Supplementary: Classifier-Oriented Calibration via Textual Prototype for Source-Free Universal Domain Adaptation

Xinghong Liu[1,2,3] and Yi Zhou⋆[1,2]

[1] School of Computer Science and Engineering, Southeast University, Nanjing Jiangsu 211189, China
[2] Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
[3] Postal Savings Bank of China, Beijing 100032, China

In this appendix, we provide more details of our approach, such as additional experiment results, implementation details, and discussion.

This supplementary is organized as follows:

- Notations
- Additional Experiment Results
- Implementation Details
  - Source Model Details
  - Hyperparameters
  - Silhouette Score
  - Pseudo Code
  - Baseline Details
- Discussion
  - K-means Clustering Invocations
  - Limitations
  - Potential Societal Impact

A Notations

We summarize the notations throughout the paper in Table 1. The notations are listed under five groups: models, spaces, variables, measures, and hyperparameters.

B Additional Experiment Results

In this section, the source model is cross-modal linear probing [7] with 16-shot source samples. Unless otherwise mentioned, the source model is based on CLIP(ViT-B/16) [14].

Impact of Varying $|\mathcal{C}|$. We evaluate the robustness of COCA by contrasting it with other methods under varying numbers of common classes $|\mathcal{C}|$ on Office-Home in OPDA. Fig. 1a and Fig. 1b illustrate that COCA overall outperforms and demonstrates greater stability than preceding models.

Table 1: **Notation Table**

	Symbol	Description
Models	G^{img}	Image encoder
	G^{text}	Text encoder
	ω	Parameters of the image/text encoders
	h_θ	Closed-set classifier
	h_γ^{EMA}	EMA teacher classifier
Spaces	$\mathcal{C}^s$	Source/Known class set
	$\mathcal{C}^t$	Target class set
	$\mathcal{C}$	Common class set
	$\overline{\mathcal{C}^s}$	Source-private class set
	$\overline{\mathcal{C}^t}$	Target-private/unknown class set
	$\mathcal{X}$	Target image set
	$\mathcal{Z}^{\text{img}}$	Image feature set
	$\mathcal{Z}^{\text{text}}$	Text feature set
	$\mathcal{V}^{\text{img}}$	Image prototype set
Variables	x_i	Unlabeled target image
	x_i^M	Unlabeled masked target image
	`a photo of a {CLS}`	Text template
	y_c	Ground truth label for `a photo of a {CLS}`
	$\hat{y}_i$	Pseudo label for target image x_i
	z_i^{img}	Target domain image feature
	z_c^{text}	Text feature
	$\{v_k^{\text{img}}\}_{k=1}^K$	Image prototype generated by K-means
	p^c	Image positive prototype for a known class c
	$\{n_k^c\}_{k=1}^{K-1}$	Image negative prototypes for a known class c
	$p(y\|x_i;\gamma)$	Soft label generated by the teacher classifier h_γ^{EMA}
Measures	R_{IB}	Information Bottleneck
	I	Mutual Information
	$U(x_i)$	Uncertainty for target image x_i
Hyperparameters	K	K-means hyperparameter
	τ	Threshold for distinguishing known and unknown images
	r	Mask ratio

Ablation Study. We conducted comprehensive ablation studies on the three datasets to assess the effectiveness of distinct components within our method. The results are summarized in Table 2, where $OS = \frac{|\mathcal{C}^s|}{|\mathcal{C}^s|+1} \times OS^* + \frac{|\mathcal{C}^s|}{|\mathcal{C}^s|+1} \times UNK$ indicates the average accuracy on different classes. Compared to **COCA-w-**p^c, **COCA** shows 3.1% improvement in HOS for OPDA on OfficeHome, 3.4% on VisDA, and 1.3% on DomainNet. It indicates that textual prototypes z_c^{img} are more appropriate than image prototypes p^c for positive prototypes due to $R_{\text{IB}}(\mathcal{Z}^{\text{text}}) > R_{\text{IB}}(\mathcal{V}^{\text{img}})$, as discussed in our paper (Eq. (5)). **COCA-w/o-**h_θ represents the combination of the ACTP module and the zero-shot CLIP without the linear classifier h_θ. The HOS results of **COCA-w/o-**h_θ highlight the potential of integrating image and text encoders within VLMs. This integration enables the precise separation of common and unknown class samples. However,

