模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

ParNeC
模式识别与神经计算研究组
PAttern Recognition and NEural Computing

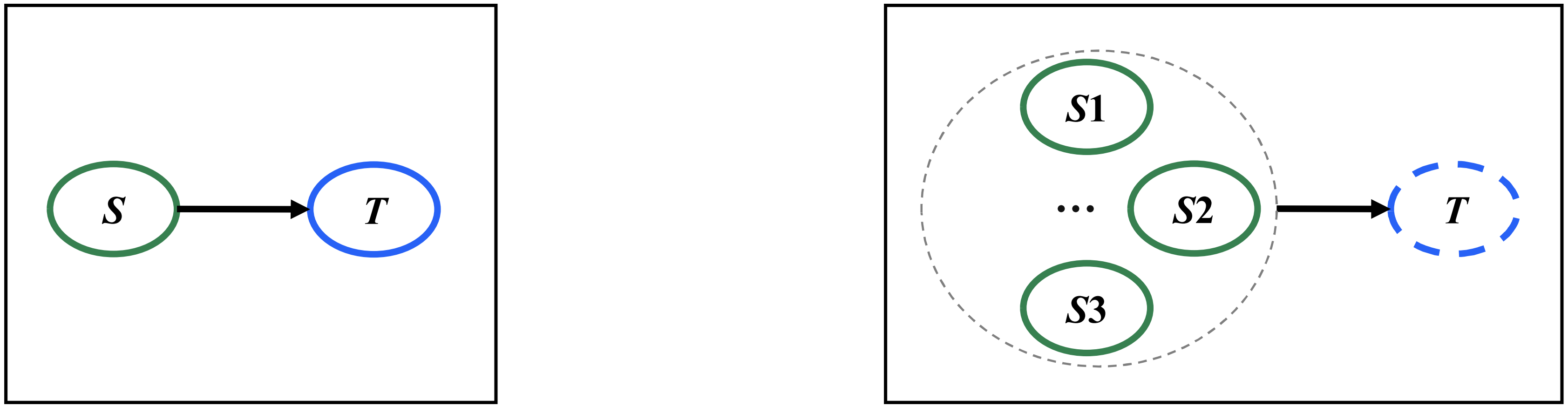
Semi-Supervised Domain Generalization with Known and Unknown Classes

Lei Zhang, Ji-Fu Li, Wei Wang*
National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
`{zhangl,lijf,wangw}@lamda.nju.edu.cn`

NIPS 2023

Background

Domain Adaptation (DA) *vs* Domain Generalization (DG)



S — source domain

T — *seen* target domain

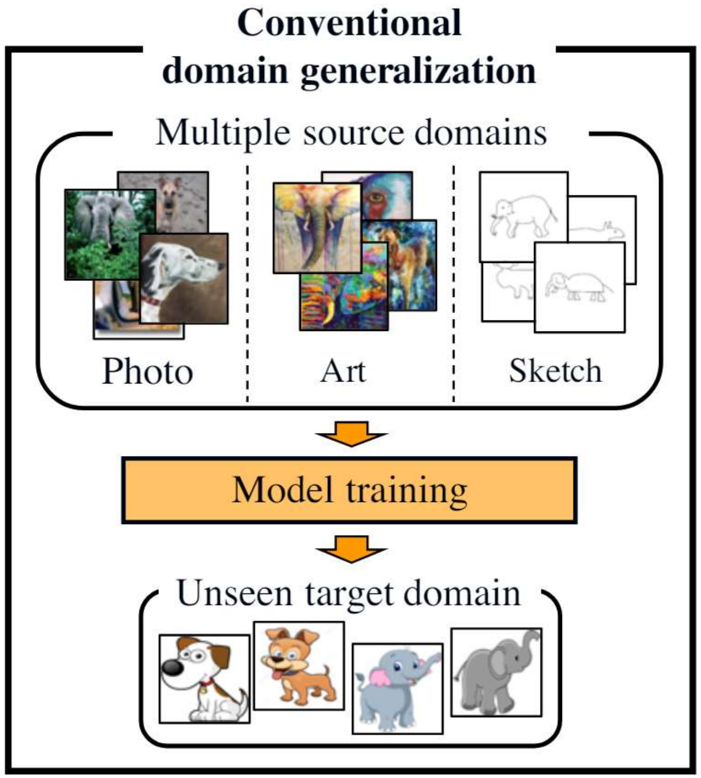
T — *unseen* target domain

Difference: ① In DA, both the source domain and target domain are accessible (There is only unlabeled target domain data in unsupervised DA, *i.e.*, UDA);

② In DG, the target domain *are not accessible*!

Background

Conventional Domain Generalization (DG)

