Boosting Novel Category Discovery Over Domains with Soft Contrastive Learning and All in One Classifier

Zelin Zang [†] Westlake University
Lei Shang[†] Alibaba Group
Senqiao Yang[†] Westlake University
Baigui Sun, Xuansong Xie Alibaba Group
Stan Z. Li Westlake University

Abstract

Unsupervised domain adaptation (UDA) has been highly successful in transferring knowledge acquired from a label-rich source domain to a label-scarce target domain. Open-set domain adaptation (ODA) and universal domain adaptation (UNDA) have been proposed as solutions to the problem concerning the presence of additional novel categories in the target domain. Existing ODA and UNDA approaches treat all novel categories as one unified unknown class and attempt to detect this unknown class during the training process. We find that domain variance leads to more significant view-noise in unsupervised data augmentation, affecting the further applications of contrastive learning (CL), as well as the current closed-set classifier and open-set classifier causing the model to be overconfident in novel class discovery. To address the above two issues, we propose Soft-contrastive All-in-one Network (SAN) for ODA and UNDA tasks. SAN includes a novel data-augmentation-based CL loss, which is used to improve the representational capability, and a more human-intuitive classifier, which is used to improve the new class discovery capability. The soft contrastive learning (SCL) loss is used to weaken the adverse effects of the data-augmentation label noise problem which is amplified in domain transfer. The All-in-One (AIO) classifier overcomes the overconfidence problem of the current mainstream closed-set classifier and open-set classifier in a more human-intuitive way. The visualization results and ablation experiments demonstrate the importance of the two proposed innovations. Moreover, extensive experimental results on ODA and UNDA show that SAN has advantages over the existing state-of-the-art methods.[1]

1. Introduction

Domain adaptation (DA) transfers knowledge from label-rich training domains to new domains where labels are scarce [1], to address the problem of generalization of deep neural networks in new domains. Traditional unsupervised domain adaptation (UDA) assumes that the source domain and the target domain completely share the sets of categories, i.e., closed-set DA. But, this assumption does not often hold in practice. There are several possible situations: the target domain contains types absent in the source (unknown categories), i.e., open-set DA (ODA) [3, 27]; the source domain includes classes absent in the target (source private categories); i.e., partial DA (PDA) [4]; a mixture of ODA and PDA, called open-partial DA (OPDA). Many approaches have been tailored for a specific setting, but an actual difficulty is that we cannot know the category shift in advance. The task of UNDA is proposed [25, 37] to account for the uncertainty about the category shift. The assumption is that the label distributions of labeled and unlabeled data can differ, but we do not know the difference in advance. The UNDA is a uniform and practical setting, since estimating the label distributions of unlabeled data is very hard in real applications.

For the ODA and UNDA tasks, it is crucial to discover private (novel) classes in the target domains. We consider that the current method's poor consideration of both unsupervised pre-training and novel category classifier design affects the further improvement of the above task (in Fig. 1).

The view-noise problem in data augmentation affects the pre-training of the backbone. Recently, data-augmentation-based contrastive learning (CL) has been used for unsupervised pre-training and has yielded excellent results in various downstream tasks [7, 39, 40]. Therefore, several studies [5, 38] have attempted to enhance UNDA by introducing CL. However, these approaches are not based on data-augmentation-based CL schemes but are based on neighbor relationships of the original data. We believe that data-augmentation-based CL schemes have great potential, as long as the view-noise problem of data augmentation during cross-domain training is addressed. Although the view-noise in CL has started to receive attention in network pre-training [9], we found that it causes more severe damage due to domain differences of UNDA. As shown at the top

[†]These authors contributed equally to this work.

