discovery

Shuo Wen [1] **Maria Brbić** [1]

Abstract

In many real-world applications, test data may commonly exhibit categorical shifts, characterized by the emergence of novel classes, as well as distribution shifts arising from feature distributions different from the ones the model was trained on. However, existing methods either discover novel classes in the open-world setting or assume domain shifts without the ability to discover novel classes. In this work, we consider a *cross-domain open-world discovery* setting, where the goal is to assign samples to seen classes and discover unseen classes under a domain shift. To address this challenging problem, we present CROW, a prototype-based approach that introduces a cluster-then-match strategy enabled by a well-structured representation space of foundation models. In this way, CROW discovers novel classes by robustly matching clusters with previously seen classes, followed by fine-tuning the representation space using an objective designed for cross-domain open-world discovery. Extensive experimental results on image classification benchmark datasets demonstrate that CROW outperforms alternative baselines, achieving an 8% average performance improvement across 75 experimental settings.

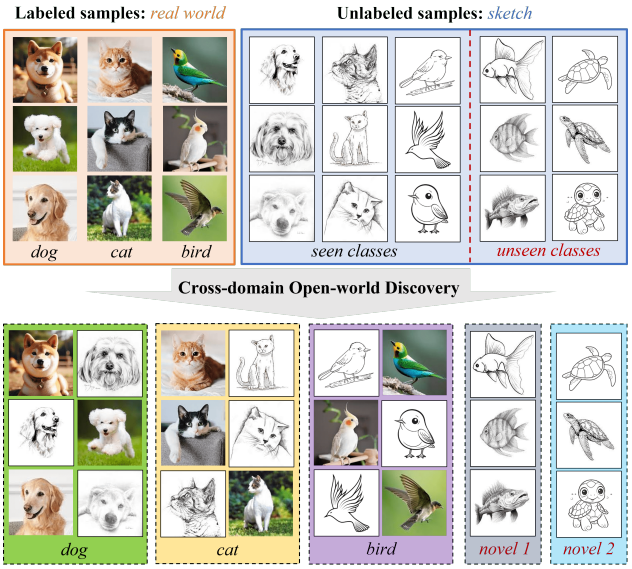

Figure 1. **Illustration of the cross-domain open-world discovery setting.** In the cross-domain open-world discovery setting, the goal is to assign samples to previously seen classes and discover new classes under a domain shift. In the example, novel classes like *'fish'* and *'turtle'*, exist in unlabeled data. Additionally, the labeled samples are from the real-world domain, while the unlabeled samples are sketches. In this setting, the goal is to assign each unlabeled sample to either a seen category (*'dog'*, *'cat'*, *'bird'*) or to a novel category that is discovered (*'novel 1'*, *'novel 2'*).

1. Introduction

The rise of deep learning has brought significant advancements, empowering machine learning systems with exceptional performance in tasks requiring extensive labeled data (LeCun et al., 2015; Schmidhuber, 2015; Silver et al., 2016). However, many models are developed within a closed-world paradigm, assuming that training and test data originate from a predetermined set of classes within the same domain. This assumption is overly restrictive in many real-world scenarios. For example, a model trained to categorize diseases in medical images from one hospital may experience domain shifts when applied to images from different hospitals. Moreover, during model deployment, novel and rare diseases may emerge that the model has never seen during training. In the open-world scenario, the model should have the capability to generalize beyond predefined classes and domains, a departure from the closed-world scenario often presumed in traditional approaches.

Open-world learning (Bendale & Boult, 2015) extends beyond closed-world paradigms by enabling models to recognize unseen classes and scenarios, addressing the dynamic challenges of real-world environments. In this context, open-world semi-supervised learning (OW-SSL) (Cao et al., 2022) defines a setting in which the objective is to annotate seen classes and discover unseen classes. However, OW-SSL

[1]EPFL, Switzerland. Correspondence to: Maria Brbić <mbrbic@epfl.ch>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

assumes that the labeled and unlabeled data belong to the same domain, which is often not the case. On the other hand, Universal Domain Adaptation (UniDA) (You et al., 2019) tackles the problem of domain and categorical shifts between labeled and unlabeled data. However, the primary objective of UniDA is to assign samples to seen classes and reject unseen samples as outliers, rather than discover novel unseen classes.

In this work, we address this gap by considering a Cross-Domain Open-World Discovery (CD-OWD) setting. In this setting, the objective is to assign samples to pre-existing (seen) classes while simultaneously being able to discover new (unseen) classes under a domain shift (Figure 1). This setting operates within a transductive learning framework, where we have access to both a labeled dataset (source set) and an unlabeled dataset (target set) during training. In contrast to OW-SSL, this setting considers not only categorical shifts but also domain shifts. In contrast to UniDA, the goal here is to discover novel classes instead of rejecting all of them as unknowns. Thus, CD-OWD needs to overcome the challenges of both open-world semi-supervised learning and universal domain adaptation. This setting has been previously considered in (Yu et al., 2022). However, their evaluation approach is not suitable for the proposed setting hindering the ability to effectively solve the proposed task.

A straightforward approach to tackle this challenge is to first apply one of the UniDA methods (Saito et al., 2020; Saito & Saenko, 2021; Chang et al., 2022; Qu et al., 2023) to annotate the seen samples and identify the unseen samples. After that, the detected unseen samples can be clustered to discover novel classes. We call this approach *match-then-cluster*. In practice, this approach encounters two problems. First, UniDA methods rely on a sensitive threshold to separate seen and unseen samples. Finding the optimal threshold using validation sets is not feasible because the domain gap between labeled and unlabeled samples prevents the creation of validation sets that accurately reflect the target domain. Second, when UniDA methods fail to perfectly separate seen and unseen samples, the seen samples misclassified as unseen introduce noise to the unseen samples, thereby reducing the quality of the clustering process.

To overcome these challenges, we propose CROW (**C**ross-domain **R**obust **O**pen-**W**orld-discovery), a method that employs a *cluster-then-match* approach, leveraging the capabilities of foundation models. The key idea in CROW is to utilize the well-structured latent space of foundation models (Radford et al., 2021; Oquab et al., 2023; Singh et al., 2022) to first cluster the data and then use a robust prototype-based matching strategy. This matching strategy enables CROW to associate multiple target prototypes with seen classes, thereby alleviating the issues of over-clustering and under-clustering. After matching prototypes, CROW combines cross-entropy loss applied to source samples with entropy maximization loss applied to target samples to further improve the representation space.

We evaluate CROW across 75 different categorical-shift and domain-shift scenarios created from four benchmark domain adaptation datasets for image classification. The results demonstrate that our approach outperforms open-world semi-supervised learning and universal domain adaptation baselines by a large margin. Specifically, CROW outperforms the strongest baseline GLC by an average of 8% on the H-score. Moreover, CROW is robust to different hyperparameters, an unknown number of target classes, and different seen/unseen splits.

2. Related work

The cross-domain open-world discovery setting is closely related to open-world semi-supervised learning and universal domain adaptation. It is a harder setting compared to these two settings as it requires overcoming the challenges of both settings — we need to discover novel classes under a domain shift. CROW builds upon the power of foundation models, allowing us to adopt the *cluster-then-match* strategy proposed in this work.

Open-world learning. Open-world learning (Bendale & Boult, 2015; 2016; Boult et al., 2019) entails annotating unlabeled data in the face of categorical shift, where new classes may arise in the unlabeled data. Open Set Label Shift (OSLS) (Garg et al., 2022) is a setting that detects the samples from the seen classes and annotates them. However, it focuses on seen classes and does not separate different unseen classes. Novel Class Discovery (NCD) (Hsu et al., 2018) aims to discover unseen classes. However, NCD assumes that all the unlabeled samples are from novel classes, so it does not need to detect common classes. Open-world semi-supervised learning (OW-SSL) (Cao et al., 2022) combines the settings of OSLS and NCD. It aims to annotate seen classes and discover unseen classes under the assumption that the unlabeled samples are from both seen and novel classes. However, OW-SSL assumes that labeled and unlabeled data belong to the same domain, which is not always true. In this work, we consider the *cross-domain open-world discovery* setting which accounts for domain shift.

Unsupervised domain adaptation. Unsupervised domain adaptation (UDA) (Ganin & Lempitsky, 2015) aims to annotate unlabeled data under domain shift between labeled and unlabeled data. However, it assumes that labeled and unlabeled data originate from the same classes. Open-Set Domain Adaptation (OSDA) (Panareda Busto & Gall, 2017) and Universal Domain Adaptation (UniDA) (You et al., 2019) extend the setting of UDA by considering unseen classes in the unlabeled data. They aim to annotate seen

classes and detect unseen samples. Prior works (Saito et al., 2018; Ma et al., 2021; Saito & Saenko, 2021; Zhu et al., 2023; Zang et al., 2023) achieved significant success within both the OSDA and UniDA setting. However, these works reject unseen samples without exploring the internal structure of the unseen part. Recent works (Saito et al., 2020; Li et al., 2021; Jing et al., 2021; Chang et al., 2022; Lai & Zhou, 2024) have started paying attention to the internal structure of the target domain, especially for the unknown samples. However, most of them explore internal structures to better detect unseen samples but not to separate them according to new classes. UniDA methods are applicable to the cross-domain discovery setting by clustering the samples detected as novel. Yu et al. (2022) consider the cross-domain discovery setting. However, their evaluation strategy is not suitable since it does not directly evaluate the ability to discover novel classes. As a result, the evaluation does not accurately reflect performance in the context of the cross-domain discovery setting, leaving its effectiveness unclear.

We compare different problem settings in Table 1.