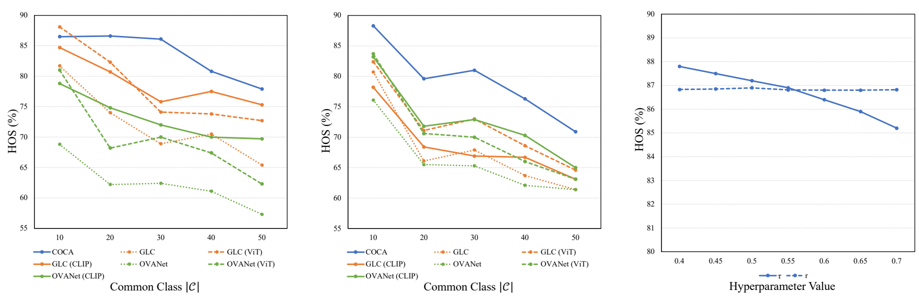
⋆ Corresponding author

Table 2: OS and HOS (%) of variants of COCA in **OPDA**.

	OfficeHome		VisDA-2017		DomainNet	
	OS	HOS	OS	HOS	OS	HOS
COCA-w/o-h_θ	88.8	83.7	83.0	77.7	50.3	63.8
COCA-w/o-MIECI	89.0	86.6	83.6	82.2	65.7	72.9
COCA-w-p^c	81.0	83.8	74.7	79.8	66.2	71.8
COCA	**90.2**	**86.9**	**85.2**	**83.2**	**66.4**	**73.1**

Table 3: OS and HOS (%) of COCA with CLIP(RN50x16) in **OPDA**.

	OS	HOS
OfficeHome	85.9	84.0
VisDA-2017	71.2	76.2
DomainNet	60.2	69.0



(a) Cl→Pr in OPDA (b) Rw→Ar in OPDA (c) HOS w.r.t. τ and r

Fig. 1: (a-b)HOS (%) with respect to the number of common class $|\mathcal{C}|$ on Office-Home in OPDA. (c) HOS (%) with respect to τ and r on OfficeHome in OPDA.

a considerable performance gap remains when compared to the full **COCA** method. The **COCA** method demonstrates significant improvements, achieving a 3.2% increase in HOS for OPDA on OfficeHome, a 5.5% improvement on VisDA, and a remarkable 9.3% enhancement on DomainNet. The result gaps of **COCA** and **COCA-w/o-h_θ** show our method's effectiveness. The innovative paradigm we propose in our paper (Fig. 2), emphasizing classifier optimization rather than the image encoder optimization seen in previous UniDA/SF-UniDA methods, presents a more fitting approach based on VLMs to tackle SF-UniDA challenges as we discussed in our paper (Sec. 3.3). **COCA-w/o-MIECI** indicates the removal of the MIECI module. A comparative analysis of results between **COCA-w/o-MIECI** and **COCA** reveals that the MIECI module plays a crucial role in promoting the learning of context relations within target images. This results in an increase in mutual information $I(\mathcal{Z}^{\text{img}}, \mathcal{Y}; \theta)$. As discussed in our paper (Sec. 3.2), this improvement directly contributes to enhanced model performance, specifically in terms of accuracy in classifying common class samples. To visually assess the separation between the common and unknown classes on VisDA-2017 in OSDA, we present the uncertainty density distribution in Fig. 2. The level of uncertainty indicates the extent to which the model regards the input image as belonging to an unknown class. The results demonstrate that while the source model performs well in classifying common classes,

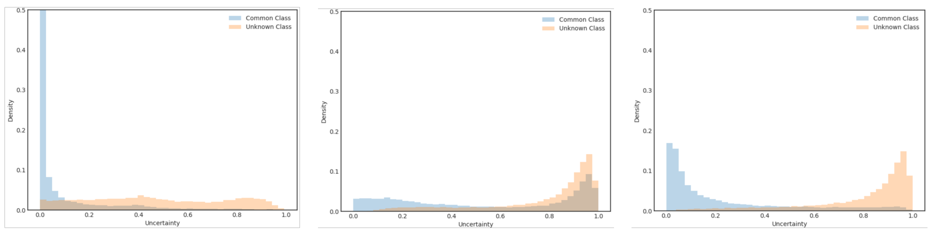
Table 4: HOS (%) with respect to prompts and K selecting methods in **OPDA**.

