UOTA: Unsupervised Open-Set Task Adaptation Using a Vision-Language Foundation Model

Youngjo Min [*1] **Kwangrok Ryoo** [*2] **Bumsoo Kim** [2] **Taesup Kim** [1]

Abstract

Human-labeled data is essential for deep learning models, but annotation costs hinder their use in real-world applications. Recently, however, models such as CLIP have shown remarkable zero-shot capabilities through vision-language pre-training. Although fine-tuning with human-labeled data can further improve the performance of zero-shot models, it is often impractical in low-budget real-world scenarios. In this paper, we propose an alternative algorithm, dubbed **U**nsupervised **O**pen-Set **T**ask **A**daptation (UOTA), which fully leverages the large amounts of open-set unlabeled data collected in the wild to improve pre-trained zero-shot models in real-world scenarios. We validate our contributions through extensive experiments on open-set domain adaptation benchmarks applicable to our settings. Despite not using any source domain model or data, our method achieves state-of-the-art performance on the benchmarks.

1. Introduction

Large amounts of human-labeled data are crucial for the performance of deep neural networks. However, collecting such data is costly, posing a challenge for real-world applications. Solutions utilizing unlabeled data (Devlin et al., 2019; Brown et al., 2020; He et al., 2022; Chen et al., 2020; He et al., 2020) have been proposed, but human-labeled data is still required for task-specific learning stages (i.e., task adaptation, fine-tuning, and transfer learning). Recent studies have proposed a new learning paradigm (Radford et al., 2021; Gao et al., 2021; Jia et al., 2021b) that achieves *zero-shot capabilities* by learning transferable representations through vast amounts of image and text pairs, although task-specific human-labeled data is needed to improve downstream performance (Radford et al., 2021; Zhou et al., 2022; Gao et al., 2021). However, to the best of our knowledge, no previous work in the literature has explored real-world scenarios where transfer performance can be enhanced solely by utilizing open-set unlabeled data, including both in-distribution (ID, task-relevant) and out-of-distribution (OOD, task-irrelevant) data.

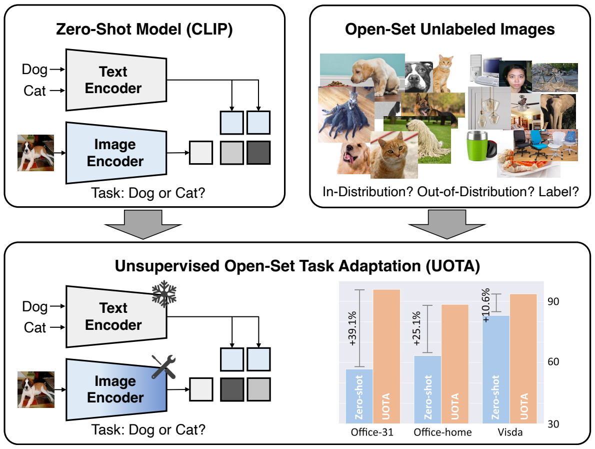


Figure 1. UOTA enhances the transfer performance of the zero-shot model (CLIP) on a downstream target task by leveraging open-set unlabeled data in the wild.

To address this problem, we begin by considering a scenario in real-world situations where only unlabeled data is available from a specific source (e.g., a camera at a specific location). This source will be the target domain of our zero-shot model based on CLIP (Radford et al., 2021) for a given downstream task, and we assume all data from it shares some characteristics, such as style and texture. We then assume the realistic, *open-set* setting (Scheirer et al., 2012; Bendale & Boult, 2016; Kong & Ramanan, 2021; Vaze et al., 2022), which does not impose any constraints on the data, where data can be randomly collected from a particular source and may contain content related to known (i.e., in-distribution; ID) or unknown (i.e., out-of-distribution; OOD) classes.

To improve zero-shot capabilities with these open-set unlabeled data, we propose **U**nsupervised **O**pen-Set **T**ask **A**daptation (UOTA), a simple and practical algorithm that

[*] Equal contribution [1] Seoul National University, Korea [2] LG AI Research, Korea. Correspondence to: Taesup Kim <taesup.kim@snu.ac.kr>.

Work presented at the ES-FoMo Workshop at ICML 2023, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

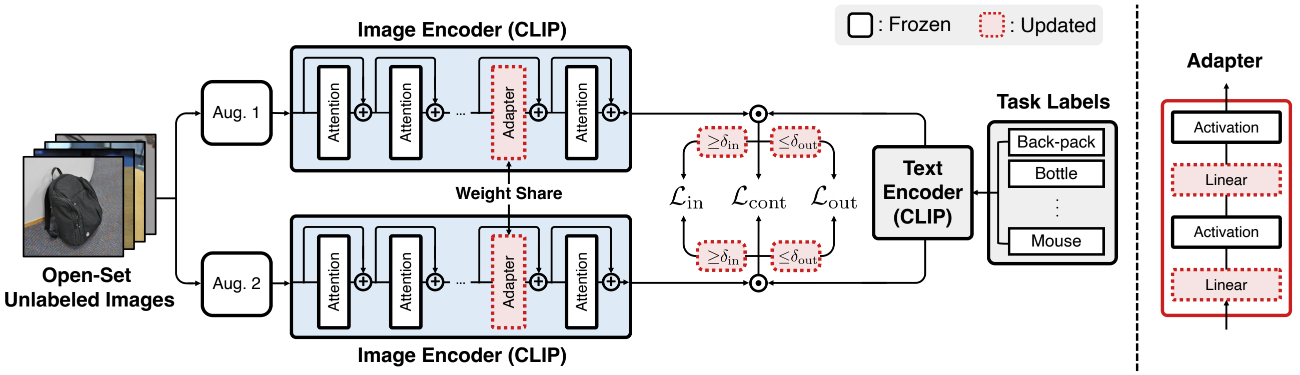


Figure 2. **Left: Implementation details of Unsupervised Open-Set Task Adaptation (UOTA).** For computational efficiency, we only update the parameters of the adapters (Houlsby et al., 2019). **Right: Architecture of the adapter module.** The adapter in our framework comprises two linear layers and two activation layers. During the training process, we update only this lightweight adapter, enabling computationally efficient training.

operates within a unified framework based on the zero-shot model. Our method to improve zero-shot capabilities consists of three objectives: (1) a self-training objective and (2) a negative learning objective (Kim et al., 2019) based on curriculum learning (Soviany et al., 2022; Zhang et al., 2017; Li et al., 2017; Huang et al., 2020; Zhou et al., 2020; Zhang et al., 2021a;b), where class-wise thresholds for detecting unknown class data and classifying known class data are adaptively adjusted according to the training status, and (3) a contrastive objective (Sohn, 2016; van den Oord et al., 2018) to push data with unknown classes away from the space of data with known classes and learn a more discriminative representation space for OOD detection.