Table 1. Comparison of different problem settings. UDA stands for Universal Domain Adaptation; OSLS for Open Set Label Shift; NCD for Novel Class Discovery; UniDA for Universal Domain Adaptation; OW-SSL for Open-World Semi-Supervised Learning; and CD-OWD for Cross-Domain Open-World Discovery.

SETTING	DOMAIN SHIFT	SEEN DETECTION	NOVEL DISCOVERY
UDA	√	-	-
OSLS	-	√	-
NCD	-	-	√
UNIDA	√	√	-
OW-SSL	-	√	√
CD-OWD	√	√	√

Transfer learning and foundation models. Existing open-world and domain adaptation methods generally use the standard pretraining and fine-tuning paradigm of transfer learning (Torrey & Shavlik, 2010; Weiss et al., 2016; Kolesnikov et al., 2020). The pretrained feature extractors provide a well-structured latent space, allowing faster training and better generalization. Previous works on OW-SSL and UniDA (Saito et al., 2020; Saito & Saenko, 2021; Chang et al., 2022; Qu et al., 2023; Cao et al., 2022) directly use the ImageNet (Deng et al., 2009) supervised pretrained or SimCLR (Chen et al., 2020) self-supervised pretrained ResNet50 (He et al., 2016) as their backbone. However, recently developed foundation models (Radford et al., 2021; Singh et al., 2022; Oquab et al., 2023) provide a better structured initial latent space, eliminating the necessity for self-supervised training on target data to achieve reliable initialization. Previous works (Deng & Jia, 2023; Yu et al., 2023; Bommasani et al., 2021) show that foundation models help alleviate domain shifts in their representation space. In this work, we build our method upon the power of the well-structured representation space of foundation models. This allows us to adopt a *cluster-then-match* strategy in contrast to the *match-then-cluster* strategy extended from existing universal domain adaptation methods.

Cluster-then-match approach. To solve the universal domain adaptation problem, DCC (Li et al., 2021) first clusters the unlabeled target samples and then matches each target cluster to one seen class for recognizing target seen classes. This approach corresponds to the strategy we call *cluster-then-match* in this work. However, a limitation of this specific instantiation of the *cluster-then-match* strategy is that DCC requires one-to-one matching between seen classes and target clusters. This requirement cannot be satisfied in the condition of under-clustering (*i.e.*, assigning multiple seen classes to the same cluster) and over-clustering (*i.e.*, splitting a single seen class into multiple clusters), as the relationship between seen classes and target clusters is no longer one-to-one. Instead, we adopt robust matching, which mitigates this problem by releasing the constraint of one-to-one matching, allowing multiple seen classes to be matched to the same cluster and a single seen class to be matched to multiple clusters.

3. Method

3.1. Cross-domain open-world discovery setting

In the cross-domain open-world discovery, we assume a transductive learning setting, where a labeled dataset (*i.e.*, source set) $D_s = \{(x_i, y_i)\}_{i=1}^n$ and an unlabeled dataset (*i.e.*, target set) $D_t = \{x_i\}_{i=1}^m$ are given during training. We denote the set of classes in the source set as C_s and the set of classes in the target set as C_t. We consider both the categorical shift and the domain shift. Under the **categorical shift**, we assume $C_s \cap C_t \neq \emptyset$ and $C_s \neq C_t$. We consider C_s as a set of seen classes and $C_t \setminus C_s$ as a set of novel classes. Additionally, under the **domain shift**, we consider $P(x)$ as the feature distribution of data x. We assume that $P(x^s) \neq P(x^t)$, where $x^s \in D_s$ and $x^t \in D_t$.

The objective is to assign each $x_i \in D_t$ a label y_i. The y_i is either from a seen class in C_s or from a novel class that is discovered.

3.2. Overview of CROW

To overcome the challenges of cross-domain open-world discovery, the novel classes need to be well separated in the representation space. Early incorporation of labels from seen classes in the training process can lead to a bias towards seen classes, hindering the ability to differentiate between samples of novel classes. The key idea in CROW is to adopt the *cluster-then-match* strategy, enabled by the well-structured representation space of foundation models. In particular, CROW first clusters the target samples in the

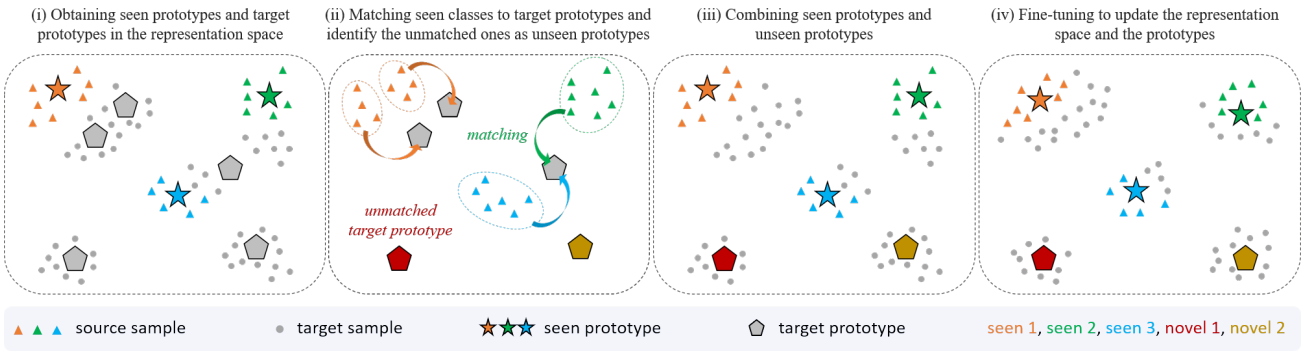

Figure 2. **Conceptual overview of CROW.** *(i)* CROW extracts features from a foundation model for both source and target samples. Seen prototypes are then obtained using labeled source samples, while target prototypes are obtained by clustering target samples. *(ii)* CROW matches seen classes to target prototypes using the source samples. Unmatched target prototypes are identified as unseen prototypes. *(iii)* CROW combines seen prototypes and unseen prototypes. *(iv)* Finally, CROW fine-tunes the foundation model to update the representation space and the prototypes.

representation space of a foundation model, followed by a robust matching that associates seen classes with the target clusters. Finally, CROW is fine-tuned using an objective specially tailored for the open-world discovery setting.

Thus, CROW adopts a three-step procedure approach, including: *(i)* clustering, *(ii)* matching, and *(iii)* fine-tuning.

3.3. Clustering step

To obtain clusters of target samples, CROW leverages a robust representation space of a foundation model (Radford et al., 2021; Singh et al., 2022; Oquab et al., 2023; Gadetsky & Brbic, 2023). The goal of the clustering step in CROW is to obtain the prototypes of target sample clusters (referred to as target prototypes) and the prototypes of the seen classes (referred to as seen prototypes).

The foundation model is used as a feature extractor f_θ. Let $\mathcal{X}$ be the input space; the feature extractor $f_\theta : \mathcal{X} \to \mathbb{R}^d$ maps the input space $\mathcal{X}$ to a d-dimensional representation space. Specifically, given input $x \in \mathcal{X}$, f_θ extracts the feature $z \in \mathbb{R}^d$ by $z = f_\theta(x)$. Note that we add an L_2-normalized layer at the end, so the feature z is L_2-normalized.

To get the seen prototypes $W_{seen} = [p^s_1, p^s_2, ..., p^s_{|C_s|}]$, where $[\cdot]$ denotes concatenation, we consider W_{seen} as a L_2-normalized linear classifier and train it on top of the representation space of a foundation model f_θ. In particular, we optimize cross-entropy loss on source samples to obtain seen prototypes W_{seen}. Specifically, for each source sample x^s, we first extract the feature z^s by $z^s = f_\theta(x^s)$. Then, we obtain the predictions using $p(y|x^s) = \sigma(W^T_{seen} \cdot z^s)$, where σ is the softmax activation function. Finally, we optimize W_{seen} by applying cross-entropy loss on $p(y|x^s)$. Note that we freeze the feature extractor f_θ during this process.

To obtain the target prototypes $W_t = [p^t_1, p^t_2, ..., p^t_{|C_t|}]$, we first extract the features of all target samples using $z^t = f_\theta(x^t)$. Then, we apply a K-means clustering with $k = |C_t|$ to get the target prototypes $W_t = [p^t_1, p^t_2, ..., p^t_{|C_t|}]$. Here, we assume the number of target classes $|C_t|$ is given as a prior.

The clustering step of CROW, which results in seen prototypes W_{seen} and target prototypes W_t is illustrated in Figure 2 *(i)*. After obtaining the seen prototypes W_{seen}, the goal is to identify the unseen prototypes W_{unseen} from target prototypes W_t in the matching step.

3.4. Matching step

In the matching step of CROW, the goal is to identify target prototypes that belong to unseen classes. This is achieved by matching seen classes to target prototypes and designating unmatched target prototypes as the prototypes of unseen classes. To accomplish this, CROW employs on a robust matching procedure that allows multiple seen classes to match a target prototype and multiple target prototypes to match a seen class.

To match seen classes to target prototypes, we first compute a co-occurrence matrix $\Gamma \in \mathbb{R}^{|C_t| \times |C_s|}$ between target prototypes and seen classes. This co-occurrence matrix represents the number of source samples from a given seen class assigned to a target prototype. We assign a source sample to a target prototype if that prototype is its nearest prototype in the representation space. In essence, the co-occurrence matrix Γ quantifies the proximity of seen classes to the target prototypes.