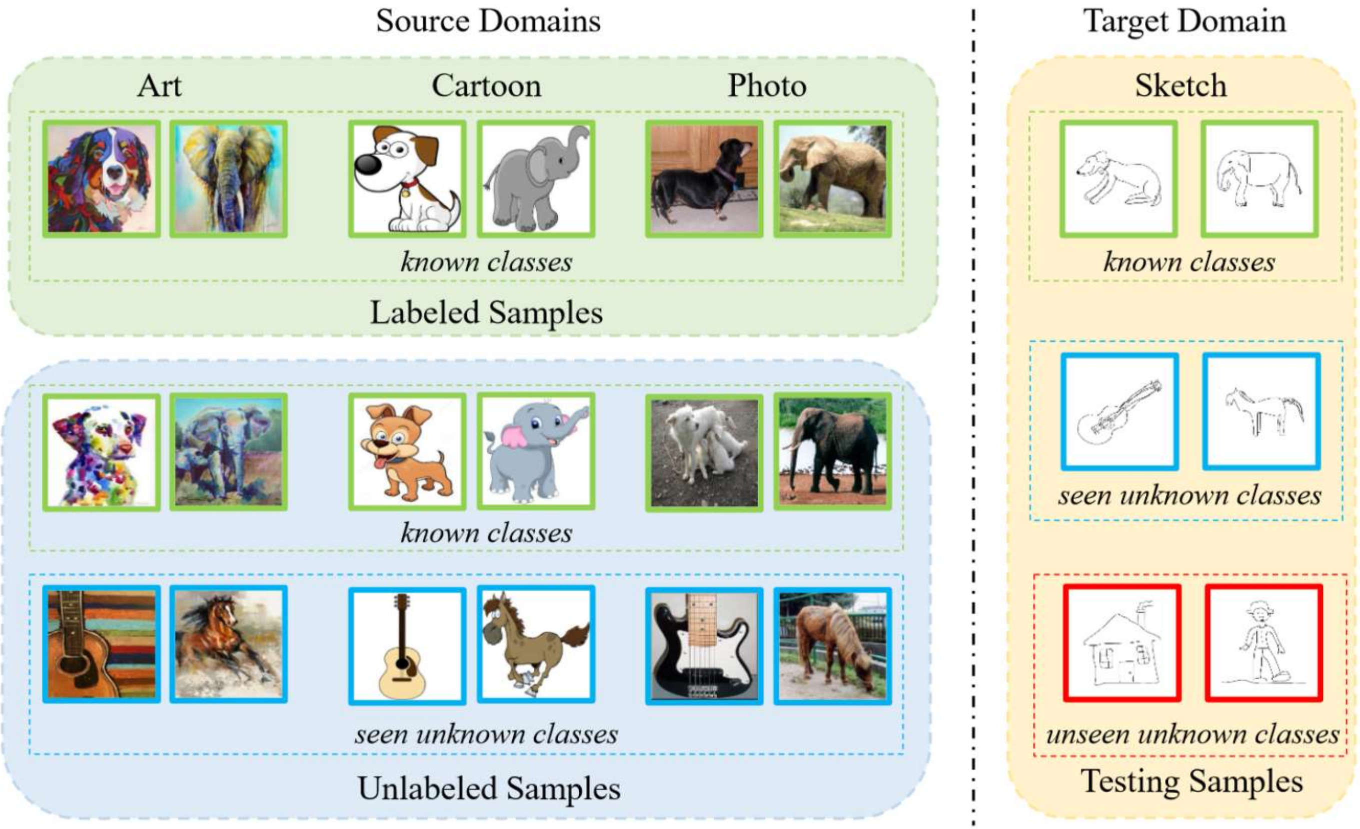


Y: {‘Gog’, ‘Elephant’}

All classes are ***known*** in
source domain and target domain !

Matsuura T, Harada T. Domain generalization using a mixture of multiple latent domains[C] AAAI 2020

Semi-Supervised Domain Generalization (SSDG) + *unknown classes*



known classes (green box) 🟩

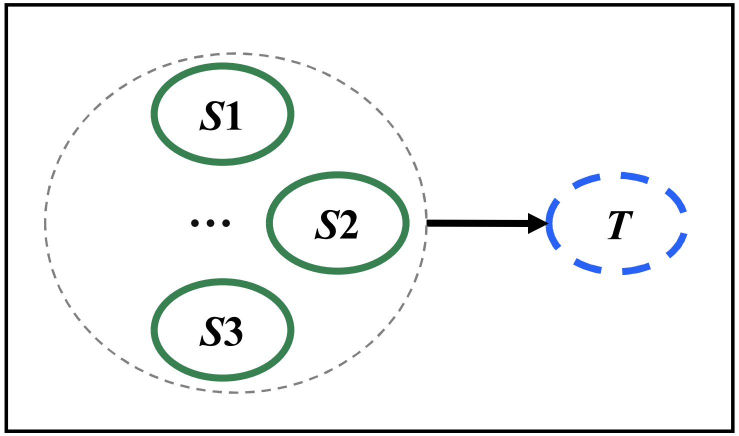
seen unknown classes (blue box) 🟦

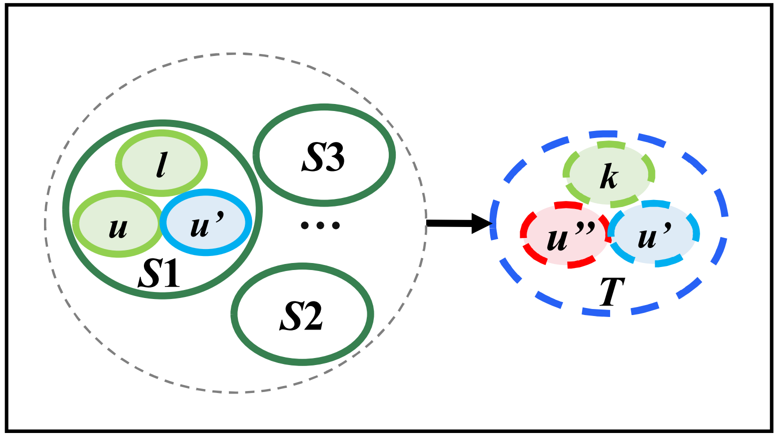
unseen unknown classes (red box) 🟥

The label set: $\mathcal{C}^l \subset \mathcal{C}^u \subset \mathcal{C}^t$

Background

Domain Generalization (DG) *vs* Semi-Supervised DG + *unknown classes*





Source:

- *l* — labeled & known
- *u* — unlabeled & known
- *u'* — unlabeled & unknown

Target:

- *k* — known & seen
- *u'* — unknown & seen
- *u''* — unknown & unseen

Method

Out-of-Distribution + Semi-Supervised + Domain Generalization

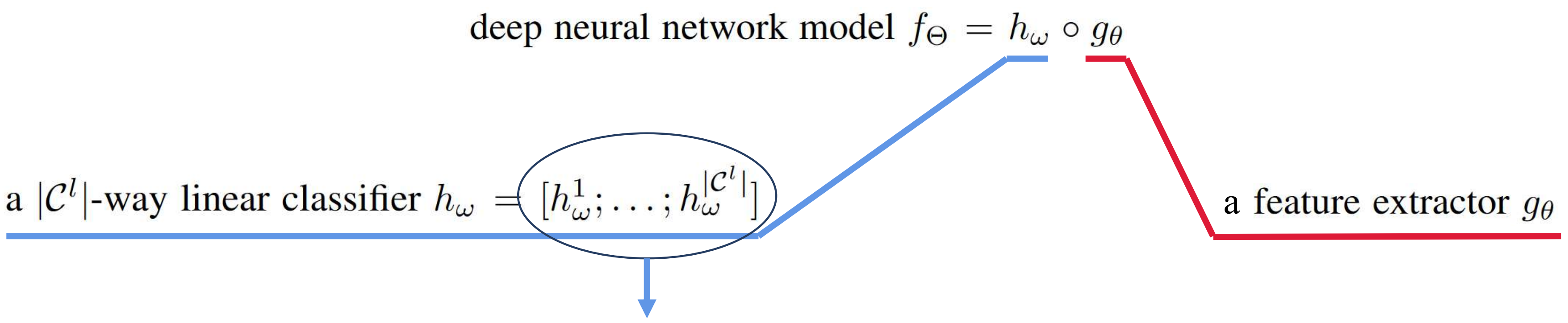
① **Detecting Known and Unknown Classes**

② **Improving Target Domain Generalization**

Method name： *Class-Wise Adaptive Exploration and Exploitation* (**CWAEE**)