(a) HOS (%) comparison of various prompts.

Prompt	OfficeHome	VisDA	DomainNet
`a photo of a {CLS}`	86.9	83.2	73.1
`a photo of some {CLS}`	87.4	82.6	72.9
`a picture of a {CLS}`	88.3	82.1	72.1
`a painting of a {CLS}`	86.5	82.2	73.0
`this is a photo of a {CLS}`	87.4	84.0	72.1
`this is a {CLS} photo`	86.6	83.3	72.5

(b) HOS (%) comparison of various K selecting methods.

Method	OfficeHome	VisDA	DomainNet
Calinski-Harabasz [1]	86.9	82.7	72.8
Davies-Bouldi [2]	86.9	83.2	73.0
Silhouette [15]	86.9	83.2	73.1



(a) Source Model (16-shot) (b) + COCA-w-p^c (c) + COCA

Fig. 2: Uncertainty distribution of the source model [7], the source model + COCA-w-p^c, and the source model + COCA for common and unknown class images on VisDA-2017 in OSDA.

it struggles with the separation of unknown classes. In contrast, **COCA-w-**p^c exhibits imprecise recognition of common classes. Notably, **COCA** achieves a better balance between common class classification and unknown class identification, highlighting the superiority of textual prototypes. The results of the source model [7] using the EfficientNet-style [17] CLIP model, *i.e.*, CLIP(RN50x16), is presented in Table 3. These results demonstrate that our approach is adaptable to various image encoder frameworks, including CNNs. The result gaps exist between **COCA-w-CLIP(ViT-B/16)** and **COCA-w-CLIP(RN50x16)**, attributed to (1) the robustness of ViTs to deal with significant distribution shifts, *e.g.*, recognizing object shapes in less textured data such as paintings [9], and (2) significant architectural differences in the image and text encoders of CNN-based CLIP. Since the classifier is initialized based on text features, when the closed-set model utilizes the pseudo label `unknown` for open-set recognition, the architectural differences hinder the classifier from adequately aligning with common class image features. Specifically, we observe that the common class accuracy of the closed-set classifier is susceptible to the pseudo label `unknown` when the source model is based on CLIP(RN50x16). We deduce that **COCA** has a stronger affinity with ViT architecture CLIP.

Hyperparameter Sensitivity. Fig. 1c demonstrates the sensitivity to the hyperparameter τ and mask ratio r in OPDA on OfficeHome. The source model [7] + COCA is stable across a range of values for both τ and r. The compar-

Table 5: Optimal $K \in [1/3|\mathcal{C}^s|, 1/2|\mathcal{C}^s|, |\mathcal{C}^s|, 2|\mathcal{C}^s|, 3|\mathcal{C}^s|]$ selected by various methods in **OPDA**.

Method	OfficeHome ($\lvert\mathcal{C}^s\rvert = 15, \lvert\mathcal{C}^t\rvert = 60$)				VisDA-2017	DomainNet ($\lvert\mathcal{C}^s\rvert = 200, \lvert\mathcal{C}^t\rvert = 295$)		
	→Ar	→Cl	→Pr	→Rw	($\lvert\mathcal{C}^s\rvert = 9, \lvert\mathcal{C}^t\rvert = 9$)	→P	→R	→S
Calinski-Harabasz [1]	45	45	45	45	27	600	600	600
Davies-Bouldi [2]	45	30	45	45	9	200	200	200
Silhouette [15]	45	45	45	45	9	200	400	400

Table 6: Batch size for source model training.

	batch size
$8 \leq 2\lvert\mathcal{C}^s\rvert < 16$	8
$16 \leq 2\lvert\mathcal{C}^s\rvert < 32$	16
$32 \leq 2\lvert\mathcal{C}^s\rvert < 64$	32
$64 \leq 2\lvert\mathcal{C}^s\rvert$	64