After computing the co-occurrence matrix Γ, we apply a column-wise softmax to Γ and obtain the distribution matrix D as follows:

$$D_{i,j} = \frac{e^{\Gamma_{i,j}}}{\sum_{k=1}^{|C_t|} e^{\Gamma_{k,j}}}. \quad (1)$$

Each column of D represents the distribution of the source samples of a seen class to the target prototypes. Finally, we obtain the matching matrix M by applying a threshold τ to D:

$$M_{i,j} = \begin{cases} 1 & D_{i,j} \geq \tau \\ 0 & D_{i,j} < \tau \end{cases} . \quad (2)$$

Here, $M_{i,j} = 1$ means that the seen class C_j is matched to target prototype p_i^t. After matching seen classes to target prototypes, we can easily identify the target prototypes that have not been matched to any seen class as the prototypes of unseen classes. This step is illustrated in Figure 2 *(ii)*.

The matching step gives us the unseen prototypes W_{unseen}, and we have already obtained the seen prototypes W_{seen} from the clustering step. We then combine them to initialize a linear classifier $W = [W_{seen}, W_{unseen}]$ on top of the feature extractor f_θ. This step is illustrated in Figure 2 *(iii)*.

Note that the number of identified unseen prototypes may not necessarily be equal to $(|C_t| - |C_s|)$. This disparity arises because the number of matched target prototypes can differ from the number of seen classes $|C_s|$ due to under-clustering and over-clustering issues.

We illustrate the matching procedure in Figure 3. In the example, samples in seen class C_1 are over-clustered into two clusters represented by the target prototypes p_1 and p_2, while samples in seen classes C_2 and C_3 are under-clustered and represented by only one target prototype p_4. After computing the matching matrix M, we match seen class C_1 to both target prototypes p_1 and p_2 and match seen classes C_2 and C_3 to target prototype p_4. Based on the result of matching, we consider p_3 and p_5, the unmatched target prototypes, to be the unseen prototypes.

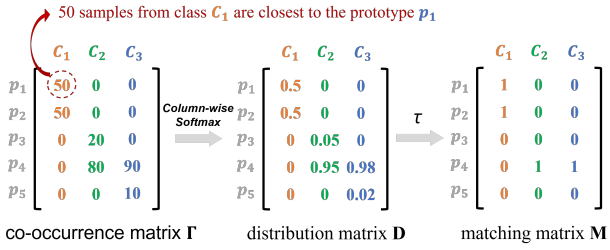

Figure 3. **The process of matching.** We first obtain the co-occurrence matrix Γ between target prototypes and seen classes. Then, we apply a column-wise softmax to the co-occurrence matrix Γ to get the distribution matrix D. Finally, we apply a threshold τ to each $D_{i,j}$ to obtain the matching matrix M. $M_{i,j} = 1$ means the class C_j is matched to the prototype p_i.

Threshold parameter τ. τ is the threshold applied to each element in the distribution matrix D (Equation 2) to determine if there is a match between a target cluster and a seen class. We chose the threshold τ by observing the distribution matrix D. From our observations across all experiment settings, most elements in the distribution matrix D are smaller than 0.02 (unmatched) or larger than 0.98 (matched), and a few elements are around 0.5 (matched, but over-clustering occurs). Based on this observation, we choose $\tau = 0.3$ across all the experiments. Note that the distribution matrix D is obtained during training, so we do not use the label of the target samples (test set) to set this threshold.

3.5. Fine-tuning step

After the *cluster-then-match* procedure, we initialize a linear classifier W using the seen and unseen prototypes on top of the feature extractor f_θ. In the final step of CROW, we fine-tune both the feature extractor f_θ and the classifier W to further improve model performance by updating the representation space and the prototypes. This step is illustrated in Figure 2 *(iv)*.

In particular, a cross-entropy loss L_s (Krizhevsky et al., 2012) is applied to the source samples $x^s \in X^s$. This loss is used to transfer the knowledge of seen classes from the source samples to the target samples, and it is used to update the feature extractor and the seen prototypes:

$$L_s(f_\theta, W_{seen}) = \frac{1}{N_s} \sum_{x^s \in X^s} -y(x^s) \log(p(y|x^s)), \quad (3)$$

where N_s is the number of source samples, $y(x)$ is the one-hot ground truth label of x, and $p(y|x) = \sigma(W^T \cdot f_\theta(x))$.

In addition, to balance the predictions of seen and unseen classes, we apply the regularization loss L_{reg} (Van Gansbeke et al., 2020; Cao et al., 2022) to the target samples $x^t \in X^t$. This term maximizes the entropy of the average of all the predictions:

$$L_{reg}(f_\theta, W) = \frac{1}{N_t} \sum_{x^t \in X^t} p(y|x^t) \log(\frac{1}{N_t} \sum_{x^t \in X^t} p(y|x^t)), \quad (4)$$

where N_t is the number of target samples.

The final objective function in the fine-tuning step of the CROW is as follows:

$$\min_{\theta, W} L_s(f_\theta, W_{seen}) + \lambda L_{reg}(f_\theta, W), \quad (5)$$

where λ denotes a regularization hyperparameter.

4. Experiments

4.1. Experimental setup

Datasets. Universal domain adaptation (UniDA) shares the same assumption on data as cross-domain open-world

Table 2. Average H-score (%) comparison of different seen/unseen splits on dataset Office, OfficeHome, VisDA, and DomainNet. We color the best and second-best results in red and blue.

	OFFICE			OFFICEHOME			VISDA			DOMAINNET			AVERAGE
SEEN/UNSEEN	21/10	16/15	10/21	45/20	33/32	20/45	8/4	6/6	4/8	240/105	173/172	105/240	
SIMPLE	64.9	66.8	78.3	62.3	66.0	65.4	55.4	50.8	50.9	53.2	55.9	57.8	60.6
GCD	62.6	58.5	58.0	48.7	47.5	48.1	31.2	32.5	25.2	35.0	41.3	41.3	44.2
ORCA	57.9	63.4	62.3	48.9	48.9	49.9	31.3	33.6	33.9	28.9	31.5	33.7	43.7
DCC	72.8	74.9	75.2	63.6	65.0	64.7	60.7	57.3	56.7	45.5	47.5	47.7	61.0
DANCE	75.4	68.9	69.9	65.7	65.6	67.1	57.2	51.5	48.4	56.3	55.7	58.8	61.7
OVANET	73.2	75.7	75.3	66.4	68.6	68.7	59.9	60.8	60.2	54.2	55.6	58.5	64.7
UNIOT	76.1	79.6	83.4	64.4	64.9	64.8	62.0	62.4	59.8	45.7	51.2	50.8	63.9
NCDDA	80.3	81.2	81.7	63.2	64.3	65.0	57.2	60.7	59.3	50.1	52.6	55.7	64.3
SAN	80.5	80.2	82.0	64.3	65.0	67.2	61.2	63.5	61.5	53.2	54.8	55.0	65.7
GLC	75.7	74.6	77.3	65.2	68.2	69.3	61.2	65.2	62.7	54.9	56.1	55.7	65.8
CROW	84.7	84.9	85.6	69.4	69.6	70.2	70.5	69.2	71.1	57.8	59.0	61.5	71.1

Table 3. Average seen accuracy (%), unseen accuracy (%), and H-score (%) of 50% seen/unseen splits on dataset Office, OfficeHome, VisDA, and DomainNet. We color the best and second-best results in red and blue.

	OFFICE (16/15)			OFFICEHOME (33/32)			VISDA (6/6)			DOMAINNET (173/172)		
	SEEN	UNSEEN	H-SCORE	SEEN	UNSEEN	H-SCORE	SEEN	UNSEEN	H-SCORE	SEEN	UNSEEN	H-SCORE
SIMPLE	62.3	72.0	66.8	69.2	63.1	66.0	68.1	40.5	50.8	69.1	47.0	55.9
GCD	54.1	63.8	58.5	46.6	48.5	47.5	43.8	25.8	32.5	42.3	40.3	41.3
ORCA	69.6	58.2	63.4	67.1	38.4	48.9	68.3	22.3	33.6	62.3	21.1	31.5
DCC	78.0	72.1	74.9	70.3	60.5	65.0	75.3	46.2	57.3	50.2	45.1	47.5
DANCE	73.3	65.1	68.9	72.0	60.3	65.6	70.2	40.7	51.5	69.4	46.5	55.7
OVANET	76.7	74.8	75.7	71.8	65.6	68.6	60.4	61.2	60.8	65.1	48.5	55.6
UNIOT	81.7	77.6	79.6	70.5	60.2	64.9	75.7	49.4	59.8	59.2	45.1	51.2
NCDDA	88.9	74.7	81.2	71.2	58.6	64.3	70.4	53.3	60.7	68.9	42.5	52.6
SAN	89.4	72.7	80.2	72.0	59.2	65.0	74.5	55.3	63.5	67.3	46.2	54.8
GLC	87.8	64.8	74.6	73.3	63.8	68.2	73.4	58.7	65.2	62.9	50.6	56.1
CROW	90.0	80.3	84.9	71.9	67.4	69.6	77.0	62.8	69.2	70.3	50.9	59.0

discovery. Thus, we evaluate our method and the baselines on the standard UniDA benchmark datasets. The **Office** (Saenko et al., 2010) dataset has 31 classes and three domains: Amazon (A), DSLR (D), and Webcam (W). There are around 3K images in domain A and 1K in domains D and W. The **OfficeHome** (Venkateswara et al., 2017) dataset comprises 65 classes and four domains: Art (A), Clipart (C), Product (P), and Real-World (R). There are around 4K images in domains C, P, and R, and 2K images in domain A. **VisDA** (Peng et al., 2017) is a synthetic-to-real (S2R) dataset with 12 classes. There are around 150K images in domain S and 50K in domain R. **DomainNet** (Peng et al., 2019) is the largest dataset, including 345 classes and six domains. Following the previous works (Fu et al., 2020; Saito & Saenko, 2021; Chang et al., 2022), we use three domains: Painting (P), Real (R), and Sketch (S).