[1]code in https://anonymous.4open.science/r/code_SAN_CVPR_5598

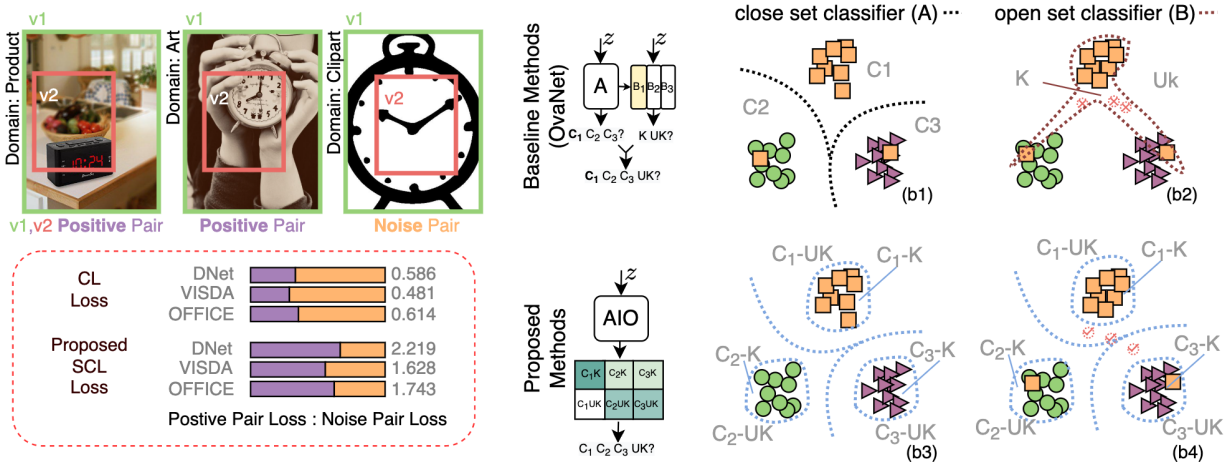

Figure 1. **The problems of current methods and the solution proposed by this paper**. (a) View-noise problem in the backbone network pre-training by the CL. (a)-top shows the views generated by the data augmentation in the three domains. The difference in content style of the Clipart domain causes the regular data augmentation to produce views with vastly different semantics, producing noisy pairs. (b) Overconfidence problem of novel category classifiers. The dashed circle with a tick/cross means the test samples are classified correctly/incorrectly.

of Fig. 1 (a), a more severe view-noise problem occurs if the same augmentation scheme is used in different domain data. In detail, view 1 (v1) and view 2 (v2) have similar semantics in the 'product' domain and 'art' domain, noted as positive pairs, but they have different semantics in the 'clipart' domain, noted as noise pairs. The noise pairs contradict the accurate semantic information and, therefore, generate false gradients, which corrupt the network training. More importantly, other data augmentation strategies, such as colour perturbation, also lead to divergent semantic changes in different domains, leading to the view-noise problem.

Overconfident classifiers (closed-set classifier and open-set classifier) affect novel category recognition performance. Recently, OVANet [26] and its variants [34] have received much attention. These methods combine closed-set classifiers and open-set classifiers to identify known and unknown classes. However, each of the open-set classifiers only corresponds to a single known class. When determining whether a sample belongs to a novel class, the open-set classifier does not compete with other open-set classifiers. It only determines whether the output of this classifier is greater than a certain threshold. This counter-intuitive strategy causes insufficient inter-class competition, which in turn leads to classifiers that are more likely to fall into overfitting and overconfidence. When label noise is present in the source domain, such data is almost inevitable and the harmful effects of overconfidence are amplified. As shown at the top of Fig. 1 (b), even though the closed-set classifier is not affected by label-noise, the classification boundary of the open-set classifier can become very sharp due to label-noise and overfitting issues, which eventually causes the target domain samples to be misclassified.

To solve the above problems, we propose the *Soft-contrastive All-in-one Network* (SAN) for UNDA and ODA. **To solve the view-noise problem**, we design a soft contrastive learning (SCL) loss. SCL loss incorporates the idea of self-distillation, where useful knowledge is extracted from the training network to counteract the detrimental effects of view-noise. Specifically, SCL loss includes the similarity of the latent space to define the reliability of the view and then further constructs the CL loss function with the help of reliability. To visualize and understand the adverse effects of noise pairs, we quantified the proportion of losses calculated from 'noise pairs' and 'positive pairs' using manual labeling. As shown at the bottom of Fig. 1 (a), the scale of the CL losses from the 'noise pairs' is much larger than the scale of the SCL losses. **To solve the overconfidence problem of independent classifiers**. An all-in-one (AIO) classifier is designed to replace the closed-set classifier and open-set classifier. What is closer to human common sense is that the AIO classifier assumes that *identifying a sample as belonging to a novel category requires determining that it does not belong to all known classes*. Based on the above assumptions, a new loss function is defined to train the AIO classifier. As shown in (b3) and (b4) of Fig. 1 (b), with the introduction of more comprehensive competition, the AIO classifier has smoother classification boundaries and reduces the adverse effects of label noise.