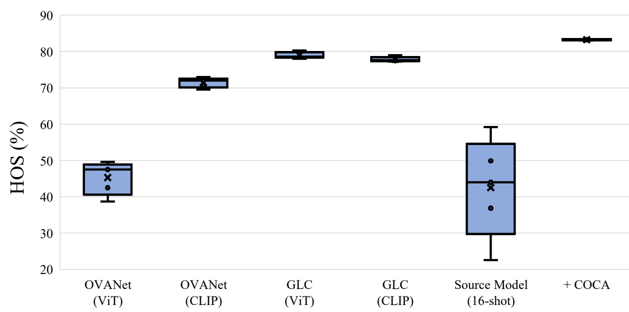


Fig. 3: HOS rate of OVANet (ViT/CLIP) [16], GLC (ViT/CLIP) [13], the source model [7], and the source model + COCA on VisDA-2017 in **OPDA**. Each boxplot ranges from the upper and lower quartiles with the median as the horizontal line and whiskers extend to 1.5 times the interquartile range.

ative experiments of prompts are shown in Table 4a, and our method exhibits stable performance across a variety of prompts. We substitute the Silhouette method [15] with alternative methods, including the Calinski-Harabasz method [1] and the Davies-Bouldin method [2], to ascertain the optimal K value for the K-means clustering. This adjustment aims to evaluate COCA's generalization capabilities, with the results presented in Table 4b. Comparing the results of Silhouettes, Calinski-Harabasz, and Davies-Bouldin methods, we deduce that COCA exhibits good generalization capabilities. This conclusion arises from the stable performance of COCA in OPDA across various methods used to determine the optimal K value for the K-means clustering. The optimal K value at the target domain adaptation phase selected by various methods [1,2,15] in OPDA is presented in Table 5.

Boxplots. An illustration of boxplots with 5 different random seeds in Fig. 3 demonstrates that COCA achieves more accurate performance in separating common and unknown classes than existing methods.

C Implementation Details

C.1 Source Model Details

The source model is linear probe CLIP [14], CLIP-Adapter [4], or cross-modal linear probing [7] based on CLIP(ViT-B/16). At the source model training phase, we freeze the image and text encoders and optimize the classifier. The classifier of linear probe CLIP [14] or cross-model linear probing [7] is the single linear layer. The classifier of CLIP-Adapter [4] is the adapter module. The basic settings are as follows: (1) Performing a learning rate warmup with 50 iterations, during which the learning rate goes up linearly from 0.00001 to the initial value. (2) Performing a cosine annealing learning rate scheduling over the course of 12800 iterations. (3) Employing early stopping based on the few-shot validation set performance evaluated every 100 iterations.

We have established the batch size for source model training as outlined in Table 6. We configure the weight decay to 0.01 for all benchmarks and the initial learning rate to 0.001 for OfficeHome [19] and VisDA-2017 [12], and 0.0001 for DomainNet [11]. The optimizer of the source model is AdamW [8]. Given that the cross-modal linear probing model [7] necessitates the inclusion of varied class names within a mini-batch for training, and that the input is comprised of a 50-50 split between images and text. For the CLIP-Adapter model [4], we follow the same 2-layer MLP architecture with the given residual ratio of 0.2.

Image Loss. Given a image feature $z_i^{\text{img,s}}$ for the source image x_i^s and the corresponding ground truth label y_i^s, the image loss ℓ_s^{img} for the source model training is $\ell_s^{\text{img}} = -\frac{1}{N^s}\sum_{i=1}^{N^s} y_i^s \log(\sigma(h_\theta^s(z_i^{\text{img,s}})))$, where N^s is the number of source samples.

Text Loss. Given a text feature z_c^{text} converted from text template `a photo of a {CLS}`, the corresponding ground truth label y_c, the text loss ℓ_s^{text} for the cross-modal linear probing model is $\ell_s^{\text{text}} = -\frac{1}{|\mathcal{C}^s|}\sum_{c=1}^{|\mathcal{C}^s|} y_c \log\left(\sigma\left(h_\theta^s\left(z_c^{\text{text}}\right)\right)\right)$.