For each experimental setting, we create a pair of domains from one dataset, designating one domain as the source and another one as the target. Samples from the source domain have labels, while those from the target domain remain unlabeled. Following the previous UniDA works (Saito et al., 2020; Saito & Saenko, 2021; Chang et al., 2022), we sort all the classes alphabetically and define the last n class as unseen classes. Then, we remove samples of the predefined unseen classes from the source set. We evaluate CROW and the baselines with different ratios of seen/unseen classes, including 70%, 50%, and 30%.

Evaluation metric. Open-world semi-supervised learning (OW-SSL) and cross-domain open-world discovery settings share the same task of recognizing seen and discovering unseen classes. Therefore, in line with the evaluation metric of the OW-SSL setting, we test the accuracy of both seen and unseen classes, referred to as seen and unseen accuracy. To compute unseen accuracy, we use the Hungarian algorithm (Kuhn, 1955) to match the unseen classes and subsequently calculate the accuracy.

To evaluate the overall performance, we calculate the H-score (Fu et al., 2020), as it provides a balanced measure of the performance of seen and unseen classes:

$$H\_score = \frac{2 \cdot acc_{seen} \cdot acc_{unseen}}{acc_{seen} + acc_{unseen}}$$

Baselines. We compare CROW to UniDA and OW-SSL baselines as their settings are the closest to cross-domain open-world discovery. Since UniDA methods cannot discover novel classes, we extend them by first applying a UniDA method and then clustering the detected unseen samples to discover novel classes. OW-SSL methods do not need to be extended since they perform the same task even if under different assumptions about the data. We include as baselines two OW-SSL methods, namely ORCA (Cao et al., 2022) and GCD (Vaze et al., 2022). We additionally compare to the six UniDA methods, namely DCC (Li et al., 2021), DANCE (Saito et al., 2020), OVANet (Saito

& Saenko, 2021), UniOT (Chang et al., 2022), SAN (Zang et al., 2023) and GLC (Qu et al., 2023). Also, we compare to NCDDA (Yu et al., 2022), which considers the cross-domain open-world discovery setting.

In addition, we design a simple baseline using the *match-then-cluster* approach, referred to as SIMPLE. SIMPLE first trains the classifier on the source set with cross-entropy loss. Then, it predicts the labels and computes the prediction entropy for target samples. Samples with entropy exceeding a predefined threshold are considered unseen and undergo clustering. After labeling all the samples from seen and unseen classes, we finetune SIMPLE using the same objective function (5) proposed in CROW. More details are provided in Appendix A.2.

Implementation details. We use CLIP (Radford et al., 2021) ViT-L (Dosovitskiy et al., 2021) as the feature extractor for CROW and all the baselines. When fine-tuning, we update only the last two blocks in ViT-L and freeze the other parts following Deng & Jia (2023), which shows that fine-tuning the whole ViT-L hurts the performance of the foundation model. More details are provided in the Appendix A.1. Our code is publicly available[1].

UniDA methods are sensitive to the threshold used to separate seen and unseen samples. This threshold is crucial to the balance of seen and unseen accuracy. However, after changing the backbone from the the ImageNet pretrained ResNet50 to the CLIP ViT-L, the original threshold τ suggested in their works can lead to accuracy bias towards seen or unseen classes, resulting in a low H-score. To improve UniDA baselines and find optimal threshold τ that results in balanced results, we adapt the threshold *using the test set*. This leads to an unrealistic evaluation setting since, in reality, we cannot use the test set to decide on the threshold, but our goal is to push the limits of the baselines in this setting. With the CLIP ViT-L backbone, our results substantially exceed the performance of all the baselines compared to their respective papers. We show the threshold τ we use for the baselines in the Appendix A.3. In contrast, CROW uses the same $\tau = 0.3$ and $\lambda = 0.1$ across all the experiments, and as we later show, it is robust to this threshold.

4.2. Results

Evaluation on benchmark datasets. We report the average H-score across four benchmark datasets: Office31, OfficeHome, VisDA, and DomainNet. We compare CROW to baselines with different ratios of seen and unseen classes, including 70%, 50%, and 30% seen/unseen splits. Table 2 shows that CROW consistently outperforms all baselines in terms of H-score. In particular, across all datasets, CROW achieves an 8% relative improvement in the average H-score over the baselines. The detailed results of the 75 different experimental settings with different pairs of source/target datasets are shown in Appendix B.7.

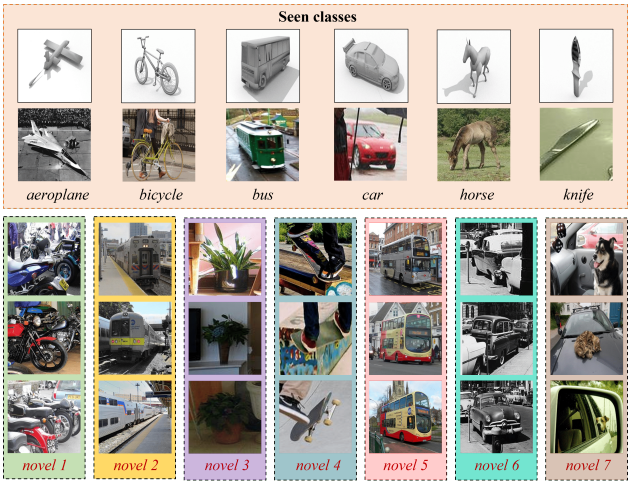

Figure 4. **Confident samples for seen and unseen classes** on VisDA. The synthetic images are from the source, and the real-world images are from the target.

We next compare the performance separately on seen and unseen classes using the 50% seen/unseen split. The results in Table 3 show that CROW consistently outperforms the baselines in discovering novel classes, achieving an 8.3% average improvement over the baselines. On seen classes, CROW outperforms baselines by a 2.9% average improvement across all datasets. We observe similar results with 70% and 30% seen/unseen splits (Appendix B.7).

In comparison to UniDA methods that adopt the match-then-cluster strategy (SIMPLE, DANCE, OVANet, UniOT, SAN, and GLC), CROW outperforms the best baseline by 5.3% in average H-score, highlighting the benefits of our cluster-then-match strategy. When compared to the DCC, which also adopts a cluster-then-match strategy but follows a one-to-one matching procedure, CROW outperforms DCC by 9.4% in H-score. This underscores the benefits of the robust matching procedure. Compared to OW-SSL methods (ORCA and GCD), we observe nearly 30% improvement in average H-scores, indicating that OW-SSL methods cannot effectively overcome domain shifts and be applied in this setting. Furthermore, CROW surpasses NCDDA in H-score by 6.8%, demonstrating its superior effectiveness in the cross-domain open-world discovery setting

In addition, we compare CROW to directly applying K-means to the CLIP features on the target datasets. We also compare our method to the CLIP zero-shot learning (Radford et al., 2021). The results show that our method outperforms these two methods by a large margin. We present the results and analysis in the Appendix B.1 and B.2.

[1] https://github.com/mlbio-epfl/crow

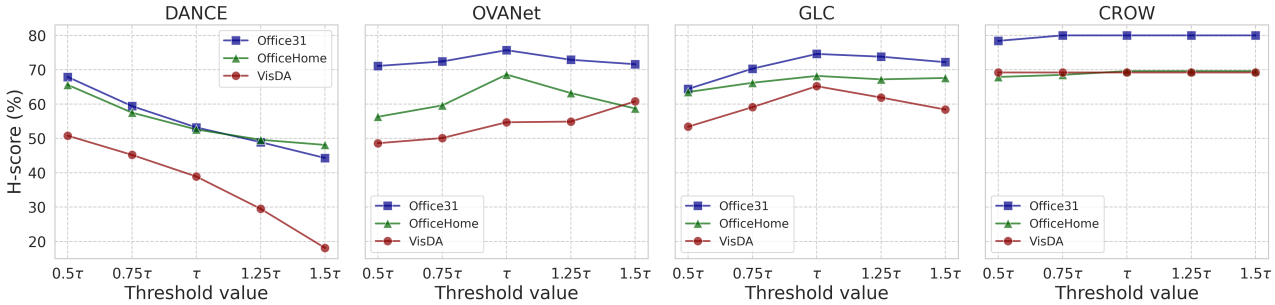

Figure 5. **Sensitivity to the threshold.** τ is the original threshold provided by our method and the previous works. We modify τ by scaling it with a multiplication factor.

4.3. Qualitative results

We visually inspect classes discovered by CROW on the VisDA dataset. Figure 4 shows the top-1 confident sample for each seen category and the top-3 confident samples for each novel category. The results reveal that, in addition to annotating seen classes, CROW successfully discovers seven unseen classes. Notably, CROW accurately discovers the VisDA predefined classes '*motorcycle*', '*plant*', '*skateboard*', and '*train*', which are absent in the source set. Additionally, the model recognizes double-decker buses, old-style cars, and cars with animals as novel classes. Despite discrepancies from the ground truth annotations (*e.g.*, cars with animals are originally labeled as cars), the classes discovered by CROW are meaningful. We further look at the VisDA predefined classes '*person*' and '*truck*' that are not discovered by CROW. We find that confident predictions are people with skateboards that are classified into a novel class that corresponds to '*skateboard*', again showing that the CROW's predictions are indeed meaningful. Furthermore, ground-truth class '*truck*' is typically assigned to '*car*', which is reasonable given many shared features. We elaborate more and show examples of failure cases in Appendix B.3. Overall, this opens interesting research directions for designing proper evaluation strategies in this challenging setting since disagreement with the ground-truth annotations may not necessarily mean that the results are wrong, and even human annotators could disagree in these failure cases.