In experiments, we extensively evaluate our method on

ODA and UNDA benchmarks and vary the proportion of unknown classes. The results show that the proposed SAN outperforms all baseline methods on various datasets of the ODA and UNDA tasks.

2. Related work

Domain Adaptation (DA). Unsupervised domain adaptation (UDA) [24] aims to learn a classifier for a target domain given labeled source and unlabeled target data. UDA includes closed-set domain adaptation (CDA), open-set domain adaptation (ODA), partial domain adaptation (PDA), and universal domain adaptation (UNDA). For CDA, we have $L_s = L_t$, where L_s and L_t is the label space of a source and a target domain [12, 19, 29]. For ODA [21, 27], we have $|L_t - L_s| > 0, |L_t \cap L_s| = |L_s|$, the $|L_t - L_s|$ presence of target-private classes. For PDA, we have $|L_s - L_t| > 0, |L_t \cap L_s| = |L_t|$. The $|L_s - L_t|$ presence of source-private classes.

Universal Domain Adaptation (UNDA). UNDA, also called open-partial domain adaptation (OPDA) in some former works, is proposed to handle the mixture of these settings, $|L_s - L_t| > 0, |L_t - L_s| > 0$. [25] emphasize the importance of measuring the robustness of a model to various category shifts since we cannot know the detail of the shifts in advance. [11, 25, 37] compute a confidence score for known classes, and samples with a score lower than a threshold are regarded as unknown. [11] validate the threshold using labeled data, which is not a realistic solution. [2] set the mean of the confidence score as the threshold, which implicitly rejects about half of the target data as unknown. [25] sets a threshold decided by the number of classes in the source, which does not always work well. [34] reveals that exploiting such inter-sample affinity can significantly improve the performance of UNDA and proposes a knowability-aware UNDA framework based on it.

Contrastive learning based UNDA. Recently, contrastive learning (CL), a kind of self-supervised learning paradigm [36], has achieved impressively superior performance in many computer vision tasks [7]. It aims to achieve instance-level discrimination and invariance by pushing semantically distinct samples away while pulling semantically consistent samples closer in the feature space [8, 32]. [5] proposes to utilize mutual nearest neighbors as positive pairs to achieve feature alignment between the two domains. [6] constructs the random walk-based MNN pairs as positive anchors intra- and inter-domains and then proposes a cross-domain subgraph-level CL objective to aggregate local similar samples and separate different samples. No data augmentation-based CL schemes are used to solve the UNDA problem.

3. Methods

Notation. In ODA and UNDA, we are given a source domain dataset $\mathcal{D}_s = \{(\mathbf{x}_i^s, \widehat{y}_i^s)\}_{i=1}^{N_s}$ and a target domain dataset $\mathcal{D}_t = \{(\mathbf{x}_i^t)\}_{i=1}^{N_t}$ which contains known categories and 'unknown' categories. L_s and L_t denote the label spaces of the source and target, respectively. We assume that there is unavoidable noise and errors in the labels, so $\widehat{y}_i^s$ is noted as sampling from the real label y_i^s. The class-conditional random noise model is given by $P(\widehat{y}_i^s \neq y_i^s) = \rho^s$. We aim to label the target samples with either one of the L_s labels or the 'unknown' label. We train the model on $\mathcal{D}_s \cup \mathcal{D}_t$ and evaluate on $\mathcal{D}_t$.

Framework. Fig. 2 introduces the conceptual overview of SAN. The proposed method includes a backbone network $F(\cdot)$, a projection head network $H(\cdot)$, and an all-in-one (AIO) classifier $C^{\text{AIO}}(\cdot)$. The backbone network $F(\cdot)$ and projection head network $H(\cdot)$ map the source domain data $\mathbf{x}_i^s$ and the target domain data $\mathbf{x}_i^t$ into latent space, $\mathbf{z}_i^s = H(\hat{z}_i^s) = H(F(x_i^s)), \quad \mathbf{z}_i^t = H(\hat{z}_i^t) = H(F(x_i^t))$.