Overall Loss. For linear probe CLIP and CLIP-Adapter models, the overall loss ℓ_s for source model training is $\ell_s = \ell_s^{\text{img}}$. For the cross-modal linear probing model, the overall loss ℓ_s for source model training is $\ell_s = \ell_s^{\text{img}} + \ell_s^{\text{text}}$.

C.2 Hyperparameters

The source model is based on CLIP(ViT-B/16) or CLIP(RN50x16). At the target domain adaptation phase, we applied the AdamW [8] optimizer, configured with beta values of (0.9, 0.999), an epsilon of 1e-08, and a weight decay of 0.01. The batch size is 64 for all benchmarks. The learning rate was adjusted according to the sample number of target domains, resulting in rates of 1e-3 for OfficeHome, 1e-4 for VisDA-2017, and 1e-5 for DomainNet. The MIECI module utilizes the following parameters: mask ratio $r = 0.5$ for CLIP(ViT-B/16) and $r = 0.01$ for CLIP(RN50x16); patch size $w = 16$ for CLIP(ViT-B/16) and $w = 4$ for CLIP(RN50x16); smooth factor $\alpha = 0.999$ as suggested by [18]; and color augmentation parameters as recommended in [5]. All of our experiments are conducted using an RTX-4090 GPU and PyTorch-2.0.1.

C.3 Silhouette Score

For a image feature $z_i^{\text{img}} \in \mathcal{C}_k$, where $\mathcal{C}_k$ is one of the K clusters, computing the mean distance between z_i^{img} and other image features z_j^{img} within the same cluster as follows:

$$a(z_i^{\mathrm{img}}) = \frac{1}{|\mathcal{C}_k| - 1} \sum_{z_j^{\mathrm{img}} \in \mathcal{C}_k, i \neq j} d(z_i^{\mathrm{img}}, z_j^{\mathrm{img}}), \tag{1}$$

where $|\mathcal{C}_k|$ denotes the number of image features belonging to cluster $\mathcal{C}_k$, and $d(z_i^{\mathrm{img}}, z_j^{\mathrm{img}})$ is the distance between z_i^{img} and z_j^{img} within the cluster $\mathcal{C}_k$.

$$b(z_i^{\mathrm{img}}) = \min_{l \neq k} \frac{1}{|\mathcal{C}_l|} \sum_{z_j^{\mathrm{img}} \in \mathcal{C}_l} d(z_i^{\mathrm{img}}, z_j^{\mathrm{img}}) \tag{2}$$

is the distance between z_i^{img} and the "neighboring cluster" of z_i^{img}. The mean distance from z_i^{img} to all image features z_j^{img} in $\mathcal{C}_l$ is calculated as the dissimilarity of z_i^{img} to another cluster $\mathcal{C}_l$, where $\mathcal{C}_l \neq \mathcal{C}_k$. The Silhouette score $s(z_i^{\mathrm{img}})$ is defined as:

$$s(z_i^{\mathrm{img}}) = \frac{b(z_i^{\mathrm{img}}) - a(z_i^{\mathrm{img}})}{\max\{a(z_i^{\mathrm{img}}), b(z_i^{\mathrm{img}})\}}. \tag{3}$$

High Silhouette scores for the majority of image features suggest that the K-means hyperparameter K value is well-chosen, indicating that image features within the same cluster are closely grouped and well-separated from those in other clusters.

C.4 Pseudo Code

The training procedure of the proposed method is summarized in Algorithm 1.

C.5 Baseline Details

We have reproduced several open-source UniDA/SF-UniDA models, and the details of the parameters are provided below:

DCC. We use CLIP(ViT-B/16) [14] as the backbone. The classifier is made up of two FC layers. We use Nesterov momentum SGD to optimize the model, which has a momentum of 0.9 and a weight decay of 5e-4. The learning rate decreases by a factor of $(1+\alpha\frac{i}{N})^{-\beta}$, where i and N represent current and global iteration, respectively, and we set $\alpha = 10$ and $\beta = 0.75$. We use a batch size of 36, and the initial learning rate is set as 1e-4 for VisDA-2017, and 1e-3 for OfficeHome and DomainNet. We use the settings detailed in the original paper [6]. PyTorch [10] is used for implementation.