4.4. Ablation studies

Benefits of fine-tuning. We evaluate how much fine-tuning helps to improve the performance of CROW on the VisDA datataset. We compare CROW in three settings: *(i)* without fine-tuning (*i.e.*, only clustering and matching steps), *(ii)* with fine-tuning only the linear classifier W, and *(iii)* with fine-tuning also the feature extractor of a foundation model. The results in Table 4 show that fine-tuning both the feature extractor f_θ and the classifier W helps to improve the performance. However, CROW can still achieve high performance even without fine-tuning, by adopting only clustering and matching steps.

Table 4. Seen, unseen accuracy (%) and H-score (%) of our method with different fine-tuning strategies on the VisDA dataset (6/6).

	SEEN	UNSEEN	H-SCORE
WITHOUT FINE-TUNE	73.8	61.2	66.9
FINE-TUNE ONLY W	75.2	61.8	67.8
FINE-TUNE f_θ AND W	77.0	62.8	69.2

Sensitivity to threshold τ. We next evaluate the sensitivity to the thresholds of CROW and UniDA baselines on the 50% seen/unseen split of the Office31, OfficeHome, and VisDA datasets. We compare CROW, DANCE, OVANet, and GLC across different values of their respective threshold. CROW has a matching threshold τ (Equation 2). DANCE has a threshold that detects unseen samples using the entropy of the prediction. OVANet and GLC have a threshold that detects unseen samples using the prediction confidence. Due to the different scales of the thresholds employed by each method, we test values from 0.5τ to 1.5τ, where τ denotes the default threshold used in these previous works. We evaluate the effect of changing the threshold on performance in Figure 5. The results show that CROW is extremely robust to the threshold variations. However, this is not the case for baseline methods. For example, DANCE demonstrates considerable sensitivity to threshold changes, and the original τ deviates from the optimal value after changing the backbone. For OVANet and GLC, their original τ yields good performance in the majority of cases, but these methods still exhibit sensitivity to the threshold.

Ablation study on the objective function. To investigate the importance of each part of the objective function in Equation 5, we conduct an ablation study on the VisDA dataset. Table 5 shows that the removal of the supervised loss L_s results in a decrease in seen accuracy, while the absence of the entropy regularization L_{reg} causes the accuracy to bias toward seen classes. The best performance is achieved when combining the two losses.

CROW with different foundation models. In all experiments, we used CLIP as the feature extractor. We next

Table 5. Ablation study on the objective function. We show the seen/unseen accuracy, and H-score (%) on the VisDA dataset (6/6).

APPROACH	SEEN	UNSEEN	H-SCORE
W/O L_s	65.6	61.5	63.5
W/O L_{reg}	77.2	56.9	65.7
CROW	77.0	62.8	69.2

compare the performance of CROW in the space of different foundation models. As foundation models, we use CLIP (Radford et al., 2021), DINO_v2 (Oquab et al., 2023), and SWAG (Singh et al., 2022) across varying sizes of the ViT. Table 6 illustrates that CROW achieves better performance with stronger feature extractors. This suggests that CROW can benefit from further advancement in the field by using stronger foundation models as a feature extractor.

Table 6. Seen, unseen accuracy (%) and H-score (%) of CROW with different pretrained foundation models on VisDA (6/6).

METHOD	BACKBONE	SEEN	UNSEEN	H-SCORE
CLIP	VIT-B	74.5	58.9	65.8
	VIT-L	77.0	62.8	69.2
DINO_V2	VIT-B	74.2	55.5	63.5
	VIT-L	76.8	57.6	65.8
	VIT-G	78.2	60.4	68.2
SWAG	VIT-B	74.8	60.2	66.7
	VIT-L	78.4	63.0	69.9
	VIT-H	79.0	63.4	70.3

Results of CROW with the estimated number of novel classes $|C_t|$ and on the UniDA data split are shown in Appendix B.5 and B.6.

4.5. Pretrained model for unseen classes

A potential problem is that the pretrained feature extractors might have encountered the unseen classes during pretraining. Indeed, this is a common issue in the research fields of novelty detection and category discovery, and it existed even before the age of foundation models. For example, most previous works on open-set/universal domain adaptation, novel class discovery, and general class discovery (Saito & Saenko, 2021; Saito et al., 2020; Li et al., 2021; Vaze et al., 2022) use ImageNet pretrained ResNet-50 as their backbone, and some of the unseen classes are present in the ImageNet dataset.

To test whether a model that has seen instances of unseen classes can artificially inflate the results, we perform an experiment on the Office31 dataset in which we train two versions of the ResNet-50 feature extractor in a supervised fashion: (1) trained on the whole ImageNet dataset, and (2) trained on the ImageNet dataset without the samples of the unseen classes (*e.g.*, we remove samples of '*desk*', '*barber chair*', '*folding chair*', '*rocking chair*' from the ImageNet dataset for the unseen class '*desk_chair*'). Table 7 shows that whether the model has seen instances of unseen classes only marginally affects performance. Furthermore, it is important to emphasize that the model trained without instances of unseen classes has also seen fewer different samples and less data in general, so we cannot fully attribute these small differences to the fact that the model has seen unseen samples. However, how to avoid this common problem in the research fields of novelty detection and category discovery still needs to be further explored (Rambhatla et al., 2021).

Table 7. Ablation study on different pretrained datasets. We show the seen, unseen accuracy (%) on the Office dataset.

DATASET	SEEN	UNSEEN
WHOLE IMAGENET	79.5	56.4
IMAGENET W.O. UNSEEN	79.6	55.8

5. Limitations

In CROW, the ability to discover novel classes heavily relies on clustering within the representation space established by the foundation models. Consequently, CROW may exhibit worse performance on datasets where the foundation models lack robustness. For example, we evaluated our method on DomainNet, from Sketch (source) to Quickdraw (target). Quickdraw contains images with grey-level lineart, and CLIP is not robust to that image style. Under the setting of the 50% seen/unseen split and the exact same training setup as described in the Implementation details in Section 4.1, CROW achieves only 20.4% seen accuracy and 24.2% unseen accuracy on this dataset. However, the strongest baseline GLC achieves even worse results with 20.1% seen accuracy and 18.6% unseen accuracy. These results indicate that further exploration needs to be done to deal with challenging datasets like DomainNet Quickdraw.

6. Conclusion

In this work, we address the gap between open-world semi-supervised learning and universal domain adaptation by considering a cross-domain open-world discovery setting that encompasses both categorical and distributional shifts. To tackle this challenging problem, we propose CROW, a prototype-based method built upon foundation models. CROW combines source seen prototypes and target unseen prototypes through a robust *cluster-then-match* approach, simultaneously accomplishing seen class recognition and unseen class discovery. By conducting experiments across 75 different categorical-shift and domain-shift situations, we demonstrate that CROW consistently outperforms alternative baselines and effectively overcomes the challenges of the cross-domain open-world discovery setting.

Acknowledgements

The authors thank Artyom Gadetsky, Liangze Jiang, Matej Grcić, and Ramon Vinas Torné, for their feedback on our manuscript. We also thank Chanakya Ekbote and Yulun Jiang for discussing this work. We gratefully acknowledge the support of EPFL and ZEISS.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bendale, A. and Boult, T. Towards open world recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Bendale, A. and Boult, T. E. Towards open set deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Boult, T. E., Cruz, S., Dhamija, A. R., Gunther, M., Henrydoss, J., and Scheirer, W. J. Learning and the unknown: Surveying steps toward open world recognition. In *AAAI Conference on Artificial Intelligence*, 2019.

Cao, K., Brbic, M., and Leskovec, J. Open-world semi-supervised learning. In *International Conference on Learning Representations*, 2022.

Cao, Z., You, K., Long, M., Wang, J., and Yang, Q. Learning to transfer examples for partial domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Chang, W., Shi, Y., Tuan, H., and Wang, J. Unified optimal transport framework for universal domain adaptation. *Advances in Neural Information Processing Systems*, 2022.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

Deng, B. and Jia, K. Universal domain adaptation from foundation models. *arXiv preprint arXiv:2305.11092*, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021.

Fu, B., Cao, Z., Long, M., and Wang, J. Learning to detect open classes for universal domain adaptation. In *European Conference on Computer Vision*, 2020.

Gadetsky, A. and Brbic, M. The pursuit of human labeling: A new perspective on unsupervised learning. *Advances in Neural Information Processing Systems*, 2023.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 2015.

Garg, S., Balakrishnan, S., and Lipton, Z. Domain adaptation under open set label shift. *Advances in Neural Information Processing Systems*, 2022.

Han, K., Vedaldi, A., and Zisserman, A. Learning to discover novel visual categories via deep transfer clustering. In *IEEE International Conference on Computer vision*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Hsu, Y.-C., Lv, Z., and Kira, Z. Learning to cluster in order to transfer across domains and tasks. In *International Conference on Learning Representations*, 2018.

Jing, T., Liu, H., and Ding, Z. Towards novel target discovery through open-set domain adaptation. In *IEEE International Conference on Computer vision*, October 2021.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In *European Conference on Computer Vision*, 2020.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2012.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 1955.

Lai, Liu, Z. and Zhou. Memory-assisted sub-prototype mining for universal domain adaptation. In *International Conference on Learning Representations*, 2024.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 2015.

Li, G., Kang, G., Zhu, Y., Wei, Y., and Yang, Y. Domain consensus clustering for universal domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2021.

Ma, X., Gao, J., and Xu, C. Active universal domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023.

Panareda Busto, P. and Gall, J. Open set domain adaptation. In *IEEE International Conference on Computer vision*, 2017.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 2019.

Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer vision*, 2019.