As shown in Figure 1, our proposed learning scheme enables the model to implicitly acquire the ability to perform OOD detection during the training process without the need for additional explicit methods to detect OOD samples. It also simultaneously enriches the model's ability to perform precise image classification with known classes. Furthermore, our proposed method is computationally efficient, as it only updates a lightweight adapter inserted in the image encoder while freezing the rest of the model. We validate our contributions by conducting extensive experiments on various open-set domain adaptation (OSDA) benchmarks that are applicable to our settings. Despite not using any source domain model or data, our method achieves state-of-the-art performance on these benchmarks.

2. Method

UOTA fully exploits the pre-trained CLIP model that has a dual-stream architecture with a text encoder $\mathcal{T}_\phi$ and an image encoder $\mathcal{I}_\theta$, where ϕ and θ are the pre-trained parameters. For a given downstream task τ with a class set $Y_\tau = \{y_i\}_{i=1}^{K_\tau}$, where K_τ denotes the number of classes to be classified, we first complete a set of class embeddings $\mathcal{C}_\tau = \{\mathcal{T}_\phi(p_i)\}_{i=1}^{K_\tau}$ by using natural language prompting p_i = "a photo of a {class name of y_i}". When image data x is given, the corresponding embedding $\mathcal{I}_\theta(x)$ is compared with the class embeddings by measuring the cosine similarity, and then we compute the task-wise classification probability as:

$$p(y = y_i|x;\phi,\theta) = \frac{e^{\alpha\cdot D(\mathcal{I}_\theta(x),\mathcal{T}_\phi(p_i))}}{\sum_{j=1}^{N} e^{\alpha\cdot D(\mathcal{I}_\theta(x),\mathcal{T}_\phi(p_j))}}, \quad (1)$$

where α is a learnable scaling factor (i.e., temperature) and $D(\cdot,\cdot)$ denotes cosine similarity between two vectors. The overall architecture is shown in Figure 2.

2.1. Self-training with open-set unlabeled data

Maximum Concept Matching (MCM) (Ming et al., 2022) computes and utilizes the maximum value $\max_i p(y = y_i|x,\phi,\theta)$ (= s_{mcm}, MCM score) of the predicted probability (described in Equation 1) for detecting OOD samples in the dataset $\mathcal{D}$ by using a CLIP model. we can confidently identify an image as ID if its MCM score is above a certain threshold for ID $(i.e., \max_i p\,(y = y_i|x,\phi,\theta) \geq \delta_{\text{in}})$ and as OOD if $1 - s_{mcm}$ is above another threshold for OOD $(i.e., 1 - \max_i p\,(y = y_i|x,\phi,\theta) \geq \delta_{\text{out}})$. Inspired by curriculum learning strategies (Soviany et al., 2022), we propose a novel approach that adaptively adjusts both δ_{in} and δ_{out} based on the model's learning status for each class.

Adaptive class-wise threshold Our adaptive class-wise thresholds for both ID and OOD are defined by scaling the fixed thresholds δ_{in} and δ_{out} as:

$$\delta_{\text{in}}(y_i) = \beta_{\text{in}}(y_i)\cdot\delta_{\text{in}} \text{ and } \delta_{\text{out}}(y_i) = \beta_{\text{out}}(y_i)\cdot\delta_{\text{out}}, \quad (2)$$

where the class-wise scaling factors $\beta_{\text{in}}(y_i)$ and $\beta_{\text{out}}(y_i)$ are computed in the same manner and updated regularly (i.e., at each epoch). For example, we update the class-wise

Table 1. **Experiment results on Office-31, Office-Home, and VisDA**. We utilize the HOS score (%) as an evaluation metric. Note that models that can perform OSDA employ both the source data and the target data during the adaptation. Methods that can do source-free OSDA (SF-OSDA) employ models pre-trained on source data but use only target data for the adaptation. In contrast with these methods, UOTA only utilizes unlabeled target data and does not use either the source data or the model pre-trained on the source data.

Method	Office-31							Office-Home													VisDA
	W	D	A	D	A	W	Avg.	R	C	A	P	C	A	P	R	A	P	R	C	Avg.	S
	A		W		D			P			R			C			A				R
OSDA (open-set domain adaptation; use labeled source domain data)																					
DANN	72.6	73.7	68.1	86.7	71.5	82.5	75.9	68.4	60.9	65.2	69.8	66.7	71.0	44.6	50.9	51.2	56.3	65.4	57.6	60.7	-
CDAN	71.0	72.7	64.9	84.3	66.8	80.5	73.4	67.6	61.7	65.1	69.7	67.1	70.7	47.2	52.7	52.9	58.6	66.0	58.2	61.4	-
STA	66.1	73.2	75.9	69.8	75.0	75.2	72.5	64.5	60.4	54.0	69.5	66.8	68.3	53.2	54.5	55.8	61.9	67.1	57.4	61.1	72.7
OSBP	73.7	75.1	82.7	97.2	82.4	91.1	83.7	72.3	64.7	65.2	73.9	70.6	72.9	53.2	54.5	55.1	63.2	66.7	64.3	64.7	69.8
PGL	70.1	69.5	74.6	76.5	72.8	72.2	72.6	52.5	36.8	45.6	41.6	45.6	55.8	46.6	0.0	29.3	47.2	11.4	10.0	35.2	74.7
ROS	77.2	77.9	82.1	96.0	82.4	99.7	85.9	75.7	65.2	69.3	74.4	68.6	76.5	56.3	60.4	60.1	60.6	68.8	58.9	66.2	-
DANCE	70.2	65.8	66.9	80.0	70.7	84.8	73.1	44.0	45.9	49.8	41.2	30.2	39.4	55.7	48.3	53.1	54.2	27.5	40.9	44.2	-
DCC	84.4	85.5	87.1	91.2	85.5	87.1	86.8	62.7	66.6	67.4	64.0	67.0	80.6	52.8	76.9	52.9	59.5	56.0	49.8	64.2	70.7
OSLPP	78.7	79.3	89.0	92.3	91.5	93.6	87.4	74.4	66.9	72.8	74.0	70.4	74.3	59.3	59.0	61.0	63.6	67.2	60.9	67.0	-
UADAL	76.5	79.7	89.1	97.8	86.0	99.5	88.1	76.8	69.5	70.8	76.9	73.4	77.4	56.6	60.6	63.2	63.0	72.1	64.2	68.7	75.3
cUADAL	75.1	80.5	90.1	98.2	87.9	99.4	88.5	76.7	68.3	71.6	76.8	72.6	77.5	54.6	59.9	63.6	62.9	72.6	65.0	68.5	75.9
SF-OSDA (source-free open-set domain adaptation; use a pre-trained source domain model)																					
SHOT	75.9	74.0	69.1	87.2	67.2	92.7	77.7	42.3	40.2	39.8	46.2	39.1	47.0	40.8	40.1	39.5	57.7	59.9	54.6	45.6	42.6
AaD	73.9	73.0	78.3	91.2	77.7	93.5	81.3	70.1	61.4	66.9	70.6	67.8	69.9	55.9	57.5	57.6	60.1	64.6	60.5	63.6	16.0
Our Setting (neither use a source domain model nor source domain data)																					
Zero-shot	48.0		57.0		65.3		56.8	57.4			63.9			63.1			69.2			63.4	83.1
UOTA	**93.8**		**94.7**		**99.3**		**96.0**	**92.8**			**92.2**			**85.4**			**83.7**			**88.5**	**93.7**
Zero-shot (Oracle)	96.8		97.8		98.0		97.5	97.0			98.7			93.6			96.3			96.4	94.0
UOTA (Oracle)	**97.2**		**100**		**100**		**99.1**	**98.1**			**98.9**			**95.1**			**97.1**			**97.3**	**96.1**