OVANet. For OVANet [16] with ViT-B/16 [3] and CLIP(ViT-B/16) backbones, we adopt the hyperparameter settings outlined in the original paper [16]. Specifically, we utilize inverse learning rate decay scheduling for the learning rate schedule and assign a weight of $\lambda = 0.1$ for the entropy minimization loss across all benchmarks. The batch size is fixed at 36, with the initial learning rate set to 0.01 for the classification layer and 0.001 for the backbone layers. PyTorch [10] is used for the implementation.

Algorithm 1 Traning Producedure of the Proposed Method

Require: Target domain dataset $\mathcal{D}^t = \{x_i\}_{i=1}^N$, prompt `a photo of a {CLS}`, image encoder G^{img}, text encoder G^{text}, source model's classifier h_θ^s, K candidate list $[1/3|\mathcal{C}^s|, 1/2|\mathcal{C}^s|, |\mathcal{C}^s|, 2|\mathcal{C}^s|, 3|\mathcal{C}^s|]$, and other necessary hyperparameters
Ensure: Target domain classifier h_θ
Freeze G^{img} and G^{text}
Input $\mathcal{D}^t$ to G^{img} to generate target-image features $\mathcal{Z}^{\text{img}} = \left\{z_i^{\text{img}}\right\}_{i=1}^N$
$bestK \leftarrow 0$, $maxScore \leftarrow 0$
for $candidateK \in [1/3|\mathcal{C}^s|, 1/2|\mathcal{C}^s|, |\mathcal{C}^s|, 2|\mathcal{C}^s|, 3|\mathcal{C}^s|]$ **do**
    $K \leftarrow candidateK$ ▷ K-means hyperparameter
    Input $\mathcal{Z}^{\text{img}}$ to K-means to cluster all target image features
    Calculate target image features' Silhouette score $s\left(z^{\text{img}}\right)$
    Take an average score $\overline{s} = \frac{1}{N}\sum_{i=1}^N s\left(z_i^{\text{img}}\right)$
    if $\overline{s} > maxScore$ **then**
        $bestK \leftarrow candidateK$, $maxScore \leftarrow \overline{s}$
    end if
end for
$h_\theta \leftarrow h_\theta^s$, $h_\gamma^{\text{EMA}} \leftarrow h_\theta^s$ ▷ Initialize the classifiers h_θ and h_γ^{EMA}
Input prompts to G^{text} to generate text features $Z^{\text{text}} = \left\{z_c^{\text{text}}\right\}_{c=1}^{|\mathcal{C}^s|}$
$K \leftarrow bestK$ ▷ K-means hyperparameter
Input $\mathcal{Z}^{\text{img}}$ to K-means to generate image prototypes $\{v_k^{\text{img}}\}_{k=1}^K$
Determine negative image prototypes $\{n_k^c\}_{k=1}^{K-1}$ for known class c
Generate a pseudo label $\hat{y}_i$ for each target image x_i
for $epoch = 1$ **to** $maxEpoch$ **do**
    Calculate the image cross-entropy loss ℓ^{img}
    Calculate the text cross-entropy loss ℓ^{text}
    Generate patch mask M and masked target image x_i^M
    Calculate the mask loss ℓ^{mask}
    $\theta \leftarrow \theta - \nabla_\theta(\ell^{\text{img}} + \ell^{\text{text}} + \ell^{\text{mask}})$ ▷ Update h_θ
    $\gamma \leftarrow \alpha\gamma + (1-\alpha)\theta$ ▷ Update the teacher classifier h_γ^{EMA}
end for

GLC. For GLC [13] with ViT-B/16 and CLIP(ViT-B/16), we employ the SGD optimizer with a momentum of 0.9 at the target model adaptation phase. The initial learning rate is set to 0.001 for OfficeHome and 0.0001 for both VisDA-2017 and DomainNet. The hyperparameter ρ is fixed at 0.75 and $|L|$ at 4 across all datasets, while η is set to 0.3 for VisDA and 1.5 for OfficeHome and DomainNet. All these hyperparameters correspond to the settings detailed in the original paper [13]. PyTorch is used for the implementation.