Qu, S., Zou, T., Röhrbein, F., Lu, C., Chen, G., Tao, D., and Jiang, C. Upcycling models under domain and category shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Rambhatla, S. S., Chellappa, R., and Shrivastava, A. The pursuit of knowledge: Discovering and localizing novel categories using dual memory. In *IEEE International Conference on Computer vision*, 2021.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. Adapting visual category models to new domains. In *European Conference on Computer Vision*, 2010.

Saito, K. and Saenko, K. Ovanet: One-vs-all network for universal domain adaptation. In *IEEE Conference on International Conference on Computer Vision*, 2021.

Saito, K., Yamamoto, S., Ushiku, Y., and Harada, T. Open set domain adaptation by backpropagation. In *European Conference on Computer Vision*, 2018.

Saito, K., Kim, D., Sclaroff, S., and Saenko, K. Universal domain adaptation through self supervision. *Advances in Neural Information Processing Systems*, 2020.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks*, 2015.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016.

Singh, M., Gustafson, L., Adcock, A., de Freitas Reis, V., Gedik, B., Kosaraju, R. P., Mahajan, D., Girshick, R., Dollár, P., and Van Der Maaten, L. Revisiting weakly supervised pre-training of visual perception models. In *European Conference on Computer Vision*, 2022.

Torrey, L. and Shavlik, J. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI global, 2010.

Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., and Van Gool, L. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, 2020.

Vaze, S., Han, K., Vedaldi, A., and Zisserman, A. Generalized category discovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. A survey of transfer learning. *Journal of Big Data*, 2016.

You, K., Long, M., Cao, Z., Wang, J., and Jordan, M. I. Universal domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Yu, Q., Ikami, D., Irie, G., and Aizawa, K. Self-labeling framework for novel category discovery over domains. In *AAAI Conference on Artificial Intelligence*, 2022.

Yu, Q., Irie, G., and Aizawa, K. Open-set domain adaptation with visual-language foundation models. *arXiv preprint arXiv:2307.16204*, 2023.

Zang, Z., Shang, L., Yang, S., Wang, F., Sun, B., Xie, X., and Li, S. Z. Boosting novel category discovery over domains with soft contrastive learning and all in one classifier. In *IEEE International Conference on Computer vision*, 2023.

Zhu, D., Li, Y., Yuan, J., Li, Z., Kuang, K., and Wu, C. Universal domain adaptation via compressive attention matching. In *IEEE International Conference on Computer vision*, 2023.

A. Implementation details

A.1. Training details

Our core algorithm is developed using PyTorch (Paszke et al., 2019). We use CLIP ViT-L14-336px as the backbone for all the methods. When fine-tuning, we update only the last two blocks of CLIP ViT-L14-336px and freeze the other parts. For the classifier, CROW uses the normalized linear classifier as described in Section 3.2. For the baselines, we use the same classifier architecture originally proposed by their works, and we only change the input dimension of the classifiers to match the feature dimension.

For optimizing, we use the SGD optimizer for all experiments, and the learning rate is set to 0.001 for the classifier and 0.0001 for the feature extractor (CLIP ViT-L14-336px). We set the batch size to 32 and train all the methods for 1K iterations. Since there is no validation set in our setting, we report the results of the last iteration.

A.2. Implementation details about baselines

We directly apply the OW-SSL methods GCD and ORCA and the method NCDDA to our experiment settings. However, for the other baselines, we adapt them to address our problem setting. The detailed procedures for adaptation are outlined below.

SIMPLE. SIMPLE shares the same network architecture as CROW (a feature extractor f_θ and a normalized linear classifier W), and it uses the match-then-cluster approach. Specifically, it first trains the classifier W on the source set with cross-entropy loss (Equation 3). Importantly, since SIMPLE trains the model only on the source set, it is likely to make the predictions biased to the seen classes. To prevent this, we freeze the feature extractor from SIMPLE.

After training the classifier, we predict the labels and compute the entropy for all the samples. Specifically, given input x, we first extract the feature by $z = f_\theta(x)$. Then, we calculate the output vector $p(y|x)$ using $p(y|x) = \sigma(W^T \cdot z)$, where σ is the softmax activate function. We predict the label using $c = \arg\max_i p^i$. Then, we calculate the entropy H for the output vector $p(y|x)$. If H is larger than a predefined threshold τ, we assign this sample to be an unseen sample. Note that since DANCE also applies a threshold to the entropy of prediction, we use the ρ in DANCE as the τ here. After predicting labels for all samples, we cluster detected unseen samples using K-means with $K = |C_t| - |C_s|$ to discover novel classes. We assume the number of target classes $|C_t|$ is given as a prior. After labeling all the samples from seen and unseen classes, we finetune SIMPLE using the same objective function 5 as in CROW.

UniDA methods (except DCC). For all the UniDA methods except DCC (DANCE, OVANet, UniOT, SAN, GLC), we use them as the match-then-cluster approach. Specifically, we first apply the methods to predict the labels of the target samples. Then, each target sample is labeled as a seen class or the class unseen. After labeling, we cluster all the samples labeled as class unseen using K-means with $K = |C_t| - |C_s|$ to discover novel classes. We assume the number of target classes $|C_t|$ is given as a prior.

DCC. Different from the other UniDA methods, we use DCC as a cluster-then-match approach. Thus, we follow the original steps of DCC. We change only one thing: in the original work, DCC estimates the number of target classes $|C_t|$, but we directly use $|C_t|$ as a prior in DCC for a fair comparison.

A.3. Threshold adaptation

As mentioned in Section 4.1, we adapt the threshold τ for the baselines when needed. Table 8 shows how we change τ for the baseline methods to obtain balanced seen and unseen accuracy. τ is the original threshold provided by the previous works, and we scale it with a multiplication factor. (τ is the ρ in DANCE and SIMPLE, 0.5 with no name in OVANet.)

Table 8. Hyper-parameter changing. τ is the original threshold provided by the previous works.

	OFFICE	OFFICEHOME	VISDA	DOMAINNET
SIMPLE	0.3τ	0.5τ	0.3τ	0.7τ
DANCE	0.3τ	0.5τ	0.3τ	0.7τ
OVANET	-	-	1.5τ	-

B. Additional results

B.1. Comparison to the K-means

This section shows the results and analysis of comparing our method CROW to applying K-means to the CLIP features.

Table 9. H-score (%) comparison between K-means and CROW.

	OFFICE	OFFICEHOME	VISDA	DOMAINNET
K-MEANS	77.2	65.9	62.4	52.7
CROW	84.9	69.9	69.2	59.0

Table 9 shows that our method outperforms K-means by a large margin. Moreover, it is important to note that our method labels the seen classes while applying K-means to the CLIP features only separates different classes without matching the clusters to the seen classes, which means our H-score is tested on a harder task.

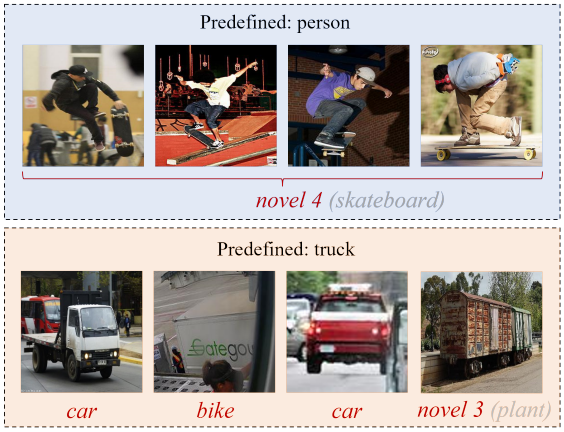

Figure 6. **Failure cases** on VisDA (6/6). The samples are the top 4 confident samples (top 1 to 4 from left to right) for class 'person' and 'truck'.

B.2. Comparison to the CLIP zero-shot learning

Another simple way to address cross-domain open-world discovery is to apply CLIP zero-shot learning with a large vocabulary list. Here, we show the results and analysis of comparing our method CROW to CLIP zero-shot learning.

We do two experiments on the Office31 dataset. In the first experiment, the vocabulary list of CLIP contains only the names of the 31 classes. In the second experiment, we create a large vocabulary list for CLIP by combining the names of the 31 classes from the Office dataset and the 345 classes from the DomianNet dataset. We remove the names of the duplicate classes from the vocabulary list. We consider the first 16 classes from the Office31 dataset seen and the last 15 classes unseen. Table 10 shows the results. The results show that our method achieves a comparable performance even if we provide CLIP with an exact ground truth vocabulary list, and our method outperforms CLIP by a large margin if CLIP uses a large-enough vocabulary list.

Table 10. H-score (%) comparison between CLIP zero-shot learning and CROW.

	H-SCORE
CLIP ZERO-SHOT + 31 CLASSES VOCABULARY LIST	85.7
CLIP ZERO-SHOT + LARGE VOCABULARY LIST	75.4
CROW	84.9

B.3. Failure cases

In Figure 4, we show that we successfully discover four predefined classes out of six. Figure 6 shows the possible reason why we do not manage to discover the other two predefined classes 'person' and 'truck'. For the predefined class 'person', we can see that the top four confident samples are all persons with skateboards, and they are classified into class 'novel 4', which is 'skateboard' as shown in Figure 4. For the predefined class 'truck', we can see that two of them are classified into 'car', and this might be because they share lots of common features with cars; one is classified into 'bike', a possible explanation is because of the green logo 'Gate' is close to bike; one is classified into class 'novel 3, which is 'plant', possibly because of the full-of-tree background.

B.4. Sensitivity to λ

There is a hyper-parameter λ in Equation 5, which stands for the weight for the regularization term. Table 11 shows that our method is robust to λ as long as it is not set to be extremely huge or tiny.