scaling factor $\beta_{\text{in}}(y_i)$ for ID as:

$$\beta_{\text{in}}(y_i) = \frac{n_{\text{in}}(y_i) + \gamma \cdot \max_j n_{\text{in}}(y_j)}{(1+\gamma) \cdot \max_j n_{\text{in}}(y_j)}, \tag{3}$$

where $n_{\text{in}}(y_i)$ denotes the number of samples in the dataset $\mathcal{D}$ whose classes are predicted as y_i while presenting s_{mcm} (MCM scores) higher than δ_{in}. Here, γ is a smoothness factor to reduce the variability of scaling factors between classes. Similarly, we also update the class-wise scaling factor $\beta_{\text{out}}(y_i)$ for OOD using $1 - s_{mcm}$ higher than δ_{out}.

Self-training with in-distribution data For each image x, we formulate the sample-level self-training loss $\mathcal{L}'_{\text{in}}$ as:

$$\mathcal{L}'_{\text{in}}(p_1(x), p_2(x)) = \mathbb{1}_{[\max p_1(x) \geq \delta_{\text{in}}(\hat{p}_1(x))]} \mathcal{L}_{\text{ce}}(\hat{p}_1(x), p_2(x)), \tag{4}$$

where $p_1(x)$ and $p_2(x)$ are the predicted probabilities $p(y|\mathcal{A}_1(x), \phi, \theta)$ and $p(y|\mathcal{A}_2(x), \phi, \theta)$, respectively. $\mathcal{A}_1(x)$ and $\mathcal{A}_2(x)$ denote two randomly augmented views of x. The hard pseudo-label $\hat{p}_1(x)$ is obtained from $p_1(x)$. We formulate the overall loss related to ID data as:

$$\mathcal{L}_{\text{in}} = \frac{1}{2|\mathcal{B}|} \sum_{x \in \mathcal{B}} \mathcal{L}'_{\text{in}}(p_1(x), p_2(x)) + \mathcal{L}'_{\text{in}}(p_2(x), p_1(x)). \tag{5}$$

Utilizing out-of-distribution data We propose to incorporate OOD samples during task adaptation by customizing negative learning to explicitly reduce the MCM scores of OOD samples. For each sample x, we define the sample-level negative learning loss $\mathcal{L}'_{\text{out}}$ as: $\mathcal{L}'_{\text{out}}$ as:

$$\mathcal{L}'_{\text{out}}(p_1(x), p_2(x)) = \mathbb{1}_{[1 - \max p_1(x) \geq \delta_{\text{out}}(\hat{p}_1(x))]} \mathcal{L}_{\text{ce}}(\hat{p}_1(x), 1 - p_2(x)). \tag{6}$$

We formulate the overall loss related to OOD data as:

$$\mathcal{L}_{\text{out}} = \frac{1}{2|\mathcal{B}|} \sum_{x \in \mathcal{B}} \mathcal{L}'_{\text{out}}(p_1(x), p_2(x)) + \mathcal{L}'_{\text{out}}(p_2(x), p_1(x)). \tag{7}$$

Contrastive loss as an additional regularizer We adopt the contrastive loss $\mathcal{L}_{\text{cont}}$ proposed in SimCLR (Chen et al., 2020) and use it as a regularizer to enhance not only OOD detection but also the adaptation performance.

Therefore, the overall loss is:

$$\mathcal{L} = \mathcal{L}_{\text{in}} + \mathcal{L}_{\text{out}} + \omega \cdot \mathcal{L}_{\text{cont}}, \tag{8}$$

where ω is used as a balancing hyper-parameter. After the task adaptation is finalized by optimizing the model with this overall loss, we use a fixed threshold δ_{ood} at test time to detect OOD samples by simply comparing it with MCM scores.

3. Experiments

UOTA is the first model to perform open-set task adaptation using only unlabeled data and without any source domain model or data, but leveraging the pre-trained zero-shot model (CLIP). As a result, there are no comparable models or experimental protocols available. Therefore, we compare our approach with the models that can perform OSDA and SF-OSDA using benchmarks utilized in OSDA. We evaluated the performance of UOTA by following the experimental settings of UADAL (Jang et al., 2022) and using a variety of benchmark datasets, including (i) Office-31 (Saenko et al., 2010), (ii) Office-Home (Venkateswara et al., 2017) and (iii) VisDA (Peng et al., 2017). We used HOS metric (Bucci et al., 2020) as it considers both known and unknown (ID and OOD) classification capabilities, providing a higher evaluation to models that excel in both.

3.1. Quantitative analysis

Tab. 1 shows that UOTA significantly improves "Zero-shot (pre-trained CLIP)" across all benchmarks. It also outperforms OSDA and SF-OSDA baselines, despite the more

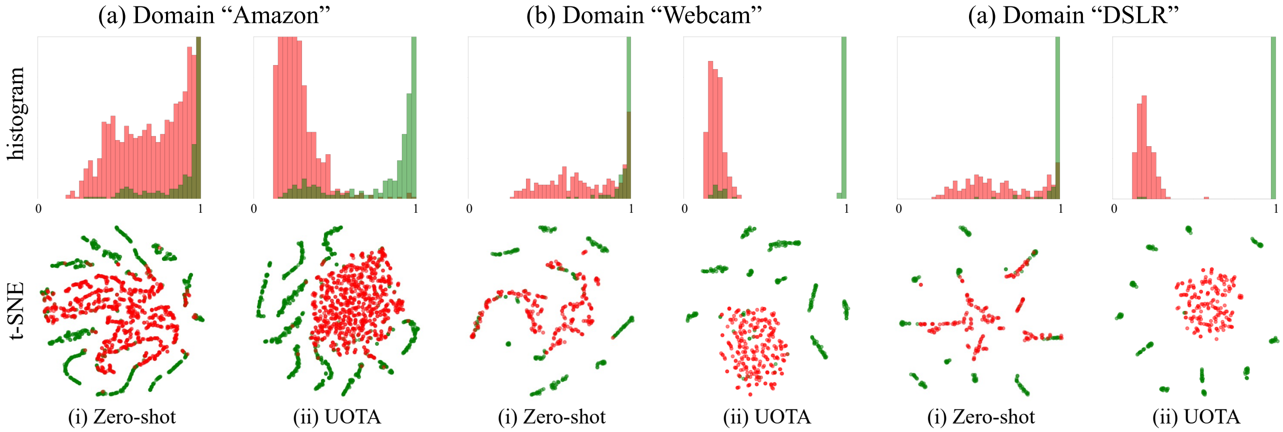


Figure 3. **Histogram and t-SNE visualization on Office-31.** The visualization results for (a) domain A, (b) domain W, and (c) domain D of the Office-31 dataset are shown with histograms and t-SNE plots. Across all domains, UOTA consistently exhibits improved performance over “Zero-shot”, with OOD samples (red) appearing more tightly clustered and a clearer separation between ID (green) and OOD samples.