D Discussion

D.1 K-means Clustering Invocations

In this subsection, we will discuss the frequency of K-means clustering invocations per epoch in OPDA with that of GLC [13].

Table 7: The number of calls of K-means clustering in OPDA. 100 clustering iterations per call.

	OfficeHome	VisDA-2017	DomainNet
GLC	15 $\times maxEpoch$	9 $\times maxEpoch$	200 $\times maxEpoch$
Ours	1	1	1

As shown in Table 7, compared to GLC [13], our method significantly reduces the times of K-means clustering. Our method merely needs to cluster all image features once, and then it can identify the negative prototypes for all known classes. In contrast, the GLC model must apply the K-means cluster $|\mathcal{C}^s|$ times per epoch to locate negative prototypes for all known classes. This suggests that our methods can save significant time on large-scale datasets, particularly when $|\mathcal{C}^s|$ is large.

GLC employs the Top-K method to obtain positive image features for a known class c. The hyperparameter of Top-K is represented as K' to differentiate it from the K-means hyperparameter K. After implementing Top-K for each known class, GLC obtains a positive image feature set $\{z_{c,i}^{\text{img, pos}}\}_{i=1}^{K'}$, where $z_{c,i}^{\text{img, pos}}$ symbolizes the positive image feature for a known class c and a negative image feature set $\{z_{c,j}^{\text{img, neg}}\}_{j=1}^{N-K'} = \{z_l^{\text{img}}\}_{l=1}^{N} / \{z_{c,i}^{\text{img, pos}}\}_{i=1}^{K'}$, where $z_{c,j}^{\text{img, neg}}$ signifies the negative image feature and $\{z_l^{\text{img}}\}_{l=1}^{N}$ represents the target image feature set, with N being the number of target samples. As the positive image feature set varies for each known class c, so too does the negative image feature set for each respective class. Thus, GLC needs to invoke the K-means clustering $|\mathcal{C}^s|$ times to obtain the negative image prototype sets $\{\{n_m^c\}_{m=1}^{K-1}\}_{c=1}^{|\mathcal{C}^s|}$ for all known classes. For instance, consider six image features $\{z_l^{\text{img}}\}_{l=1}^{6}$, two known classes $\{c_1, c_2\}$ and the unknown class `unknown`, where $\{z_1^{\text{img}}, z_2^{\text{img}}\}$ belong to c_1, $\{z_3^{\text{img}}, z_4^{\text{img}}\}$ to c_2, and $\{z_5^{\text{img}}, z_6^{\text{img}}\}$ to `unknown`. GLC uses Top-K ($K' = 2$) to select the positive image features $\{z_{c_1,i}^{\text{img,pos}}\}_{i=1}^{2} = \{z_1^{\text{img}}, z_2^{\text{img}}\}$ for the class c_1 and $\{z_{c_2,i}^{\text{img,pos}}\}_{i=1}^{2} = \{z_3^{\text{img}}, z_4^{\text{img}}\}$ for the class c_2; the negative image features for c_1 are $\{z_{c_1,j}^{\text{img,neg}}\}_{j=1}^{4} = \{z_3^{\text{img}}, z_4^{\text{img}}, z_5^{\text{img}}, z_6^{\text{img}}\}$ and for c_2 are $\{z_{c_2,j}^{\text{img,neg}}\}_{j=1}^{4} = \{z_1^{\text{img}}, z_2^{\text{img}}, z_5^{\text{img}}, z_6^{\text{img}}\}$. Given that the negative image feature sets $\{\{z_{c,j}^{\text{img, neg}}\}_{j=1}^{4}\}_{c=c_1}^{c_2}$ varies for the known classes c_1, c_2, GLC requires to invoke K-means clustering $|\mathcal{C}^s| = 2$ times to generate the negative image prototype sets $\{n_m^{c_1}\}_{m=1}^{K-1} = \text{K-means}\left(\{z_{c_1,j}^{\text{img, neg}}\}_{j=1}^{4}\right)$ and $\{n_m^{c_2}\}_{m=1}^{K-1} = \text{K-means}\left(\{z_{c_2,j}^{\text{img, neg}}\}_{j=1}^{4}\right)$ for all known classes. Furthermore, in GLC, since both the image encoder and the bottleneck layer—situated between the image encoder and the classifier for local census clustering—require updates at each epoch, the K-means clustering must be invoked at each epoch.