Method

① Detecting Known and Unknown Classes



deep neural network model $f_\Theta = h_\omega \circ g_\theta$

a $|\mathcal{C}^l|$-way linear classifier $h_\omega = [h_\omega^1; \ldots; h_\omega^{|\mathcal{C}^l|}]$

a feature extractor g_θ

Since <u>no extra classifiers</u> are designed for new classes,

a naive idea is to

I) replace the *softmax* function with the *sigmoid* function; $\{\ \hat{p}_i^c = \sigma(z_i^c)\ \}$

II) set a fixed threshold $\delta = 0.5$ for known and unknown classes. $\{\ \hat{p}_i^c \geq \delta$ as known $\}$

$\{\ \hat{p}_i^c < \delta$ as unknown $\}$

Method

① Detecting Known and Unknown Classes

$$\tau^c = \operatorname{argmin} - \sum_{(\mathbf{x}_i, y_i)\in\mathcal{D}^v} \mathbb{I}\left(y_i = c\right)\log\left(\sigma(\mathbf{z}_i^c/\tau^c)\right) + \mathbb{I}\left(y_i \neq c\right)\log\left(1-\sigma(\mathbf{z}_i^c/\tau^c)\right)$$

To get well-calibrated probabilities :

I) $\hat{p}_i^c = \sigma(z_i^c)$ → $\tilde{p}_i^c = \sigma(z_i^c/\tau^c)$

A more reasonable way is to use the class-wise adaptive threshold :

II) $\delta = 0.5$ → $\underline{\delta_{knw}^j \text{ and } \delta_{unk}^j}$

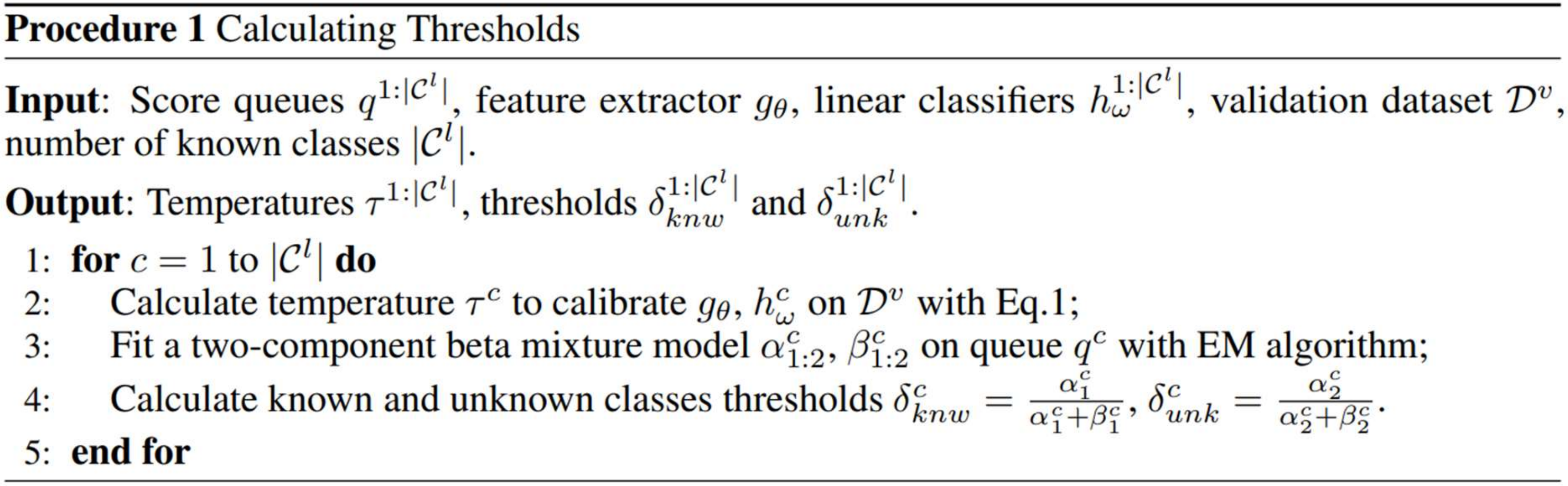
$$\begin{cases} \hat{p}_i^j \geq \delta_{knw}^j & \text{as known classes} \\ \hat{p}_i^j < \delta_{unk}^j & \text{as unknown classes} \\ \delta_{knw}^j \leq \hat{p}_i^j < \delta_{knw}^j & \text{as } \textbf{\textit{null}} \end{cases}$$

use a two-component beta mixture model to model the score distributions of known classes and unknown classes

Method

① Detecting Known and Unknown Classes

δ_{knw}^{j} and δ_{unk}^{j} can be set as the *mean* values of two fitted beta distributions.



Procedure 1 Calculating Thresholds

Input: Score queues $q^{1:|\mathcal{C}^l|}$, feature extractor g_θ, linear classifiers $h_\omega^{1:|\mathcal{C}^l|}$, validation dataset $\mathcal{D}^v$, number of known classes $|\mathcal{C}^l|$.

Output: Temperatures $\tau^{1:|\mathcal{C}^l|}$, thresholds $\delta_{knw}^{1:|\mathcal{C}^l|}$ and $\delta_{unk}^{1:|\mathcal{C}^l|}$.

```
for $c = 1$ to $|\mathcal{C}^l|$ do
    Calculate temperature $\tau^c$ to calibrate $g_\theta$, $h_\omega^c$ on $\mathcal{D}^v$ with Eq.1;
    Fit a two-component beta mixture model $\alpha_{1:2}^c, \beta_{1:2}^c$ on queue $q^c$ with EM algorithm;
    Calculate known and unknown classes thresholds $\delta_{knw}^c = \frac{\alpha_1^c}{\alpha_1^c+\beta_1^c}, \delta_{unk}^c = \frac{\alpha_2^c}{\alpha_2^c+\beta_2^c}$.
end for
```

Method

② Improving Target Domain Generalization

The overall loss of the training process is formulated as:

$$\mathcal{L} = \underline{\mathcal{L}^l} + \underline{\lambda_1 \mathcal{L}^u_{knw} + \lambda_2 \mathcal{L}^u_{unk}} + \underline{\lambda_3 \mathcal{L}^u_{con}}$$

For labeled training data, we use the following supervised loss,

$$\mathcal{L}^l = -\frac{1}{n^l} \sum_{i=1}^{n^l} \sum_{c=1}^{|\mathcal{C}^l|} \mathbb{I}(y_i = c) \ln(\hat{\mathbf{p}}_i^c) + \frac{1}{|\mathcal{C}^l| - 1} \mathbb{I}(y_i \neq c) \ln(1 - \hat{\mathbf{p}}_i^c)$$