3.1. View-noise and Soft Contrastive Learning Loss

Data augmentation-based contrastive learning (CL) requires consideration of the relationships of data views. Given two views v_1 and v_2, the CL can be interpreted as binary classification operating over pairs of samples. If v_1, v_2 are sampled from the joint distribution $(v_1, v_2) \sim P_{v_1,v_2}$, then with label $\mathcal{H}_{v_1,v_2} = 1$. If v_1, v_2 are sampled from the product of marginals $(v_1, v_2) \sim P_{v_1} P_{v_2}$, then with label $\mathcal{H}_{v_1,v_2} = 0$. In practice, some negative pairs could be mislabeled as positive, and some positive pairs $(v_1, v_2) \sim P_{v_1,v_2}$ could be mislabeled as negative, introducing noisy labels.

Typical CL learns representations by maximizing the similarity between positive samples and minimizing the similarity between negative samples. To see this more concretely, consider the InfoNCE loss [30]:

$$\begin{aligned} &L_{\text{CL}}\left(v_i, v_j, \{v_k\}_{k\in\{1,\ldots,N_K\}}\right) \\ =&-\log \frac{\exp(z_i^T z_j/\tau)}{\sum_{k=1}^{N_K} \exp(z_i^T z_k/\tau)} = -\log \frac{\exp(S_{ij})}{\sum_{k=1}^{N_K} \exp(S_{ik})} \end{aligned} \tag{1}$$

where (v_i, v_j) is positive pair and (v_i, v_k) is negative pair, and z_i, z_j, z_k are the embedding of v_i, v_j, v_k, N_K is the number of the negative pair. The similarity $S(z_i, z_j)$ is typically defined by cosine similarity $S_{ij} = \exp(z_i \cdot z_j)$.

Typical CL loss considers the case of one positive sample and multiple negative samples. To design a smooth CL loss, we transform it into a form based on positive and negative sample labels $\mathcal{H}_{ij}$. The details of the transformation from

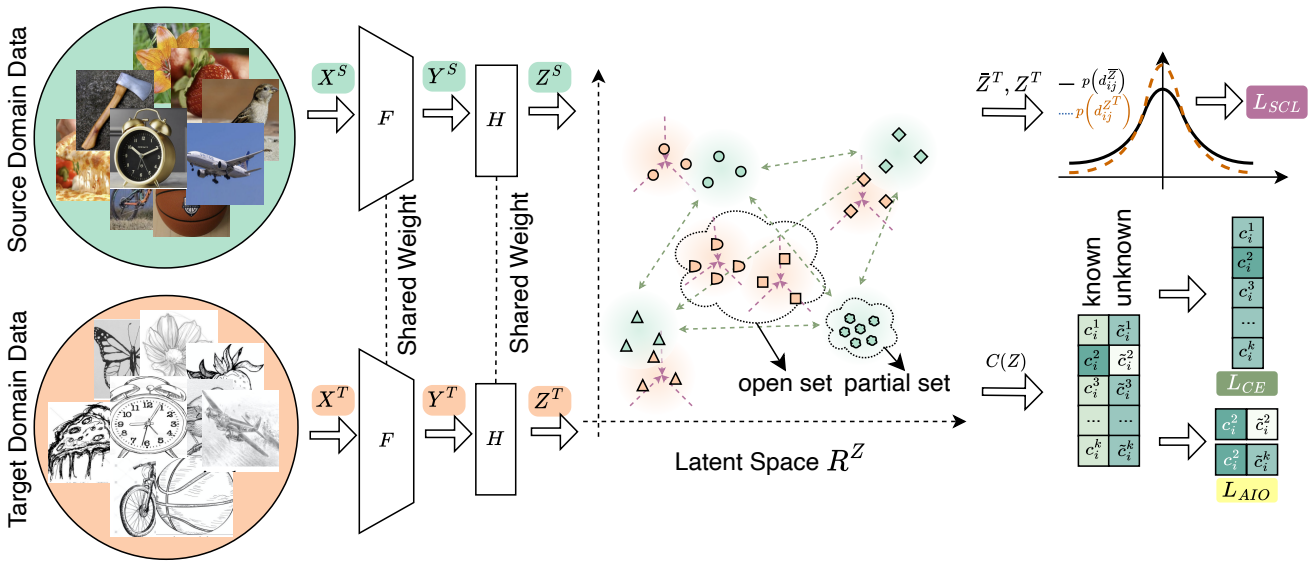

Figure 2. **Framework of SAN.** The proposed method includes a backbone network $F(\cdot)$, a projection head network $H(\cdot)$, and an all-in-one (AIO) classifier $C(\cdot)$. The backbone network $F(\cdot)$ and projection head network $H(\cdot)$ map the source domain data $\mathbf{x}^s$ and the target domain data $\mathbf{x}^t$ into latent space.