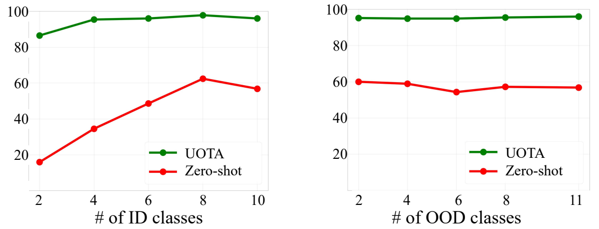
Table 2. **Ablation on the proposed training objectives.** The result demonstrates that the performance of the model is maximized when all losses are jointly used.

METHOD	OFFICE-31			
	A	W	D	AVG.
ZERO-SHOT	48.0	57.0	65.3	56.8
$\mathcal{L}_{\text{in}}$	DIVERGED			
$\mathcal{L}_{\text{out}}$	87.6	89.3	92.6	89.8
$\mathcal{L}_{\text{cont}}$	66.9	64.2	68.1	66.4
$\mathcal{L}_{\text{in}} + \mathcal{L}_{\text{cont}}$	47.6	57.4	66.3	57.1
$\mathcal{L}_{\text{out}} + \mathcal{L}_{\text{cont}}$	89.2	89.9	92.9	90.7
$\mathcal{L}_{\text{in}} + \mathcal{L}_{\text{out}}$	92.9	94.7	94.6	94.1
$\mathcal{L}_{\text{in}} + \mathcal{L}_{\text{out}} + \mathcal{L}_{\text{cont}}$ (UOTA)	**93.8**	**94.7**	**99.3**	**96.0**

challenging setting assumed for it. Additionally, to evaluate the effectiveness of the training strategy and determine the maximum performance of UOTA, we utilize an oracle setting that assumes perfect OOD detection performance, and the HOS score increases only if ID classification accuracy improves. UOTA consistently outperforms “Zero-shot” in an oracle setting, although the effect of its loss functions is largely offset since one of UOTA's training objectives mainly focuses on learning a discriminative representation for ID and OOD separation.

3.2. Distinguishing OOD samples

In the histograms of Figure 3, the horizontal axis represents the MCM score, while the vertical axis indicates the number of samples. We observe that “Zero-shot” is unable to clearly distinguish between ID (green) and OOD (red) samples. In contrast, UOTA effectively separates ID and OOD samples by predicting generally low MCM scores for OOD samples and high scores for ID samples. In the next step, we present the t-SNE visualizations of the learned features by “Zero-shot” and UOTA in Figure 3. Each data point in the figure represents the classification probability vector (as described in Equation 1) for each sample. The figure illustrates that the features for OOD samples (red) obtained by “Zero-shot” are not well distinguished from the features for ID samples (green). In contrast, UOTA precisely segregates OOD samples from ID data.



(a) Different # of ID classes.

(b) Different # of OOD classes.

Figure 4. **Robustness on different number of ID and OOD classes.**

3.3. Effectiveness of the proposed training objectives

When all of the losses are used together (UOTA), the separation of ID and OOD samples becomes more precise, and the performance of ID image classification greatly improves, resulting in the highest performance as shown in Tab. 2.

3.4. Robustness on varying the ratio between ID and OOD samples.

As shown in Figure 4, regardless of the varying number of known or unknown classes, UOTA (green) consistently outperforms “Zero-shot” (red) by a significant margin. We use the average HOS scores of all domains in Office-31.

4. Conclusion

We address the challenge of building a reliable image classification model in real-world scenarios by leveraging large amounts of unlabeled data in the wild, including both ID and OOD classes. To achieve this, we propose UOTA that significantly improves the zero-shot capabilities of CLIP, without requiring any task-specific human-labeled data. UOTA offers a promising direction for utilizing unlabeled data and enhancing zero-shot model transferability.

References

Bendale, A. and Boult, T. E. Towards open set deep networks. In *CVPR*, 2016.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *NeurIPS*, 2020.

Bucci, S., Loghmani, M. R., and Tommasi, T. On the effectiveness of image rotation for open set domain adaptation. In *ECCV*, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *JMLR*, 2016.

Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., and Qiao, Y. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021.

Ghifary, M., Kleijn, W. B., and Zhang, M. Domain adaptive neural networks for object recognition. In *PRICAI*, 2014.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

Huang, Y., Wang, Y., Tai, Y., Liu, X., Shen, P., Li, S., Li, J., and Huang, F. Curricularface: adaptive curriculum learning loss for deep face recognition. In *CVPR*, 2020.

Huynh, D. and Elhamifar, E. Fine-grained generalized zero-shot learning via dense attribute-based attention. In *CVPR*, 2020.

Jang, J., Na, B., Shin, D., Ji, M., Song, K., and Moon, I.-C. Unknown-aware domain adversarial learning for open-set domain adaptation. *arXiv preprint arXiv:2206.07551*, 2022.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021a.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021b.

Kim, Y., Yim, J., Yun, J., and Kim, J. Nlnl: Negative learning for noisy labels. In *ICCV*, 2019.

Kong, S. and Ramanan, D. Opengan: Open-set recognition via open data generation. In *ICCV*, 2021.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, G., Kang, G., Zhu, Y., Wei, Y., and Yang, Y. Domain consensus clustering for universal domain adaptation. In *CVPR*, 2021.

Li, J., Savarese, S., and Hoi, S. C. Masked unsupervised self-training for zero-shot image classification. *arXiv preprint arXiv:2206.02967*, 2022.

Li, S., Zhu, X., Huang, Q., Xu, H., and Kuo, C.-C. J. Multiple instance curriculum learning for weakly supervised object detection. In *WACV*, 2017.

Liang, J., Hu, D., and Feng, J. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

Liu, H., Cao, Z., Long, M., Wang, J., and Yang, Q. Separate to adapt: Open set domain adaptation via progressive separation. In *CVPR*, 2019.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. In *NeurIPS*, 2018.

Ming, Y., Cai, Z., Gu, J., Sun, Y., Li, W., and Li, Y. Delving into out-of-distribution detection with vision-language representations. In *NeurIPS*, 2022.

Mu, N., Kirillov, A., Wagner, D., and Xie, S. Slip: Self-supervision meets language-image pre-training. In *ECCV*, 2022.

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.

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 *ICML*, 2021.

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

Saito, K., Yamamoto, S., Ushiku, Y., and Harada, T. Open set domain adaptation by backpropagation. In *ECCV*, 2018.

Saito, K., Kim, D., Sclaroff, S., and Saenko, K. Universal domain adaptation through self supervision. In *NeurIPS*, 2020.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boult, T. E. Toward open set recognition. *TPAMI*, 2012.

Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016.

Soviany, P., Ionescu, R. T., Rota, P., and Sebe, N. Curriculum learning: A survey. *IJCV*, 2022.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *CoRR*, 2018.