Method

② Improving Target Domain Generalization

The overall loss of the training process is formulated as:

$$\mathcal{L} = \underline{\mathcal{L}^l} + \underline{\lambda_1 \mathcal{L}^u_{knw} + \lambda_2 \mathcal{L}^u_{unk}} + \underline{\lambda_3 \mathcal{L}^u_{con}}$$

For unlabeled training data predicted as known classes (*i.e.*, $\hat{y}_i \in \mathcal{C}^l$), the loss is defined as

$$\mathcal{L}^u_{knw} = -\frac{1}{n^u_{knw}} \sum_{i=1}^{n^u_{knw}} \sum_{c=1}^{|\mathcal{C}^l|} \mathbb{I}(\hat{y}_i = c) \ln(\hat{\mathbf{p}}^c_{i,strong}) + \frac{1}{|\mathcal{C}^l| - 1} \mathbb{I}(\hat{y}_i \neq c) \ln(1 - \hat{\mathbf{p}}^c_{i,strong})$$

For unlabeled training data predicted as unknown classes (*i.e.*, $\hat{y}_i = unknown$), the loss is defined as

$$\mathcal{L}^u_{unk} = -\frac{1}{n^u_{unk}} \sum_{i=1}^{n^u_{unk}} \sum_{c=1}^{|\mathcal{C}^l|} \ln(1 - \hat{\mathbf{p}}^c_{i,strong})$$

Method

② Improving Target Domain Generalization

The overall loss of the training process is formulated as:

$$\mathcal{L} = \underline{\mathcal{L}^l} + \underline{\lambda_1 \mathcal{L}^u_{knw} + \lambda_2 \mathcal{L}^u_{unk}} + \underbrace{\underline{\lambda_3 \mathcal{L}^u_{con}}}$$

To avoid overfitting on domain-related low-level statistics, we minimize the consistency regularization loss between the original unlabeled samples and the augmented ones to push the model to pay attention to the high-level semantics of the samples, defined as

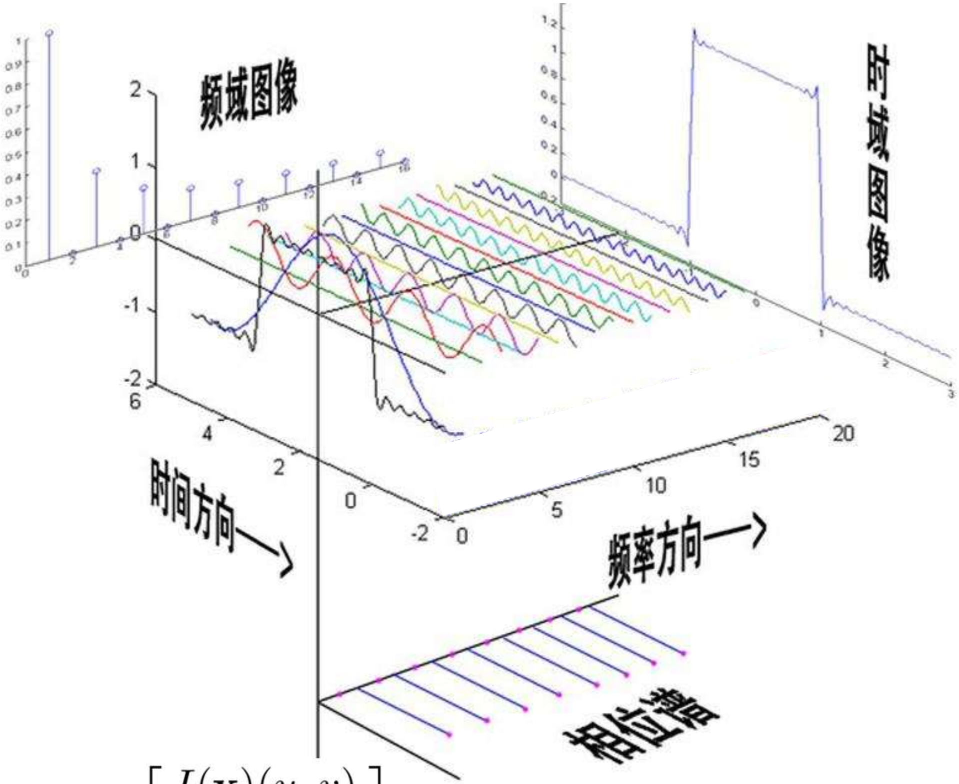
$$\mathcal{L}^u_{con} = \frac{1}{n^u} \sum_{i=1}^{n^u} \sum_{c=1}^{|\mathcal{C}^l|} |\hat{\mathbf{p}}^c_i(\mathbf{x}_i) - \hat{\mathbf{p}}^c_i(\boxed{\tilde{\mathbf{x}}_i})|^2$$

How to make the augmented $\tilde{\mathbf{x}}_i$?

② Improving Target Domain Generalization — Fourier Transformation

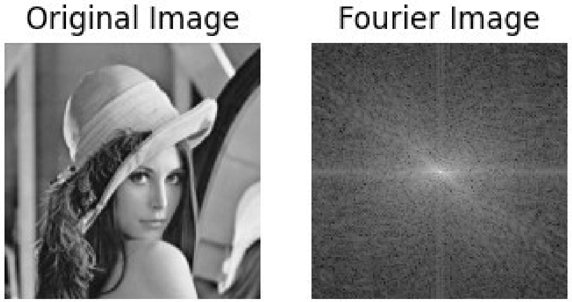
Specifically, for each x, its Fourier Transformation F (x) is formulated as:

$$\mathcal{F}(\mathbf{x})(u,v)=\sum_{h=0}^{H-1}\sum_{w=0}^{W-1}\mathbf{x}(h,w)e^{-J2\pi\left(\frac{h}{H}u+\frac{w}{W}v\right)}$$



The amplitu

then respect


Original Image
Fourier Image

onent P(x) are

$\mathcal{A}(\mathbf{x})(u,v)$

$$,\mathcal{P}(\mathbf{x})(u,v)=\arctan\left[\frac{I(\mathbf{x})(u,v)}{R(\mathbf{x})(u,v)}\right]$$