Eq. (1) to Eq. (2) are in Appendix A.

$$\begin{aligned} &L_{\mathrm{CL}}(x_i, \{x_j\}_{k\in\{1,\dots,N_J\}}) \\ &= -\sum_{j=1}\{\mathcal{H}_{ij}\log Q_{ij} + (1-\mathcal{H}_{ij})\log \dot{Q}_{ij}\} \end{aligned} \tag{2}$$

The $\mathcal{H}_{ij}$ indicates whether i and j are augmented from a same data, if $\mathcal{H}_{ij} = 1$ indicates (v_i, v_j) is positive pair, and $\mathcal{H}_{ij} = 0$ indicates (v_i, v_j) is negative pair. The $Q_{ij} = \exp S(z_i, z_j)$ is the density ratio, defined in [30], estimated by the backbone network.

Generally speaking, the positive pairs come from stochastic data augmentation, meaning that the learning process inevitably introduces view-noise. Further, view-noise inevitably introduces the wrong gradient, which corrupts the network's training. In particular, suitable data augmentation schemes for all domains are difficult to find for UNDA data with a vast amount of domain variance. We consider that the above reasons explain why no data enhancement-based UNDA methods have been proposed.

To solve the above problem, we propose Soft Contrastive Learning (SCL) with the assistance self distillation. The SCL predicts the confidence probability of the sample based on the the under the backbone network. Then, the SAN is further trained based on the confidence probability. the SCL loss is

$$\begin{aligned} &L_{\mathrm{SCL}}(x_i, \{x_j\}_{j\in\{1,\dots,N_J\}}) \\ &= -\sum_{j=1}\{P_{ij}\log(Q_{ij}) + (1-P_{ij})\log(1-Q_{ij})\}, \end{aligned} \tag{3}$$

where the P_{ij} is a soft version of $\mathcal{H}_{ij}$ in Eq. (2), and Q_{ij} is the density ratio.

$$\begin{aligned} P_{ij} &= \begin{cases} e^{\alpha}\kappa(y_i, y_j) & \text{if } \mathcal{H}(x_i, x_j) = 1 \\ \kappa(y_i, y_j) & \text{otherwise} \end{cases}, \\ Q_{ij} &= \kappa(z_i, z_j), \end{aligned} \tag{4}$$

Soft contrastive learning aims to let $Q_{ij} \to P_{ij}$. The $\alpha \in [0, 1]$ is a softened hyper-parameters, which is used to introduce the prior knowledge of data augmentation $\mathcal{H}(x_i, x_j)$ into the model training.

The kernel function $\kappa(\cdot)$ is used to map the high dimensional embedding vector (such as (y_i, y_j)) into probability value. All commonly used kernel functions can be employed, such as Gaussian kernel functions [15], radial basis kernel functions [28] and t-distribution kernel functions [20]. In this paper, we use the t-distribution kernel function $\kappa^{\nu}(\cdot)$ because it exposes the degrees of freedom and allows us to adjust the closeness of the distribution in the dimensionality reduction mapping [17, 39].

$$\kappa^{\nu}(z_i, z_j) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1 + \frac{\|z_i - z_j\|_2^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \tag{5}$$

where $\Gamma(\cdot)$ is the Gamma function. The degrees of freedom ν control the shape of the kernel function. The different degrees of freedom (ν^y, ν^z) is used in $\mathcal{R}^y$ and $\mathcal{R}^z$ for the dimensional reduction mapping.

SCL loss uses a softened optimization target instead of the hard target of a typical CL loss while avoiding the strong misresponse to noise labels. A formal discussion of the differences between the two losses is in Appendix A.3. In addition, we can prove that SCL loss can maintain a higher

signal-to-noise ratio in the view-noise, see more details in Appendix A.3.