Table 11. Seen accuracy (%), unseen accuracy (%), and H-score (%) of CROW with different λ on the VisDA (6/6).

λ	SEEN	UNSEEN	H-SCORE
0.1	78.5	62.0	69.3
0.3	78.8	61.3	69.0
0.5	77.8	61.7	68.8
0.7	77.6	62.7	69.4
0.9	77.3	62.6	69.2
1.0	77.0	62.8	69.2

B.5. CROW with the estimated number of clusters

In the clustering step, we assume we know the number of unseen classes $|C^t|$ following the OW-SSL setting. However, in practice, we sometimes do not know the real $|C^t|$. Under this condition, we need to estimate $|C^t|$.

We estimate $|C^t|$ using the technique proposed in (Han et al., 2019). In the original work, it estimates $|C^t|$ by applying K-means with different K on both source and target samples. Then, it tests the cluster accuracy using Hungarian algorithm (Kuhn, 1955) for the labeled samples and selects the K that leads to the best cluster accuracy. However, since there is a domain shift between source and target set in our problem setting, we apply K-means only on the target data instead of both source and target data. Other steps remain the same. Table 12 shows the results of using the real $|C^t|$ and the estimated $|C^t|$, and we can see that the results are still good with estimated $|C^t|$.

Table 12. H-score of using estimated $|C^t|$. We show 50% seen/unseen split as an example.

	OFFICE	OFFICEHOME	VISDA	DOMAINNET
KNOWN	84.9	69.6	69.2	59.0
ESTIMATED	81.5	67.2	68.0	57.8

B.6. Evaluation on the UniDA data split.

Since UniDA is the closest setting to CD-OWD, we demonstrate the efficacy of our method on the UniDA data split.

There are four possible relationships: closed set (Ganin & Lempitsky, 2015), partial set (Cao et al., 2019), open set (Panareda Busto & Gall, 2017), and open partial set (You et al., 2019). The term *partial* means there are classes that exist only in the source set. UniDA setting considers all four possible relationships between source and target sets. Thus, we test our method on the four conditions of closed set, partial set, open set, and open partial set on the VisDA dataset. We use the same evaluation metric as the previous experiments. Note that we assume we know the number of classes of the target sets but do not know the relationship between source and target sets. For example, in the condition of the closed set, we assume we know that there are 12 classes in the target set. However, we do not know if there are novel classes, so we will still detect unseen. Table 13 shows that our method achieves comparable performance with UniDA data split.

Table 13. Results of UniDA data split on VisDA. The numbers of (shared classes/source private classes/target private classes) for close-set, partial-set, open-set, and open-partial-set are 12/0/0, 6/6/0, 6/0/6, and 6/3/3. The results are H-score when the number of target private classes is not zero; otherwise, we show the accuracy of shared classes.

	CLOSE	PARTIAL	OPEN	OPEN-PARTIAL
DCC	80.1	79.8	57.3	65.2
OVANET	78.0	73.2	60.8	61.2
GLC	75.6	81.2	65.2	72.2
CROW	79.9	80.4	69.2	73.4

B.7. Detailed experimental results

Table 2 only shows the average H-score on each dataset. Here, we show the detailed results of each categorical-shift and domain-shift scenario on different datasets in Table 14 to 22. Each table shows the result of one dataset with one data split. For example, Office (21/10) shows the result of the Office dataset with a 21/10 seen/unseen data split. In each table, we show all the domain-shift scenarios. For example, there are three domains in the Office dataset, Amazon (A), DSLR (D), and Webcam (W). Then, there are six pairs: A2D, A2W, D2A, D2W, W2A, and W2d, where A2D means from Amazon (source set) to DSLR (target set).

Table 3 only shows the average seen accuracy, unseen accuracy, and H-score on each dataset of 50% seen/unseen split. Here, we show the average seen accuracy, unseen accuracy, and H-score on each dataset of 50% seen/unseen split in Table 14 to 22.

Table 14. H-score (%) of dataset Office (21/10) on different pairs of domain. We color the best and second-best results in red and blue.

	A2D	A2W	D2A	D2W	W2A	W2D	AVG.
SIMPLE	62.6	60.8	43.4	86.2	42.3	90.3	64.9
GCD	62.6	58.5	58.0	48.7	47.5	48.1	31.2
ORCA	65.9	52.5	51.5	65.3	54.8	52.1	57.9
DCC	66.3	72.5	69.8	84.5	66.9	76.8	72.8
DANCE	79.4	77.5	61.4	82.9	62.3	87.8	75.4
OVANET	77.4	67.1	59.8	88.0	55.9	90.7	73.2
UNIOT	82.8	70.7	65.3	78.1	66.5	91.5	76.1
NCDDA	81.2	76.4	78.7	80.8	76.0	87.1	80.3
SAN	80.7	76.7	77.8	81.4	80.6	85.2	80.5
GLC	78.5	71.9	73.2	72.6	73.4	81.9	75.5
CROW	87.9	90.3	70.3	93.0	70.2	92.8	84.7

Table 15. H-score (%) of dataset Office (16/15) on different pairs of domain. We color the best and second-best results in red and blue.

	A2D	A2W	D2A	D2W	W2A	W2D	AVG.
SIMPLE	65.8	57.6	47.6	84.8	47.8	90.6	66.8
GCD	58.8	37.0	50.6	76.6	46.6	69.9	58.5
ORCA	61.4	67.4	54.3	80.3	46.1	65.9	63.4
DCC	68.3	72.5	71.8	85.5	68.9	77.8	74.9
DANCE	65.7	67.3	59.7	83.4	54.1	82.7	68.9
OVANET	81.6	76.3	61.5	91.2	51.2	90.6	75.7
UNIOT	79.6	83.8	72.7	90.9	75.0	85.7	81.5
NCDDA	83.9	76.4	80.0	78.7	81.8	85.5	81.2
SAN	79.5	76.0	78.3	80.3	80.6	85.5	80.2
GLC	85.4	74.2	69.8	79.1	63.2	71.3	74.6
CROW	83.8	85.9	73.9	94.8	78.1	91.2	84.9

Table 16. H-score (%) of dataset Office (10/21) on different pairs of domain. We color the best and second-best results in red and blue.

	A2D	A2W	D2A	D2W	W2A	W2D	AVG.
SIMPLE	76.7	78.9	71.9	84.1	71.1	85.5	78.3
GCD	53.0	38.2	51.0	77.5	40.9	70.9	58.0
ORCA	61.0	57.3	44.9	75.0	60.2	69.0	62.3
DCC	69.3	76.5	74.8	86.5	69.5	74.8	75.2
DANCE	77.6	73.3	58.2	74.5	56.8	77.3	69.9
OVANET	84.8	78.1	50.7	90.8	48.6	90.0	75.3
UNIOT	84.1	84.5	69.9	90.9	77.2	93.7	83.4
NCDDA	84.1	75.0	80.6	80.3	82.3	87.1	81.7
SAN	80.8	79.7	81.0	83.7	81.3	85.6	82.0
GLC	82.9	77.5	72.6	77.3	73.0	78.5	77.3
CROW	85.8	82.9	77.5	96.0	79.1	91.7	85.6

Table 17. H-score (%) of dataset DomainNet (240/105) on different pairs of domain. We color the best and second-best results in red and blue.

	P2R	P2S	R2P	R2S	S2P	S2R	AVG.
SIMPLE	58.0	45.7	52.2	54.8	48.9	58.7	53.2
GCD	39.4	28.7	33.6	36.4	34.1	36.9	35.0
ORCA	32.7	20.4	21.0	33.3	28.4	36.0	28.9
DCC	52.4	42.6	44.5	44.1	41.1	47.3	45.5
DANCE	61.4	52.3	55.6	52.1	54.3	61.5	56.3
OVANET	59.6	49.1	54.0	49.9	51.8	59.2	54.2
UNIOT	49.1	42.6	45.8	43.4	42.8	49.6	45.7
NCDDA	61.4	47.9	41.6	45.8	41.0	61.3	50.1
SAN	62.7	51.2	45.2	52.5	42.8	63.3	53.2
GLC	61.3	51.5	46.4	53.1	48.0	65.6	54.9
CROW	68.0	52.3	51.7	49.2	53.6	69.9	57.8

Table 18. H-score (%) of dataset DomainNet (173/172) on different pairs of domain. We color the best and second-best results in red and blue.

	P2R	P2S	R2P	R2S	S2P	S2R	AVG.
SIMPLE	59.6	53.4	53.5	56.3	52.5	59.6	55.9
GCD	45.9	33.7	41.1	38.6	39.0	48.5	41.3
ORCA	36.7	30.4	32.1	30.9	27.7	31.1	31.5
DCC	53.6	42.7	45.9	45.6	43.7	50.9	47.5
DANCE	60.2	51.1	54.1	52.5	53.9	61.8	55.7
OVANET	61.3	50.6	53.9	52.6	52.8	61.3	55.6
UNIOT	54.3	45.4	52.3	48.1	50.5	55.1	51.2
NCDDA	61.2	49.1	45.2	50.4	48.8	57.5	52.6
SAN	62.6	53.4	49.1	51.0	47.8	64.2	54.8
GLC	62.1	51.4	53.3	54.4	51.9	60.8	56.1
CROW	67.9	54.2	53.7	51.7	54.8	70.4	59.0

Table 19. H-score (%) of dataset DomainNet (105/240) on different pairs of domain. We color the best and second-best results in red and blue.