Method

② Improving Target Domain Generalization — Fourier Transformation

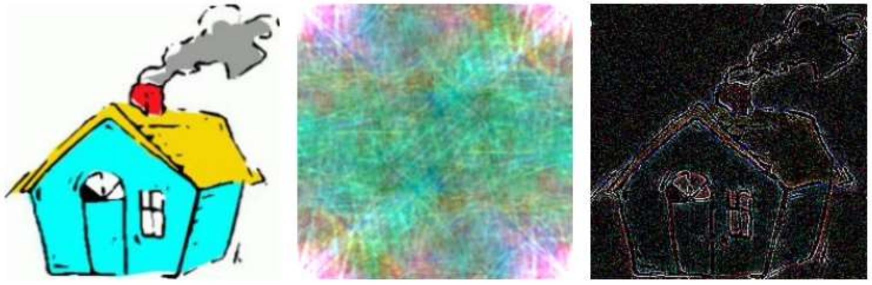
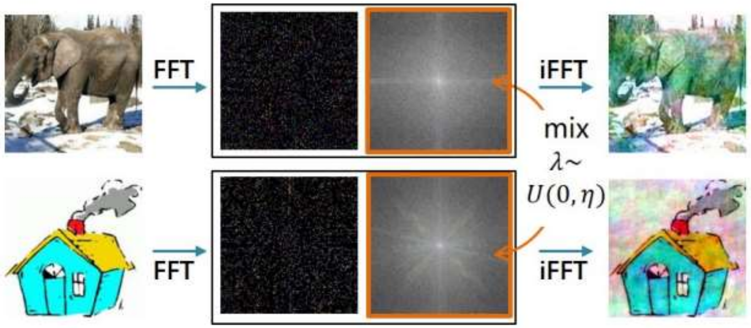


Figure 1. Examples of the amplitude-only and phase-only reconstruction: (a) original image; (b) reconstructed image with amplitude information only by setting the phase component to a constant; (c) reconstructed image with phase information only by setting the amplitude component to a constant.

Amplitude Mix (AM)



(b)

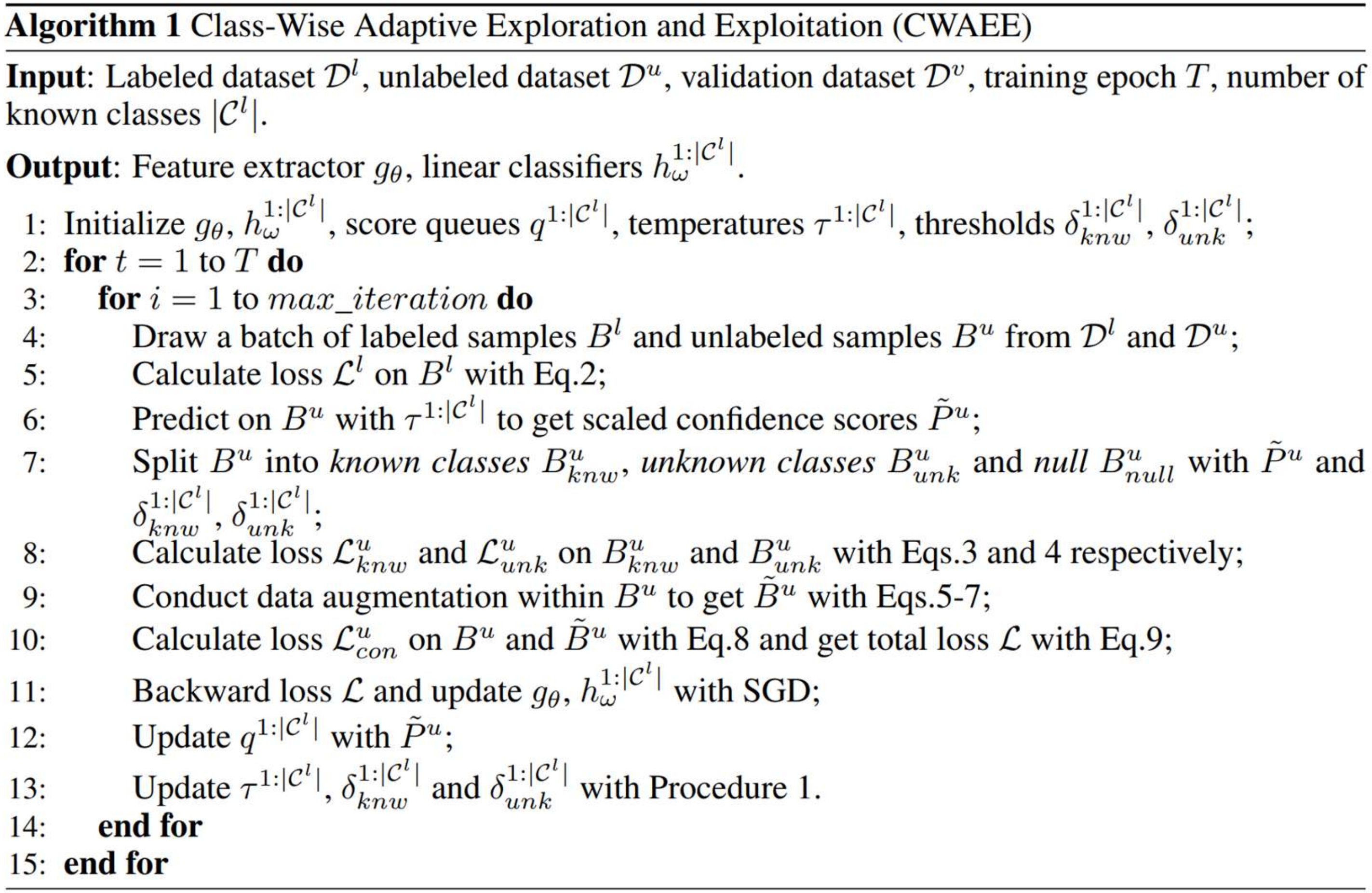
Figure 3. Illustration of (a) AS and (b) AM strategy.

$$\tilde{\mathcal{A}}(\mathbf{x}_i) = (1-\lambda)\mathcal{A}(\mathbf{x}_i) + \lambda\mathcal{A}(\mathbf{x}_{i'}), \tilde{\mathbf{x}}_i = \mathcal{F}^{-1}\left(\tilde{\mathcal{A}}(\mathbf{x}_i) * e^{-J*\mathcal{P}(\mathbf{x}_i)}\right)$$

Xu Q, Zhang R, Zhang Y, et al. A fourier-based framework for domain generalization. CVPR 2021