3.2. Overconfidence and All in One (AIO) Classifier

Current advanced UNDA [26, 34] combines closed-set classifier C^A, and open-set classifiers $\{C_k^B\}_{k\in\mathcal{K}}$ to identify samples belonging to an unknown class or a specific known class, where the $\mathcal{K}$ is known classes set $\mathcal{K} \in \{1, \ldots, N_K\}$ and N_K is the number of the known classes. The inference process consists of two steps. Firstly, C^A identifies the most likely target class. Secondly, one of the sub-classifiers C_k^B determines whether the sample is a known or unknown class (see Fig. 1 (b) baseline method). In training a single open-set classifier C_k^B, samples with $y_i = k$ are defined as positive samples, while samples with $y_i \neq k$ are expressed as negative samples. As a result, the open-set classifier is overconfident due to focusing on labels consisting of information from single classes and ignoring the competing relationships of different known classes [34]. Above overconfidence is manifested in sharp category boundaries and in failures to generalize from the source domain to the target domain. In addition, the noise in the labels compounds the damaging effects of the overconfidence problem.

We attribute the reason to the inadequate competition of a single open-set classifier. Specifically, each open-set classifier completes only binary classification while neglecting to observe more diverse labels. As a result, the simple learning task guides the classifier to overfit and produce exceptionally sharp classification boundaries. Another important reason is that it is inconsistent with human common sense for open-set classifiers to consider only one known class to identify new classes. Humans need to judge classes not belonging to all known classes before they can identify them as new classes.

To this end, we propose the All-in-One (AIO) classifier $C^{\text{AIO}}(\cdot)$. The forward propagation of $C^{\text{AIO}}(\cdot)$ is

$$\mathcal{C}_{x_i} = \left\{c_{x_i}^k, \tilde{c}_{x_i}^k | k \in \mathcal{K}\right\} = \sigma\left(C^{\text{AIO}}\left(z_{x_i}\right)\right), \tag{6}$$

The $c_{x_i}^k$ and $\tilde{c}_{x_i}^k$ are the probability of x_i being identified as a known and unknown class by kth category, $\sum_k \left\{c_{x_i}^k + \tilde{c}_{x_i}^k\right\} = 1$. The $\sigma(\cdot)$ is a 'top_n softmax' function to ensure $\sum_{k\in\mathcal{T}^N}\{c_{x_i}^k + \tilde{c}_{x_i}^k\} = 1$, $\mathcal{T}^N$ is the top $N = 20$ item of $\mathcal{C}_{x_i}$. The AIO classifier assigns two output neurons to each known category, representing belonging to the known or unknown class of the category respectively.

We propose two principles for designing an intuitive UNDA classifier to train the AIO classifier to solve the dilemma in the previous section.

- (a) If the classifier assigns the sample x_i to a known class y^s, it needs to make sure that the sample does not belong to other known classes $c_{x_i}^{y^s} > \max\{c_{x_i}^k\}_{k\in\mathcal{K}/y^s}$, and does not belong to an unknown class, $c_{x_i}^{y^s} > \{\tilde{c}_{x_i}^k\}_{k\in\mathcal{K}}$.
- (b) If the classifier assigns the sample x_i to an unknown class, it needs to confirm that the sample does not belong to all known classes, $\max\{\tilde{c}_{x_i}^k\}_{k\in\mathcal{K}} > \max\{c_{x_i}^k\}_{k\in\mathcal{K}}$.

Next, we combine the two principles to obtain the following objective. For a sample of the source domain,

$$c_{x_i}^{y^s} > \max\{\tilde{c}_{x_i}^k\}_{k\in\mathcal{K}} > \max\{c_{x_i}^k\}_{k\in\mathcal{K}/y^s}, \tag{7}$$

Based on Eq. (7), we formulate the loss function as,

$$\begin{aligned} L_{\text{AIO}}(x^s, y^s) = -[&log(c_{x_i}^{y^s}) + \min\{log(\tilde{c}_x^k)\}_{k\in\mathcal{K}/y^s} \\ &+ log\left(c_{x_i}^{y^s} - \max\{\tilde{c}_{x_i}^k\}_{k\in\mathcal{K}}\right)], \end{aligned} \tag{8}$$

The first and second terms of L_{AIO} increase $c_{x_i}^{y^s}$ and $\{\tilde{c}_x^k\}_{k\in\mathcal{K}/y^s}$, thus guarantee that they have sufficiently positive predictions and are larger than $\{c_{x_i}^k\}_{k\in\mathcal{K}/y^s}$. Also, the third term guarantees that $c_{x_i}^{y^s} > \max\{\tilde{c}_{x_i}^k\}_{k\in\mathcal{K}}$. Implicitly, $\{c_{x_i}^k\}_{k\in\mathcal{K}/y^s}$ is guided to have the lowest activation.