Vaze, S., Han, K., Vedaldi, A., and Zisserman, A. Open-set recognition: A good closed-set classifier is all you need. In *ICLR*, 2022.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

Wang, W., Zheng, V. W., Yu, H., and Miao, C. A survey of zero-shot learning: Settings, methods, and applications. *ACM TIST*, 2019.

Wang, X., Ye, Y., and Gupta, A. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, 2018.

Xian, Y., Schiele, B., and Akata, Z. Zero-shot learning-the good, the bad and the ugly. In *CVPR*, 2017.

Yang, S., Wang, Y., Wang, K., Jui, S., et al. Attracting and dispersing: A simple approach for source-free domain adaptation. In *NeurIPS*, 2022.

Zhang, B., Wang, Y., Hou, W., WU, H., Wang, J., Okumura, M., and Shinozaki, T. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *NeurIPS*, 2021a.

Zhang, D., Yang, L., Meng, D., Xu, D., and Han, J. Spftn: A self-paced fine-tuning network for segmenting objects in weakly labelled videos. In *CVPR*, 2017.

Zhang, J., Xu, X., Shen, F., Lu, H., Liu, X., and Shen, H. T. Enhancing audio-visual association with self-supervised curriculum learning. In *AAAI*, 2021b.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. *IJCV*, 2022.

Zhou, T., Wang, S., and Bilmes, J. Curriculum learning by dynamic instance hardness. In *NeurIPS*, 2020.

UOTA: Unsupervised Open-Set Task Adaptation Using a Vision-Language Foundation Model

- Supplementary Materials -

We provide supplementary materials for "UOTA: Unsupervised Open-Set Task Adaptation Using a Vision-Language Foundation Model" in this document.

A. Related work

Multimodal zero-shot model The conventional approach to zero-shot learning involves training a model on base classes (Xian et al., 2017; Wang et al., 2019) and using auxiliary information such as attributes (Huynh & Elhamifar, 2020) or knowledge graphs (Wang et al., 2018) to recognize unseen classes. CLIP (Radford et al., 2021) introduced a new method for open-vocabulary zero-shot image classification using natural language supervision on large datasets. ALIGN (Jia et al., 2021a) is similar to CLIP, but it aligns the visual and language representations in a shared latent space and shows improved performance, even with noisy image-text paired data. SLIP (Mu et al., 2022) proposed a combined pre-training objective that consists of CLIP's loss function and self-supervision. Some recent works have attempted to adapt CLIP to downstream tasks using labeled data (Zhou et al., 2022; Gao et al., 2021) or unsupervised fine-tuning (Li et al., 2022). Moreover, MCM (Ming et al., 2022) proposed a training-free OOD detection using pre-trained CLIP, but its limitations include relying solely on the zero-shot transferability that pre-trained CLIP originally possesses and not improving its OOD detection ability through training. Going beyond the existing works, we propose, for the first time in the literature, a novel method that significantly and simultaneously improves the OOD detection and image classification capabilities of CLIP by utilizing only open-set unlabeled data.

Open-set domain adaptation In real-world scenarios, the set of classes in the target distribution may expand to include *unknown* classes, which leads to the field of open-set domain adaptation (OSDA) (Saito et al., 2018; Liu et al., 2019). Previous OSDA methods have focused on aligning the features of known classes in the source and target domains through domain adversarial learning (Saito et al., 2018; Liu et al., 2019). However, this approach may not be sufficient for learning the feature space of unknown classes in the target domain as there is no alignment signal provided by target-unknown instances. As a result, the classifier may not be able to learn an effective decision boundary for unknown classes as the target-unknown instances are not well-separated in the aligned feature space. Some methods attempt to address this issue by learning intrinsic target structures through self-supervised learning (Li et al., 2021; Saito et al., 2020). While conventional OSDA methods allow access to source domain data during the adaptation stages, recently proposed source-free OSDA (SF-OSDA) (Yang et al., 2022; Liang et al., 2020) methods utilize a model pre-trained on the source domain but do not use source data during the adaptation stage. In this paper, we propose a more restrictive setting than previous SF-OSDA, where the model uses neither source domain data nor a model pre-trained on source data. Despite being a more challenging scenario, our model significantly outperforms all existing OSDA and SF-OSDA methods.

B. Implementation details

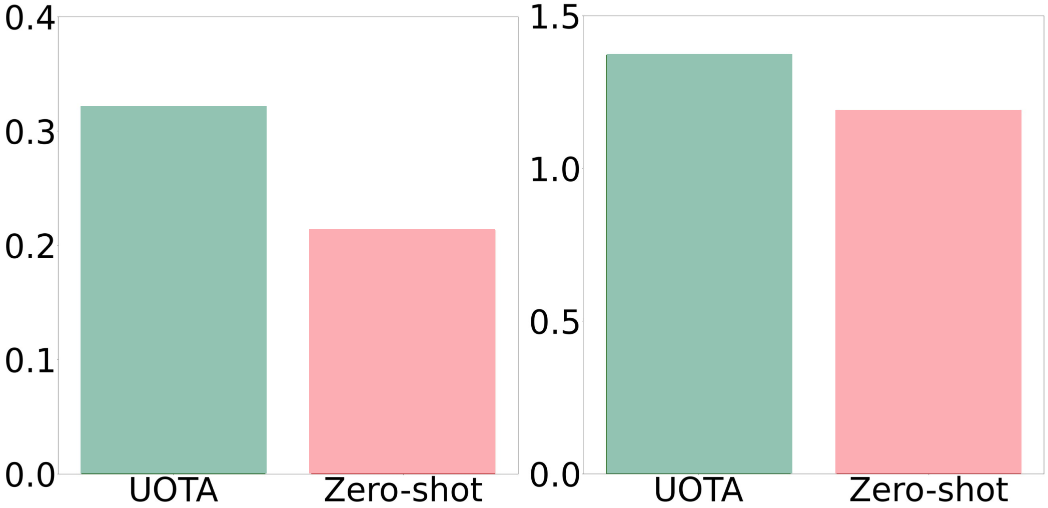
We evaluated the performance of UOTA by following the experimental settings of UADAL (Jang et al., 2022) and using a variety of benchmark datasets, including (i) Office-31 (Saenko et al., 2010), (ii) Office-Home (Venkateswara et al., 2017), and (iii) VisDA (Peng et al., 2017). The Office-31 dataset, described in (Saenko et al., 2010), consists of three distinct domains named Amazon (A), Webcam (W), and DSLR (D), encompassing a total of 31 classes. The Office-Home dataset presents a more challenging scenario with four different domains: Artistic (A), Clipart (C), Product (P), and Real-world (R) with a total of 65 classes. The VisDA dataset is a large-scale dataset consisting of images from synthetic to real-world scenarios with 12 classes.