Method

② Improving Target Domain Generalization



Algorithm 1 Class-Wise Adaptive Exploration and Exploitation (CWAEE)

```
Input: Labeled dataset $\mathcal{D}^l$, unlabeled dataset $\mathcal{D}^u$, validation dataset $\mathcal{D}^v$, training epoch $T$, number of known classes $|\mathcal{C}^l|$.
Output: Feature extractor $g_\theta$, linear classifiers $h_\omega^{1:|\mathcal{C}^l|}$.
Initialize $g_\theta$, $h_\omega^{1:|\mathcal{C}^l|}$, score queues $q^{1:|\mathcal{C}^l|}$, temperatures $\tau^{1:|\mathcal{C}^l|}$, thresholds $\delta_{knw}^{1:|\mathcal{C}^l|}$, $\delta_{unk}^{1:|\mathcal{C}^l|}$;
for $t = 1$ to $T$ do
    for $i = 1$ to $max\_iteration$ do
        Draw a batch of labeled samples $B^l$ and unlabeled samples $B^u$ from $\mathcal{D}^l$ and $\mathcal{D}^u$;
        Calculate loss $\mathcal{L}^l$ on $B^l$ with Eq.2;
        Predict on $B^u$ with $\tau^{1:|\mathcal{C}^l|}$ to get scaled confidence scores $\tilde{P}^u$;
        Split $B^u$ into known classes $B^u_{knw}$, unknown classes $B^u_{unk}$ and null $B^u_{null}$ with $\tilde{P}^u$ and $\delta_{knw}^{1:|\mathcal{C}^l|}$, $\delta_{unk}^{1:|\mathcal{C}^l|}$;
        Calculate loss $\mathcal{L}^u_{knw}$ and $\mathcal{L}^u_{unk}$ on $B^u_{knw}$ and $B^u_{unk}$ with Eqs.3 and 4 respectively;
        Conduct data augmentation within $B^u$ to get $\tilde{B}^u$ with Eqs.5-7;
        Calculate loss $\mathcal{L}^u_{con}$ on $B^u$ and $\tilde{B}^u$ with Eq.8 and get total loss $\mathcal{L}$ with Eq.9;
        Backward loss $\mathcal{L}$ and update $g_\theta$, $h_\omega^{1:|\mathcal{C}^l|}$ with SGD;
        Update $q^{1:|\mathcal{C}^l|}$ with $\tilde{P}^u$;
        Update $\tau^{1:|\mathcal{C}^l|}$, $\delta_{knw}^{1:|\mathcal{C}^l|}$ and $\delta_{unk}^{1:|\mathcal{C}^l|}$ with Procedure 1.
    end for
end for
```

Experiment

Main Results

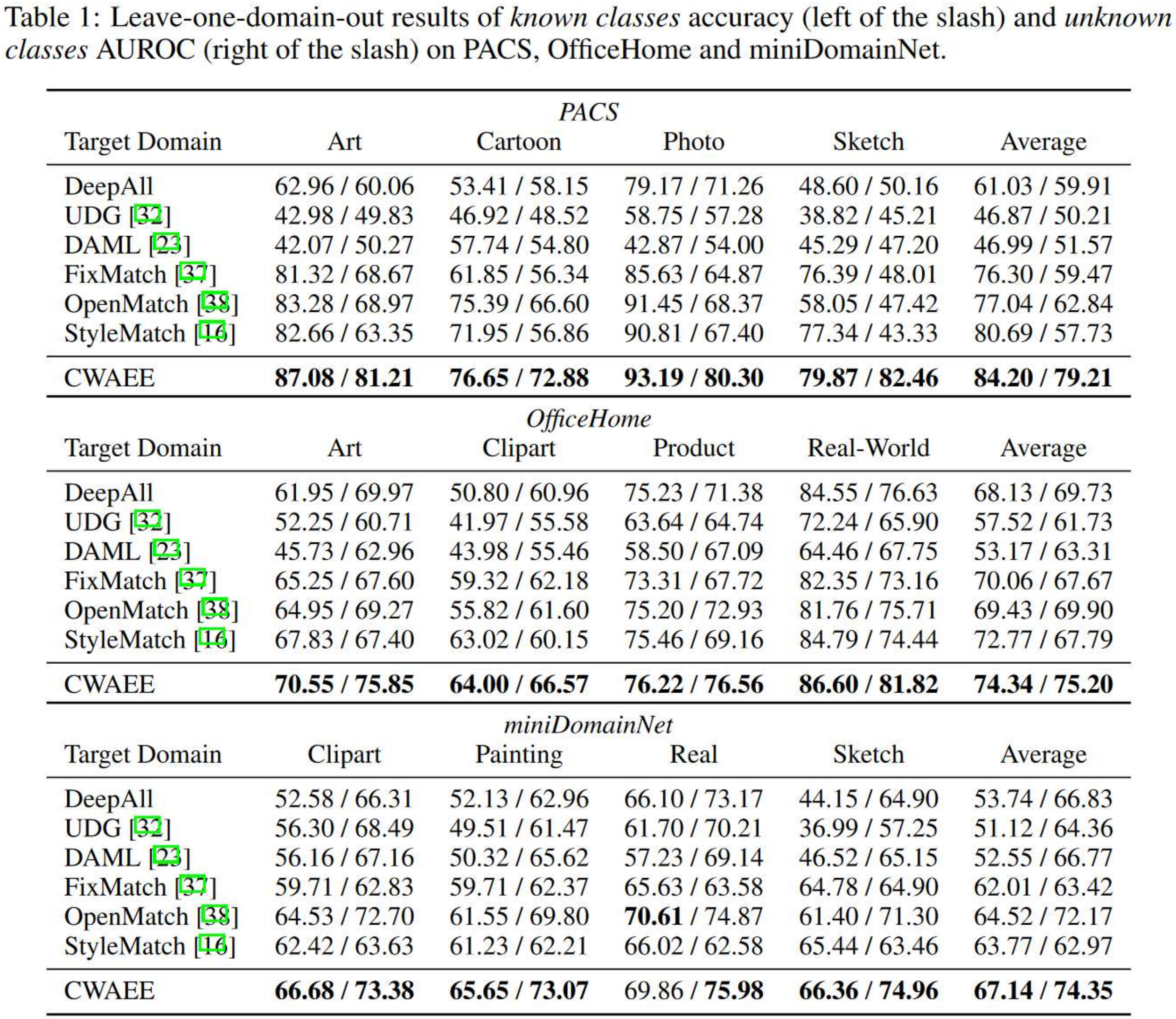


Table 1: Leave-one-domain-out results of *known classes* accuracy (left of the slash) and *unknown classes* AUROC (right of the slash) on PACS, OfficeHome and miniDomainNet.