3.3. Learning & Inference

Learning. We combine the SCL loss and AIO loss to train the SAN. The overall loss is

$$\begin{aligned} &\mathcal{L}_{all} = E_{(\mathbf{x}_i^s, {y_i}^s)\sim\mathcal{D}_s}\mathcal{L}_{src}(\mathbf{x}_i^s, {y_i}^s) + \lambda E_{\mathbf{x}_i^t\sim\mathcal{D}_t}\mathcal{L}_{scl}(\mathbf{x}_i^t), \\ &\mathcal{L}_{src}(\mathbf{x}_i^s, {y_i}^s) = \mathcal{L}_{ce}(\mathbf{x}_i^s, {y_i}^s) + \beta\mathcal{L}_{AIO}(\mathbf{x}_i^s, {y_i}^s). \end{aligned} \tag{9}$$

The parameters of networks are optimized to minimize the loss. Note that λ and β are the weighted hyperparameters. This method is much more straightforward than existing ODA and UNDA methods [2, 11, 25], all of which require setting the threshold manually and/or multiple training phases.

Inference. If $c_{x_i}^k$ of the AIO classifier achieves the maximum value, then the sample is identified as a known class k. On the other hand, if any of $\{\tilde{c}_{x_i}^k\}_{k\in\mathcal{K}}$ of the AIO classifier achieves the maximum value, then the sample is identified as an unknown class.

4. Results

We evaluate our method in ODA and UNDA settings along with ablation studies. In addition, we assess the robustness with respect to the change of unknown target category size by varying the number of unknown categories.

Datasets. We utilize popular datasets in DA: Office [24], OfficeHome [31], VisDA [23], and DomainNet [22]. Unless otherwise noted, we follow existing protocols [26] to split the datasets into source-private ($|L_s - L_t|$), target-private ($|L_t - L_s|$) and shared categories ($|L_s \cap L_t|$).

Baselines. We aim to compare methods of universal domain adaptation (UNDA), which can reject unknown samples, such as, CMU [11], DANCE [25], DCC [18],

Table 1. **H-score comparison of Office and DomainNet datasets in the UNDA setting.** Single SAN indicates that uniform settings are used, and SAN* indicates selecting the best hyperpatameters for each setting using the grid search. **Bolded** means best performance, underlined means 2% better than other methods. The brackets after the dataset indicate ($|L_s - L_t|$, $|L_t - L_s|$, $|L_s \cap L_t|$).

Method	REF	Office (10 / 10 / 11)						Avg	DomainNet (150 / 50 / 145))						Avg
		A2D	A2W	D2A	D2W	W2D	W2A		P2R	R2P	P2S	S2P	R2S	S2R	
DANCE	NeurIPS2020 [25]	78.6	71.5	79.9	91.4	87.9	72.2	80.3	21.0	47.3	37.0	27.7	46.7	21.0	33.5
DCC	CVPR2021 [18]	88.5	78.5	70.2	79.3	88.6	75.9	80.2	56.9	50.3	43.7	44.9	43.3	56.2	49.2
USFDA	CVPR2020 [34]	85.5	79.8	83.2	90.6	88.7	81.2	84.8	—	—	—	—	—	—	—
OVANet	ICCV2021 [26]	85.8	79.4	80.1	95.4	94.3	84.0	86.5	56.0	51.7	47.1	47.4	44.9	57.2	50.7
TNT	AAAI2022 [5]	85.7	80.4	83.8	92.0	91.2	79.1	85.4	—	—	—	—	—	—	—
GATE	CVPR2022 [6]	87.7	81.6	84.2	94.8	94.1	83.4	87.6	—	—	—	—	—	—	—
D+SPA	NeurIPS2022 [16]	**90.4**	83.8	83.1	90.5	88.6	86.5	87.2	**59.1**	52.7	47.6	45.4	46.9	56.7	51.4
SAN	ours	89.9	**87.6**	**87.6**	**98.4**	**98.3**	89.0	**91.8**	57.4	**52.9**	**47.9**	**48.2**	**47.0**	57.9	**52.0**
SAN*	ours	**90.4**	**89.9**	**87.8**	**98.9**	**98.3**	**95.6**	**93.5**	57.8	**52.9**	**47.9**	**48.4**	**47.2**	57.9	**52.1**

Table 2. **H-score comparison of OfficeHome datasets in the UNDA setting.** D+SPA means DCC+SPA. Single SAN indicates that uniform settings are used, and SAN* indicates selecting the best hyperparameters for each setting using the grid search. **Bolded** means best performance, underlined means 2% better than others. The brackets after the dataset indicate ($|L_s - L_t|$, $|L_t - L_s|$, $|L_s \cap L_t|$).