On the other hand, in our approach, the target image feature set $\{z_l^{\text{img}}\}_{l=1}^{N}$ remains constant. We first apply K-means to $\{z_l^{\text{img}}\}_{l=1}^{N}$ to derive all image prototypes $\{v_k^{\text{img}}\}_{k=1}^K = \text{K-means}\left(\{z_l^{\text{img}}\}_{l=1}^N\right)$. Subsequently, we perform matrix multiplication between the text feature z_c^{text} of known class c and the image prototype set $\{v_k^{\text{img}}\}_{k=1}^K$ to identify positive and negative image prototypes. As matrix multiplication is considerably more efficient than K-means, our approach significantly reduces computational time in comparison to GLC. In our method, we only need to invoke the K-means clustering at the first epoch since the image and text encoders are frozen.

D.2 Limitations

The proposed approach may be unsuitable for small DA datasets since they cannot provide enough negative images to adapt the classifier. Furthermore, we observe that the quality of pseudo labels affects the model performance. In cases where the dataset does not consist of natural image datasets, *e.g.*, medical images, vision-language models pre-trained on large-scaled natural datasets such as CLIP may not yield high-quality pseudo labels, thereby failing to guide the classifier adaptation accurately.

D.3 Potential Societal Impact

Our method can adapt a trained few-shot learner to unlabeled target datasets with uncertainty domain and category shifts by optimizing the classifier. In numerous instances where source datasets are unobtainable and the quantity of source samples is restricted, our approaches do not need to directly access source samples and substantially reduce the label cost of source samples. This might make technology more accessible to organizations and individuals with limited resources. However, one potential downside is the increased availability of the systems to those seeking to exploit them for unlawful purposes. While we report an enhanced performance in comparison to the current state-of-the-art methods, the results remain unsatisfactory in extreme scenarios of domain shift or category shift. Thus, our approach should not be deployed in critical applications or for making significant decisions without human supervision.

References

1. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics **3**(1), 1–27 (1974)
2. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **1**(2), 224–227 (1979)
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Int. Conf. Learn. Represent. (2021)
4. Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., Qiao, Y.: Clip-adapter: Better vision-language models with feature adapters (2021)

5. Hoyer, L., Dai, D., Wang, H., Van Gool, L.: MIC: Masked image consistency for context-enhanced domain adaptation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023)
6. Li, G., Kang, G., Zhu, Y., Wei, Y., Yang, Y.: Domain consensus clustering for universal domain adaptation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9757–9766 (2021)
7. Lin, Z., Yu, S., Kuang, Z., Pathak, D., Ramanan, D.: Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 19325–19337 (2023)
8. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Int. Conf. Learn. Represent. (2019)
9. Naseer, M., Ranasinghe, K., Khan, S., Hayat, M., Khan, F., Yang, M.H.: Intriguing properties of vision transformers. In: Adv. Neural Inform. Process. Syst. (2021)
10. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Adv. Neural Inform. Process. Syst. (2019)
11. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Int. Conf. Comput. Vis. pp. 1406–1415 (2019)
12. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: Visda: The visual domain adaptation challenge (2017)
13. Qu, S., Zou, T., Röhrbein, F., Lu, C., Chen, G., Tao, D., Jiang, C.: Upcycling models under domain and category shift. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 20019–20028 (2023)
14. 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: Int. Conf. Mach. Learn. pp. 8748–8763 (2021)
15. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987)
16. Saito, K., Saenko, K.: Ovanet: One-vs-all network for universal domain adaptation. In: Int. Conf. Comput. Vis. pp. 9000–9009 (2021)
17. Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: Int. Conf. Mach. Learn. pp. 6105–6114 (2019)
18. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Adv. Neural Inform. Process. Syst. (2017)
19. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5018–5027 (2017)