	P2R	P2S	R2P	R2S	S2P	S2R	AVG.
SIMPLE	65.0	55.1	57.0	56.9	52.8	59.9	57.8
GCD	48.0	30.6	40.3	37.2	38.3	52.5	41.3
ORCA	33.2	37.1	29.4	29.2	30.4	42.2	33.7
DCC	57.0	41.4	43.8	42.4	44.8	55.6	47.7
DANCE	64.7	54.5	57.2	54.4	56.2	65.4	58.8
OVANET	64.7	56.0	55.4	53.8	54.5	66.1	58.5
UNIOT	54.5	50.4	48.1	48.6	47.6	55.6	50.8
NCDDA	67.0	54.3	50.7	47.2	49.2	64.1	55.7
SAN	63.7	53.9	49.9	49.9	51.3	60.9	55.0
GLC	62.8	50.7	53.4	50.8	50.8	62.3	55.7
CROW	71.3	56.5	56.6	54.9	58.2	70.8	61.5

Table 20. H-score (%) of dataset OfficeHome (45/20) on different pairs of domain. We color the best and second-best results in red and blue.

	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	AVG.
SIMPLE	55.4	68.2	74.9	58.7	69.4	63.1	47.8	47.1	69.6	59.6	55.6	72.9	62.3
GCD	39.4	56.1	64.4	37.0	45.8	47.3	39.4	32.2	60.8	34.2	43.2	62.0	48.7
ORCA	44.7	60.8	50.9	43.6	53.9	44.4	44.4	42.5	61.4	43.1	43.2	45.5	48.9
DCC	60.2	69.3	68.2	46.3	74.7	69.9	48.5	61.9	69.3	52.6	64.2	76.0	63.6
DANCE	56.6	72.5	78.0	65.7	70.4	68.5	59.5	55.8	70.2	65.1	57.7	65.4	65.7
OVANET	57.2	71.1	78.2	63.5	67.6	73.7	55.0	48.4	73.7	66.9	59.6	77.2	66.4
UNIOT	58.1	70.4	72.9	57.3	69.3	65.6	54.3	55.1	59.3	67.5	63.1	76.9	64.4
NCDDA	64.4	76.3	64.5	53.5	73.4	55.4	48.7	54.6	60.1	60.2	59.0	81.5	63.2
SAN	57.8	69.8	71.8	53.9	72.1	71.0	52.0	55.9	63.3	61.7	61.9	76.9	64.3
GLC	59.3	71.9	61.9	57.7	73.3	61.7	54.3	58.7	66.6	64.0	56.7	73.5	65.2
CROW	63.1	82.5	79.5	48.3	83.1	75.6	51.8	64.7	75.2	54.8	67.8	84.5	69.4

Table 21. H-score (%) of dataset OfficeHome (33/32) on different pairs of domain. We color the best and second-best results in red and blue.

	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	AVG.
SIMPLE	59.9	72.2	77.8	64.3	73.5	65.1	57.0	55.8	70.1	63.4	58.3	72.9	66.0
GCD	44.2	46.7	59.6	40.8	46.6	50.1	30.7	28.2	55.1	39.3	45.5	62.0	47.5
ORCA	47.4	62.5	44.9	44.0	52.3	46.7	38.4	29.3	52.5	52.2	48.5	63.3	48.9
DCC	64.1	74.9	71.1	48.5	78.1	70.6	51.1	58.6	68.0	48.8	60.5	82.9	65.0
DANCE	52.9	70.8	74.3	66.9	69.3	73.4	59.5	54.2	70.9	67.0	52.8	72.0	65.6
OVANET	57.9	72.5	78.4	67.6	69.6	73.8	57.8	53.2	74.9	69.4	61.1	82.2	68.6
UNIOT	59.5	72.1	71.0	57.1	74.2	63.2	52.4	56.8	67.5	65.8	58.9	76.9	64.9
NCDDA	64.9	76.8	64.8	56.7	78.1	55.7	49.7	55.2	60.1	61.2	59.8	82.4	64.3
SAN	60.5	70.8	72.2	56.3	71.6	69.4	54.7	59.0	63.3	61.5	59.5	77.4	65.0
GLC	64.4	71.8	78.9	51.3	76.0	76.6	48.2	63.5	80.4	55.9	60.0	75.4	68.2
CROW	66.5	83.4	77.7	50.3	82.2	75.4	53.2	62.6	74.0	56.2	66.2	84.2	69.6

Table 22. H-score (%) of dataset OfficeHome (20/45) on different pairs of domain. We color the best and second-best results in red and blue.

	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	AVG.
SIMPLE	58.7	70.5	76.6	63.8	72.9	64.6	57.0	56.5	69.1	63.4	57.7	72.9	65.4
GCD	37.8	51.8	59.3	46.6	46.3	49.4	26.0	18.3	54.3	49.9	45.2	69.4	48.1
ORCA	42.4	55.4	57.0	60.2	36.7	62.2	40.7	44.1	55.9	47.3	22.2	58.0	49.9
DCC	63.9	74.4	71.0	47.6	78.4	70.1	50.9	58.5	67.7	50.2	59.8	81.3	64.7
DANCE	55.7	72.0	76.4	66.7	69.5	71.4	60.7	51.4	73.5	71.3	56.9	72.9	67.1
OVANET	58.7	72.2	76.7	66.9	70.3	73.5	58.4	56.3	77.5	69.5	62.4	74.6	68.7
UNIOT	58.0	72.2	68.1	57.1	75.6	63.6	49.4	59.2	67.5	66.0	59.1	77.7	64.8
NCDDA	65.9	78.3	66.0	58.4	76.6	57.3	51.5	59.1	60.1	61.2	61.9	78.9	65.0
SAN	63.4	78.3	74.1	56.9	76.0	73.0	57.8	56.5	67.9	61.8	59.4	78.9	67.2
GLC	69.2	85.8	83.7	64.0	75.2	77.8	55.2	52.9	74.5	62.3	57.2	72.2	69.3
CROW	69.2	79.0	78.0	52.6	81.7	73.7	53.1	68.1	76.8	56.6	69.6	82.6	70.2

Table 23. Average seen accuracy (%), unseen accuracy (%), and H-score (%) of 70% seen/unseen splits on dataset Office, OfficeHome, VisDA, and DomainNet. We color the best and second-best results in red and blue.

	OFFICE (21/10)			OFFICEHOME (45/20)			VISDA (8/4)			DOMAINNET (240/105)		
	SEEN	UNSEEN	H-SCORE	SEEN	UNSEEN	H-SCORE	SEEN	UNSEEN	H-SCORE	SEEN	UNSEEN	H-SCORE
SIMPLE	64.6	65.3	64.9	65.3	59.5	62.3	57.3	53.7	55.4	69.1	43.2	53.2
GCD	67.6	58.4	62.6	50.9	46.7	48.7	31.8	30.6	31.2	39.9	31.2	35.0
ORCA	74.0	47.6	57.9	69.8	37.6	48.9	65.2	20.6	31.3	58.1	19.3	28.9
DCC	74.6	71.1	72.8	66.1	61.3	63.6	73.9	51.5	60.7	45.7	45.4	45.5
DANCE	79.2	71.9	75.4	74.2	58.9	65.7	64.9	51.2	57.2	67.9	48.1	56.3
OVANET	78.7	68.5	73.2	67.9	65.1	66.4	56.5	63.7	59.9	60.2	49.2	54.2
UNIOT	81.7	71.2	76.1	70.6	59.3	64.4	72.3	54.2	62.0	49.1	42.7	45.7
NCDDA	91.6	71.5	80.3	66.9	59.8	63.2	67.3	49.8	57.2	58.1	44.1	50.1
SAN	93.1	71.0	80.5	68.8	60.4	64.3	69.3	54.8	61.2	67.3	44.0	53.2
GLC	89.4	65.4	75.5	73.8	58.4	65.2	58.6	64.1	61.2	61.9	49.3	54.9
CROW	90.9	79.2	84.7	68.6	70.3	69.4	76.8	65.1	70.5	69.8	49.3	57.8

Table 24. Average seen accuracy (%), unseen accuracy (%), and H-score (%) of 30% seen/unseen splits on dataset Office, OfficeHome, VisDA, and DomainNet. We color the best and second-best results in red and blue.

	OFFICE (10/21)			OFFICEHOME (20/45)			VISDA (4/8)			DOMAINNET (105/240)		
	SEEN	UNSEEN	H-SCORE	SEEN	UNSEEN	H-SCORE	SEEN	UNSEEN	H-SCORE	SEEN	UNSEEN	H-SCORE
SIMPLE	86.3	71.8	78.3	72.0	59.9	65.4	58.4	45.1	50.9	76.6	46.4	57.8
GCD	49.7	69.6	58.0	42.1	56.1	48.1	29.6	21.9	25.2	40.4	42.3	41.3
ORCA	71.2	55.4	62.3	67.1	39.7	49.9	69.3	22.4	33.9	68.7	22.3	33.7
DCC	73.8	76.6	75.2	72.1	58.9	64.7	65.8	49.8	56.7	57.5	40.7	47.7
DANCE	83.5	60.1	69.9	69.1	65.3	67.1	75.2	35.7	48.4	71.3	50.1	58.8
OVANET	74.7	75.9	75.3	72.7	65.2	68.7	62.3	58.2	60.2	70.3	50.1	58.5
UNIOT	90.3	77.5	83.4	73.9	57.6	64.8	75.7	49.4	59.8	61.6	43.3	50.8
NCDDA	93.4	72.5	81.7	71.8	59.4	65.0	70.8	51.0	59.3	70.1	46.2	55.7
SAN	95.7	71.8	82.0	77.1	59.5	67.2	74.3	52.4	61.5	68.7	45.8	55.0
GLC	91.7	66.9	77.3	77.7	62.6	69.3	76.3	53.2	62.7	65.7	48.3	55.7
CROW	88.8	82.6	85.6	70.4	70.0	70.2	68.9	73.4	71.1	72.5	53.5	61.5