PACS

Target Domain	Art	Cartoon	Photo	Sketch	Average
DeepAll	62.96 / 60.06	53.41 / 58.15	79.17 / 71.26	48.60 / 50.16	61.03 / 59.91
UDG [32]	42.98 / 49.83	46.92 / 48.52	58.75 / 57.28	38.82 / 45.21	46.87 / 50.21
DAML [23]	42.07 / 50.27	57.74 / 54.80	42.87 / 54.00	45.29 / 47.20	46.99 / 51.57
FixMatch [37]	81.32 / 68.67	61.85 / 56.34	85.63 / 64.87	76.39 / 48.01	76.30 / 59.47
OpenMatch [38]	83.28 / 68.97	75.39 / 66.60	91.45 / 68.37	58.05 / 47.42	77.04 / 62.84
StyleMatch [16]	82.66 / 63.35	71.95 / 56.86	90.81 / 67.40	77.34 / 43.33	80.69 / 57.73
CWAEE	**87.08 / 81.21**	**76.65 / 72.88**	**93.19 / 80.30**	**79.87 / 82.46**	**84.20 / 79.21**

OfficeHome

Target Domain	Art	Clipart	Product	Real-World	Average
DeepAll	61.95 / 69.97	50.80 / 60.96	75.23 / 71.38	84.55 / 76.63	68.13 / 69.73
UDG [32]	52.25 / 60.71	41.97 / 55.58	63.64 / 64.74	72.24 / 65.90	57.52 / 61.73
DAML [23]	45.73 / 62.96	43.98 / 55.46	58.50 / 67.09	64.46 / 67.75	53.17 / 63.31
FixMatch [37]	65.25 / 67.60	59.32 / 62.18	73.31 / 67.72	82.35 / 73.16	70.06 / 67.67
OpenMatch [38]	64.95 / 69.27	55.82 / 61.60	75.20 / 72.93	81.76 / 75.71	69.43 / 69.90
StyleMatch [16]	67.83 / 67.40	63.02 / 60.15	75.46 / 69.16	84.79 / 74.44	72.77 / 67.79
CWAEE	**70.55 / 75.85**	**64.00 / 66.57**	**76.22 / 76.56**	**86.60 / 81.82**	**74.34 / 75.20**

miniDomainNet

Target Domain	Clipart	Painting	Real	Sketch	Average
DeepAll	52.58 / 66.31	52.13 / 62.96	66.10 / 73.17	44.15 / 64.90	53.74 / 66.83
UDG [32]	56.30 / 68.49	49.51 / 61.47	61.70 / 70.21	36.99 / 57.25	51.12 / 64.36
DAML [23]	56.16 / 67.16	50.32 / 65.62	57.23 / 69.14	46.52 / 65.15	52.55 / 66.77
FixMatch [37]	59.71 / 62.83	59.71 / 62.37	65.63 / 63.58	64.78 / 64.90	62.01 / 63.42
OpenMatch [38]	64.53 / 72.70	61.55 / 69.80	**70.61** / 74.87	61.40 / 71.30	64.52 / 72.17
StyleMatch [16]	62.42 / 63.63	61.23 / 62.21	66.02 / 62.58	65.44 / 63.46	63.77 / 62.97
CWAEE	**66.68 / 73.38**	**65.65 / 73.07**	69.86 / **75.98**	**66.36 / 74.96**	**67.14 / 74.35**

Experiment

Ablation Study

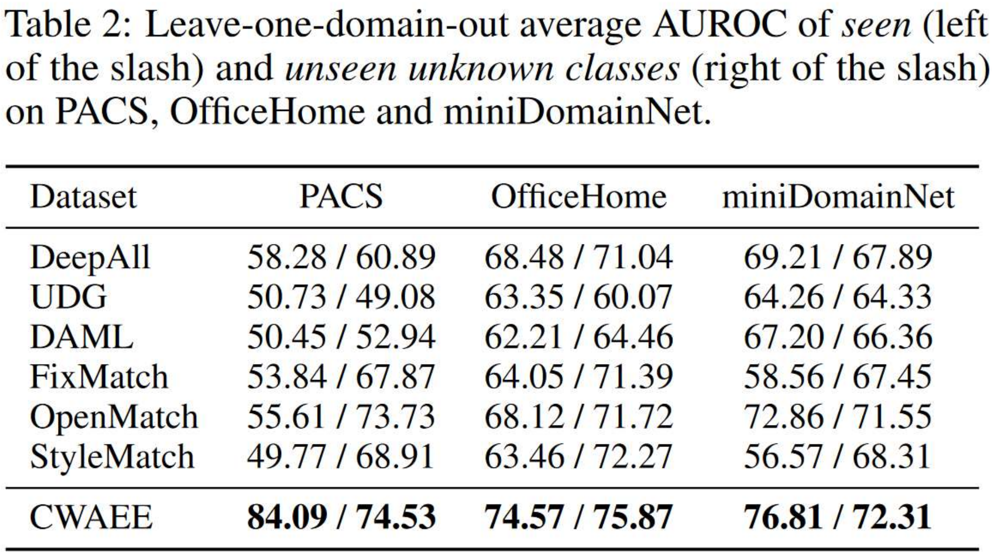


Table 2: Leave-one-domain-out average AUROC of *seen* (left of the slash) and *unseen unknown classes* (right of the slash) on PACS, OfficeHome and miniDomainNet.

Dataset	PACS	OfficeHome	miniDomainNet
DeepAll	58.28 / 60.89	68.48 / 71.04	69.21 / 67.89
UDG	50.73 / 49.08	63.35 / 60.07	64.26 / 64.33
DAML	50.45 / 52.94	62.21 / 64.46	67.20 / 66.36
FixMatch	53.84 / 67.87	64.05 / 71.39	58.56 / 67.45
OpenMatch	55.61 / 73.73	68.12 / 71.72	72.86 / 71.55
StyleMatch	49.77 / 68.91	63.46 / 72.27	56.57 / 68.31
CWAEE	**84.09 / 74.53**	**74.57 / 75.87**	**76.81 / 72.31**

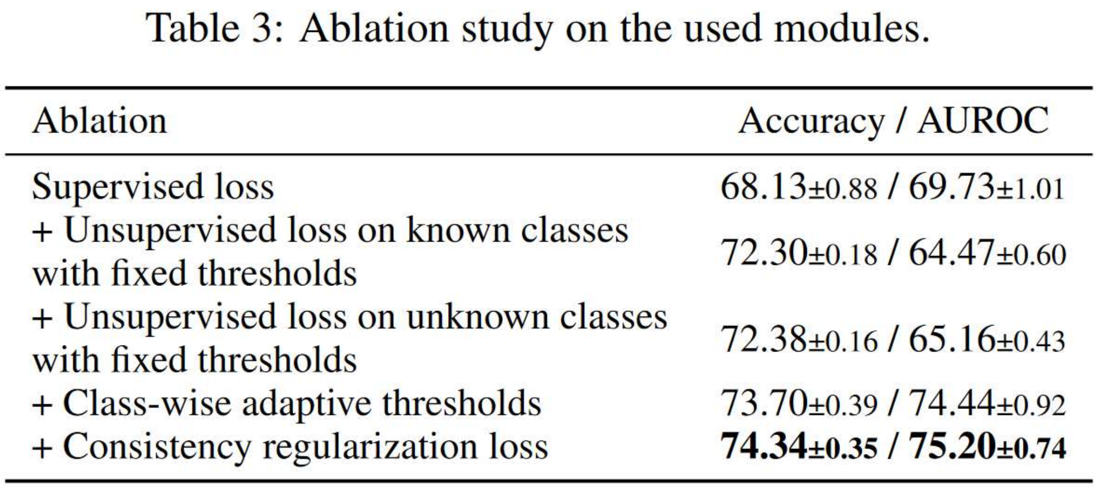


Table 3: Ablation study on the used modules.