To quantitatively evaluate the performance of UOTA, we compare it with several existing methods that can perform OSDA, including DANN (Ganin & Lempitsky, 2015), CDAN (Long et al., 2018), STA (Liu et al., 2019), OSBP (Saito et al., 2018), ROS (Bucci et al., 2020), DANCE (Saito et al., 2020), DCC (Li et al., 2021), UADAL (cUADAL) (Jang et al., 2022). Furthermore, we compare UOTA with state-of-the-art models capable of performing SF-OSDA, such as SHOT (Liang et al., 2020) and AaD (Yang et al., 2022). We also demonstrate the effectiveness of UOTA by comparing it with a pre-trained CLIP model (Radford et al., 2021), denoted as "Zero-shot". In particular, it is used as our initialization, and the main goal of our method is to further improve it. Hence, we can identify the effectiveness of UOTA through comparison with "Zero-shot". In addition, by adopting an oracle setting that assumes perfect OOD (Out-of-Distribution) detection performance, we can verify the upper bound of UOTA's performance. In this setting, we can also examine how much the UOTA can enhance the ID (In-Distribution) classification ability of "Zero-shot", even when the effect of one of its training objectives that aims to improve the UOTA's ability to separate ID and OOD data is largely offset.

To effectively evaluate the performance of UOTA, we utilized the HOS metric, which is commonly used as an evaluation criterion by existing OSDA approaches (Bucci et al., 2020; Jang et al., 2022; Yang et al., 2022). The HOS metric is calculated by taking the harmonic mean of OS* and UNK, where OS* represents the mean accuracy over known classes and UNK represents the accuracy of the unknown class. This metric is particularly suitable for evaluating models in OSDA tasks as it considers both known and unknown (ID and OOD) classification capabilities, providing a higher evaluation to models that excel in both. Therefore, we follow the established protocols of OSDA and mainly employ the HOS score as the evaluation metric.

C. Distance between ID and OOD feature distribution

We measure the distance between the ID and OOD feature distributions produced by "Zero-shot" and UOTA. For this, we use Proxy $\mathcal{A}$-Distance (PAD) (Ganin et al., 2016) and Maximum Mean Discrepancy (MMD) (Ghifary et al., 2014), and the corresponding results are shown in Figure 5. Higher PAD and MMD values indicate clearer discrimination between ID and OOD feature distributions. Our analysis reveals that UOTA (green) exhibits approximately 50% and 15% higher PAD and MMD, respectively, compared to "Zero-shot (red)". This suggests that UOTA is better able to distinguish between ID and OOD feature distributions.



(a) MMD

(b) PAD

Figure 5. **MMD and PAD values between known and unknown feature distributions.** UOTA (green) consistently exhibits a noticeable improvement in both metrics over "Zero-shot" (pre-trained CLIP, red). Each metric value is an average result across all domains of Office-31. This result demonstrates that UOTA more accurately distinguishes between ID and OOD distributions in comparison to "Zero-shot".

D. Robustness on different backbones

We also conduct ablation on backbones (i.e., feature extractors) to observe if UOTA consistently improves "Zero-shot" when given backbones with different scales. We compare three different backbones, denoted as (1) "ViT-B/16", (2) "ViT-B/32", and (3) "ViT-L/14" (our default backbone). We use the Office-31, Office-Home, and VisDA datasets, with the HOS score as the evaluation metric. As presented in Tab. 3, UOTA consistently shows improved average HOS scores compared to "Zero-shot" and presents state-of-the-art performance, even when the backbone is changed.

Table 3. **Ablation on different backbones.** UOTA achieves higher HOS scores than "Zero-shot" for all datasets and target domains, regardless of the scale of its backbone. The bold results represent the best scores, while the underlined one is the second-best score.

Method	Office-Home				
	P	R	C	A	Avg.
Zero-shot-ViT-B/16	61.6	66.2	67.7	69.8	66.3
Zero-shot-ViT-B/32	65.6	68.1	67.2	69.5	67.6
Zero-shot-ViT-L/14	57.4	63.9	63.1	69.2	63.4
UOTA-ViT-B/16	87.0	87.4	79.1	79.1	83.2
UOTA-ViT-B/32	84.2	86.2	75.1	75.9	80.4
UOTA-ViT-L/14 (Ours)	**92.8**	**92.2**	**85.4**	**83.7**	**88.5**

Method	Office-31				VisDA
	A	W	D	Avg.	R
Zero-shot-ViT-B/16	53.7	49.4	55.2	52.8	85.2
Zero-shot-ViT-B/32	52.7	66.1	63.6	60.8	84.0
Zero-shot-ViT-L/14	48.0	57.0	65.3	56.8	83.1
UOTA-ViT-B/16	89.6	**96.0**	98.1	94.6	89.7
UOTA-ViT-B/32	89.5	87.2	88.6	88.4	85.3
UOTA-ViT-L/14 (Ours)	**93.8**	94.7	**99.3**	**96.0**	**93.7**

E. Hyperparameters

Tab. 4 presents the hyperparameters utilized for "Zero-shot (pre-trained CLIP)" and UOTA in our experiment. While some adjustments were made to a few hyperparameters for specific datasets, it is noteworthy that the experiment was mostly conducted without any significant hyperparameter tuning. In fact, slight differences in hyperparameters did not have a considerable impact on the experimental results. This demonstrates our model's robustness on hyperparameters.

Table 4. **List of hyperparameters.**

Hyper-parameter	Office-31	Office-Home	VisDA	Office-31 (Oracle)	Office-Home (Oracle)	VisDA (Oracle)
batch size	32	32	32	32	32	32
optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
learning rate	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5
δ_{in}	0.95	0.95	0.95	0.95	0.95	0.95
δ_{out}	0.5	0.5	0.8	0.5	0.5	0.8
γ	4.0	4.0	4.0	4.0	4.0	4.0
ω	1.0	1.0	1.0	1.0	10.0	10.0
δ_{ood}	0.6	0.6	0.6	0.6	0.6	0.6

F. Supplemental experiment results on Office-31, Office-Home, and VisDA

In Tab. 5, 6, and 7, we provide additional results measuring the performance of UOTA and other existing models using OS* (accuracy over known classes) and UNK (accuracy of unknown classes). Note that, different from the HOS score, the OS* and UNK are biased evaluation metrics that do not simultaneously consider a model's ID classification and OOD detection capabilities. We employ Office-31, Office-Home, and VisDA as datasets. We utilize methods that can perform OSDA (DANN, CDAN, OSBP, STA, PGL, ROS, DANCE, DCC, OSLPP, UADAL, and cUADAL) and SF-OSDA (SHOT and AaD) as comparison approaches. We conduct experiments using the best settings for each of these models on their respective datasets (e.g., use ResNet50 as a backbone for Office-31 and Office-Home, and use VGGNet as a backbone for VisDA).

Table 5. Additional results on Office-31.