Ablation	Accuracy / AUROC
Supervised loss	68.13±0.88 / 69.73±1.01
+ Unsupervised loss on known classes with fixed thresholds	72.30±0.18 / 64.47±0.60
+ Unsupervised loss on unknown classes with fixed thresholds	72.38±0.16 / 65.16±0.43
+ Class-wise adaptive thresholds	73.70±0.39 / 74.44±0.92
+ Consistency regularization loss	**74.34±0.35 / 75.20±0.74**

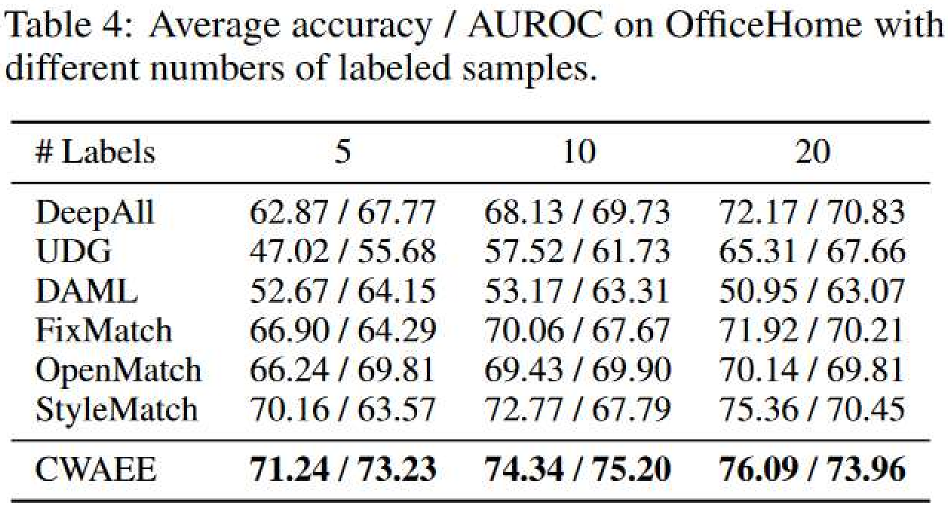
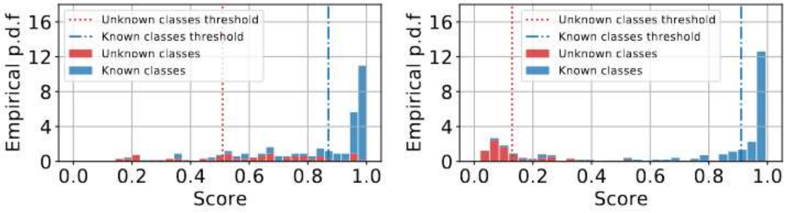


Table 4: Average accuracy / AUROC on OfficeHome with different numbers of labeled samples.

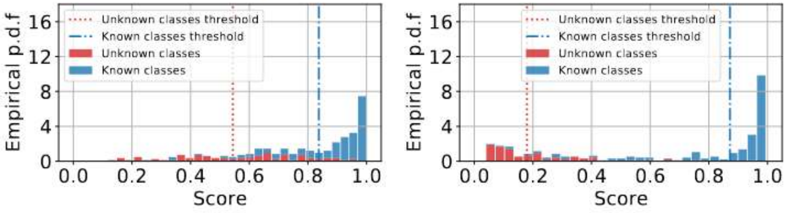
# Labels	5	10	20
DeepAll	62.87 / 67.77	68.13 / 69.73	72.17 / 70.83
UDG	47.02 / 55.68	57.52 / 61.73	65.31 / 67.66
DAML	52.67 / 64.15	53.17 / 63.31	50.95 / 63.07
FixMatch	66.90 / 64.29	70.06 / 67.67	71.92 / 70.21
OpenMatch	66.24 / 69.81	69.43 / 69.90	70.14 / 69.81
StyleMatch	70.16 / 63.57	72.77 / 67.79	75.36 / 70.45
CWAEE	**71.24 / 73.23**	**74.34 / 75.20**	**76.09 / 73.96**

Experiment

Some Figures

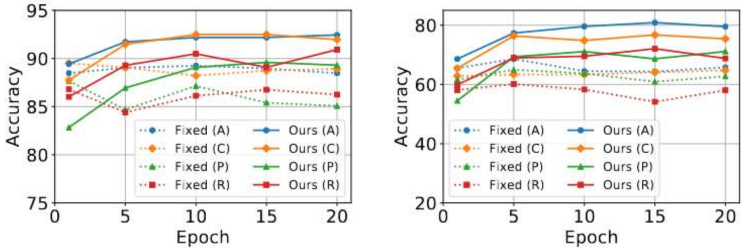


(a) Class 0 in epoch 1 (b) Class 0 in epoch 20



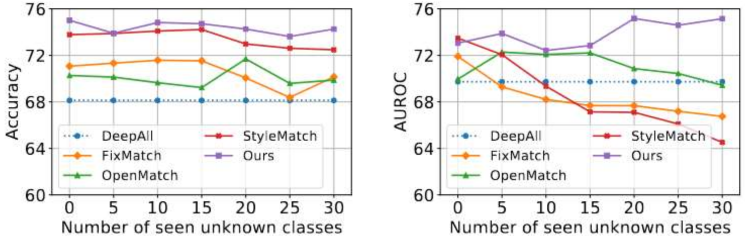
(c) Class 1 in epoch 1 (d) Class 1 in epoch 20

Figure 4: The empirical p.d.f. of the confidence scores of unlabeled data.



(a) Known classes (b) Unknown classes

Figure 2: The accuracy of pseudo-labels on unlabeled samples.



(a) Accuracy (b) AUROC

Figure 3: The results with various numbers of seen unknown classes.

Thanks