METHOD	OFFICE-31																				
	W			D			A			D			A			W			AVG.		
	A						W						D								
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS
DANN	72.1	73.1	72.6	72.9	74.5	73.7	87.4	55.7	68.1	99.3	77.0	86.7	90.8	59.2	71.5	100.0	70.2	82.5	87.1	68.3	75.9
CDAN	72.8	69.3	71.0	74.9	70.6	72.7	90.3	50.7	64.9	99.6	73.2	84.3	92.2	52.4	66.8	100.0	67.3	80.5	88.3	63.9	73.4
OSBP	73.0	74.4	73.7	76.1	72.3	75.1	86.8	79.2	82.7	97.7	96.7	97.2	90.5	75.5	82.4	99.1	84.2	91.1	87.2	80.4	83.7
STA	66.2	68.0	66.1	83.1	65.9	73.2	86.7	67.6	75.9	94.1	55.5	69.8	91.0	63.9	75.0	84.9	67.8	75.2	84.3	64.8	72.5
PGL	80.8	61.8	70.1	80.6	61.2	69.5	82.7	67.9	74.6	87.5	68.1	76.5	82.1	65.4	72.8	82.8	64.0	72.2	82.7	64.7	72.6
ROS	69.7	86.6	77.2	74.8	81.2	77.9	88.4	76.7	82.1	99.3	93.0	96.0	87.5	77.8	82.4	100.0	99.4	99.7	86.6	85.8	85.9
DANCE	83.7	60.6	70.2	85.3	53.6	65.8	98.7	50.7	66.9	100.0	66.8	80.0	96.5	55.9	70.7	100.0	73.7	84.8	94.0	60.2	73.1
DCC	-	-	84.4	-	-	85.5	-	-	87.1	-	-	91.2	-	-	85.5	-	-	87.1	-	-	86.8
OSLPP	78.9	78.5	78.7	82.1	76.6	79.3	89.5	88.4	89.0	96.9	88.0	92.3	92.6	90.4	91.5	95.8	91.5	93.6	89.3	85.6	87.4
UADAL	67.4	88.4	76.5	73.3	87.3	79.7	84.3	94.5	89.1	99.3	96.3	97.8	85.1	87.0	86.0	99.5	99.4	99.5	84.8	92.1	88.1
cUADAL	65.6	87.8	75.1	74.2	87.8	80.5	85.5	95.1	90.1	98.7	97.7	98.2	85.6	90.4	87.9	99.3	99.4	99.4	84.8	93.0	88.5
SHOT	72.2	80.1	75.9	75.5	72.5	74.0	74.5	64.4	69.1	96.7	79.4	87.2	82.0	56.9	67.2	98.8	87.2	92.7	83.3	73.4	77.7
AaD	70.8	78.2	73.9	69.8	77.4	73.0	74.6	83.5	78.3	90.2	92.5	91.2	75.3	80.9	77.7	92.1	95.2	93.5	78.8	84.6	81.3
UOTA	89.6	98.5	93.8	89.6	98.5	93.8	90.0	100.0	94.7	90.0	100.0	94.7	98.7	100.0	99.3	98.7	100.0	99.3	92.8	99.5	96.0

Table 6. Additional results on Office-Home.

METHOD	OFFICE-HOME																	
	R			C			A			P			C			A		
	P									R								
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS
CDAN	70.9	64.6	67.6	51.6	76.8	61.7	61.7	68.8	65.1	69.8	69.7	69.7	61.5	73.7	67.1	75.2	66.7	70.7
OSBP	76.3	68.6	72.3	67.0	62.7	64.7	71.8	59.8	65.2	76.2	71.7	73.9	72.0	69.2	70.6	79.3	67.5	72.9
STA	77.1	55.4	64.5	61.8	59.1	60.4	68.0	48.4	54.0	76.2	64.3	69.5	67.0	66.7	66.8	78.6	60.4	68.3
PGL	84.8	38.0	52.5	73.9	24.5	36.8	78.9	32.1	45.6	84.8	27.6	41.6	70.2	33.8	45.6	87.7	40.9	55.8
ROS	72.0	80.0	75.7	59.8	71.6	65.2	68.4	70.3	69.3	70.8	78.4	74.4	65.3	72.2	68.6	75.8	77.2	76.5
DANCE	86.2	29.6	44.0	76.3	32.8	45.9	84.0	35.4	49.8	86.5	27.1	41.2	83.9	18.4	30.2	89.8	25.3	39.4
DCC	-	-	62.7	-	-	66.6	-	-	67.4	-	-	64.0	-	-	67.0	-	-	80.6
LGU	83.2	46.8	59.9	71.7	4.1	7.8	80.5	49.3	61.2	82.8	41.2	55.0	77.6	46.4	58.1	86.5	47.5	61.3
OSLPP	78.4	70.8	74.4	61.6	73.3	66.9	72.5	73.1	72.8	77.0	71.2	74.0	67.2	73.9	70.4	80.1	69.4	74.3
UADAL	77.4	76.2	76.8	62.1	78.8	69.5	69.1	72.5	70.8	71.6	83.1	76.9	69.1	78.3	73.4	81.3	73.7	77.4
cUADAL	77.8	75.6	76.7	61.1	77.4	68.3	69.4	73.9	71.6	71.2	83.4	76.8	69.3	76.3	72.6	82.2	73.3	77.5
SHOT	84.4	28.2	42.3	77.5	27.2	40.2	81.8	26.3	39.8	85.8	31.6	46.2	80.0	25.9	39.1	87.5	32.1	47.0
AaD	69.7	70.6	70.1	59.5	63.5	61.4	64.6	69.4	66.9	68.4	72.8	70.6	67.4	68.3	67.8	73.1	66.9	69.9
UOTA	88.2	97.9	92.8	88.2	97.9	92.8	88.2	97.9	92.8	88.6	96.1	92.2	88.6	96.1	92.2	88.6	96.1	92.2

METHOD	OFFICE-HOME																				
	P			R			A			P			R			C			AVG.		
	C									A											
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS
DANN	30.1	86.3	44.6	37.1	80.9	50.9	37.1	82.7	51.2	42.4	83.9	56.3	56.8	77.1	65.4	43.8	84.3	57.6	52.6	77.1	60.7
CDAN	33.1	82.4	47.2	40.3	75.8	52.7	39.7	78.9	52.9	45.8	81.2	58.6	59.8	73.6	66.0	44.9	82.8	58.2	54.5	74.6	61.4
OSBP	44.5	66.3	53.2	48.0	63.0	54.5	50.2	61.1	55.1	59.1	68.1	63.2	66.1	67.3	66.7	59.4	70.3	64.3	64.1	66.3	64.7
STA	44.2	67.1	53.2	49.9	61.1	54.5	46.0	72.3	55.8	54.2	72.4	61.9	67.5	66.7	67.1	51.4	65.0	57.4	61.8	63.3	61.1
PGL	59.2	38.4	46.6	68.8	0.0	0.0	63.3	19.1	29.3	73.7	34.7	47.2	81.5	6.1	11.4	85.9	5.3	10.0	76.1	25.0	35.2
ROS	46.5	71.2	56.3	51.5	73.0	60.4	50.6	74.1	60.1	57.3	64.3	60.6	67.0	70.8	68.8	53.6	65.5	58.9	61.6	72.4	66.2
DANCE	48.2	67.4	55.7	60.1	41.3	48.3	54.4	53.7	53.1	70.7	43.9	54.2	79.2	16.7	27.5	72.9	28.4	40.9	74.4	35.0	44.2
DCC	-	-	52.8	-	-	76.9	-	-	52.9	-	-	59.5	-	-	56.0	-	-	49.8	-	-	64.2
LGU	54.5	18.1	27.2	63.4	29.6	40.4	58.6	32.6	41.9	69.1	50.9	58.6	77.5	48.9	60.0	67.2	30.8	42.2	72.7	38.9	50.7
OSLPP	53.1	67.1	59.3	54.4	64.3	59.0	55.9	67.1	61.0	54.6	76.2	63.6	60.8	75.0	67.2	49.6	79.0	60.9	63.8	71.7	67.0
UADAL	43.4	81.5	56.6	51.1	74.5	60.6	54.9	74.7	63.2	50.5	83.7	63.0	66.7	78.6	72.1	53.5	80.5	64.2	62.6	78.0	68.7
cUADAL	41.2	80.7	54.6	51.8	71.1	59.9	55.0	75.6	63.6	50.9	82.4	62.9	66.8	79.6	72.6	53.8	82.0	65.0	62.5	77.6	68.5
SHOT	59.3	31.0	40.8	65.3	28.9	40.1	67.0	28.0	39.5	66.3	51.1	57.7	73.5	50.6	59.9	66.8	46.2	54.6	74.6	33.9	45.6
AaD	45.4	72.8	55.9	49.0	69.6	57.5	50.7	66.4	57.6	47.3	82.4	60.1	54.5	79.0	64.6	48.2	81.1	60.5	58.2	71.9	63.6
UOTA	76.9	95.9	85.4	76.9	95.9	85.4	76.9	95.9	85.4	79.1	88.9	83.7	79.1	88.9	83.7	79.1	88.9	83.7	83.2	94.7	88.5

Table 7. Additional results on VisDA.

Method	VisDA		
	OS*	UNK	HOS
STA	63.9	84.2	72.7
OSBP	59.2	85.1	69.8
PGL	82.8	68.1	74.7
DCC	68.0	73.6	70.7
UADAL	61.1	93.3	75.3
cUADAL	64.3	92.6	75.9
SHOT	44.6	40.7	42.6
AaD	13.8	23.3	16.0
UOTA	89.4	98.4	93.7

G. Pytorch-style pseudocode for UOTA

Algorithm 1 UOTA: PyTorch Pseudocode

```
# img_1, img_2, img_encoder, txt_feat: View 1, view 2, image encoder, and text feature, respectively.
# alpha_1, alpha_2 : Learnable temperatures for sharpening the prediction.
# norm, batch_size, CE: Normalization, batch size, and cross-entropy loss, respectively.
# data_num, known_cls_num: Total number of images in a dataset and the number of known classes, respectively.

# count_in, count_out: Used for collecting pseudo-labels for each image. Initialized with -1s.
# omega, gamma: A balancing weight and a smoothness factor, respectively.
# delta_in, delta_out: Thresholds for IDs and OODs, respectively.



for (img_1, img_2, idx) in train_loader:

   img_feat_1, img_feat_2 = img_encoder(img_1), img_encoder(img_2)
   cos_sim_1 = alpha_1 * norm(img_feat_1, dim=1) @ norm(txt_feat, dim=1).T
   cos_sim_2 = alpha_1 * norm(img_feat_2, dim=1) @ norm(txt_feat, dim=1).T
   max_prob_1, max_idx_1 = max(softmax(cos_sim_1, dim=1), dim=1)
   max_prob_2, max_idx_2 = max(softmax(cos_sim_2, dim=1), dim=1)

   count_in, count_out, thres_in, thres_out = classwise_threshold(count_in, count_out, beta_in, beta_out)
   mask_in_1, mask_in_2 = max_prob_1.ge(thres_in[max_idx_1]), max_prob_2.ge(thres_in[max_idx_2])
   mask_out_1, mask_out_2 = (1-max_prob_1).ge(thres_out[max_idx_1]), (1-max_prob_2).ge(thres_out[max_idx_2])

   loss_in = (CE(max_prob_2,max_idx_1) *mask_in_1 + CE(max_prob_1,max_idx_2) * mask_in_2) / 2.0
   loss_out = (CE((1-max_prob_2),max_idx_1) * mask_out_1 + CE((1-max_prob_1),max_idx_2) * mask_out_2) / 2.0
   loss_cont = contrastive_loss(img_feat_1, img_feat_2, batch_size)

   loss = loss_in + loss_out + omega * loss_cont
   loss.backward()
   update(img_encoder.parameters())

   count_in_temp, count_out_temp = ones(data_num))*(-1), ones(data_num)*(-1)
   idx_in_1, idx_in_2 = max_prob_1.ge(delta_in), max_prob_2.ge(delta_in)
   idx_out_1, idx_out_2 = (1-max_prob_1).ge(delta_out), (1-max_prob_2).ge(delta_out)
   count_in_temp[idx[idx_in_1]], count_in_temp[idx[idx_in_2]] = max_idx_1[idx_in_1], max_idx_2[idx_in_2]
   count_out_temp[idx[idx_out_1]], count_out_temp[idx[idx_out_2]] = max_idx_1[idx_out_1], max_idx_2[
       idx_out_2]
   count_in_temp, count_out_temp = Counter(count_in_temp), Counter(count_out_temp)

   momentum = (batch_size*2 / data_num)
   for i in range(known_cls_num):
      count_in[i] = count_in[i]* (1-momentum) + count_in_temp[i]
      count_out[i] = count_out[i]* (1-momentum) + count_out_temp[i]


def classwise_threshold(count_in, count_out, beta_in, beta_out):

   for (img_1, img_2, idx) in train_loader:
      img = cat([img_1,img_2], dim=0)
      img_feat = img_encoder(img)
      cos_sim = alpha_1 * norm(img_feat, dim=1) @ norm(txt_feat, dim=1).T
      max_prob, max_idx = max(softmax(cos_sim, dim=1), dim=1)
      idx_in, idx_out = max_prob.ge(delta_in), (1-max_prob).ge(delta_out)
      count_in[idx[idx_in]], count_out[idx[idx_out]] = max_idx[idx_in], max_idx[idx_out]

   count_in, count_out = Counter(count_in), Counter(count_out)
   max_in, max_out = max(count_in.values()), max(count_out.values())

   for i in range(known_cls_num):
      if i in count_in:
         beta_in = (count_in[i] + gamma*max_in) / (1+gamma)*max_in
      if i in count_out:
         beta_out = (count_out[i] + gamma*max_out) / (1+gamma)*max_out

   return count_in, count_out, beta_in*delta_in, beta_out*delta_out


def contrastive_loss(feat_1, feat_2, batch_size):

   feat_1, feat_2 = norm(feat_1, dim=1), norm(feat_2, dim=1)
   label = arange(batch_size)
   mask = eye(batch_size) * 1e9
   matrix = feat_1 @ feat_2.T
   matrix1 = feat_1 @ feat_1.T - mask
   matrix2 = feat_2 @ feat_2.T - mask
   matrix1, matrix2 = cat([matrix, matrix1], dim=0), cat([matrix.T, matrix2], dim=0)
   loss = (CE(matrix1 / alpha_2, label) + CE(matrix2 / alpha_2, label)) / 2.0
   return loss
```