Method	REF	OfficeHome (10 / 5 / 50)												Avg
		A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	
DANCE	NeurIPS2020 [25]	61.0	60.4	64.9	65.7	58.8	61.8	73.1	61.2	66.6	67.7	62.4	63.7	63.9
DCC	CVPR2021 [18]	58.0	54.1	58.0	74.6	70.6	77.5	64.3	73.6	74.9	**81.0**	75.1	80.4	70.2
OVANet	ICCV2021 [26]	62.8	75.6	78.6	70.7	68.8	75.0	71.3	58.6	80.5	76.1	64.1	78.9	71.8
TNT	AAAI2022 [5]	61.9	74.6	80.2	73.5	71.4	79.6	74.2	69.5	82.7	77.3	70.1	81.2	74.7
GATE	CVPR2022 [6]	63.8	75.9	81.4	74.0	72.1	79.8	74.7	70.3	82.7	79.1	71.5	**81.7**	75.6
D+SPA	NeurIPS2022 [16]	59.3	79.5	81.5	**74.7**	71.7	**82.0**	68.0	**74.7**	75.8	74.5	**75.8**	81.3	74.9
SAN	ours	**66.7**	79.4	**86.6**	73.2	**73.0**	79.5	**75.7**	64.0	82.6	79.4	66.8	80.0	**75.9**
SAN*	ours	**68.2**	**80.6**	**86.7**	73.4	**73.0**	79.8	**76.5**	64.9	**83.3**	80.1	67.1	80.1	**76.1**

OVANet [26], TNT [5], GATE [6] and D+SPA [16]. We are looking at some contemporaneous work such as KUADA [34], UACP [33] and UEPS [35], which we did not include in the comparison because the source code was not available and some of these works were not peer-reviewed. Instead of reproducing the results of these papers, we directly used the results reported in the papers with the same configuration.

Implementation. Following previous works, such as OVANet [26] and GATE [6], we employ ResNet50 [13] pretrained on ImageNet [10] as our backbone network. We train our models with inverse learning rate decay scheduling. The performance of the proposed SAN in uniform settings is listed in the penultimate row of the table. A grid hyperparameter search is performed for each setup, and the optimal results obtained are marked with *. The selected hyperparameters for searching include λ, β, and α. For all experiments, $\nu^y = 100$ and $\nu^z = 10$. The network $H(\cdot)$ uses a two-layer MLP network with 2048 neurons. In summary, our method outperforms or is comparable to the baseline method in all different settings. More details of the implementation are in the Appendix.

Evaluation Metric. The H-score is usually used to evaluate standard or ODA methods because it considers the trade-off between the accuracy of known and unknown classes [2]. H-score is the harmonic mean of the accuracy on common classes A_c and the accuracy on "unknown" classes A_t, H-score $= (2A_c \cdot A_t)/(A_c + A_t)$. The evaluation metric is high only when both the A_c and A_t are high. So, H-score can measure both accuracies of UNDA methods well. However, we find concerns about the fairness of the Hscore when the sample sizes of the known and unknown classes of the dataset differ significantly. For exam-

Table 3. **H-score of Office datasets in the ODA setting.**

Method	Office (10 / 0 / 11)						Avg
	A2D	A2W	D2A	D2W	W2D	W2A	
DANCE	84.9	78.8	79.1	78.8	88.9	68.3	79.8
DCC	58.3	54.8	67.2	89.4	80.9	85.3	72.6
USFDA	85.5	79.8	83.2	90.6	88.7	81.2	84.8
OVANet	**90.5**	88.3	86.7	98.2	98.4	88.3	91.7
TNT	85.8	82.3	80.7	91.2	96.2	81.5	86.3
GATE	88.4	86.5	84.2	95.0	96.7	86.1	89.5
D+SPA	92.3	91.7	90.0	96.0	97.4	91.5	93.2
SAN	**90.5**	**93.5**	91.7	**98.9**	**100**	**92.8**	**94.6**
SAN*	**90.5**	**93.8**	**92.7**	**99.3**	**100**	**93.7**	**95.0**

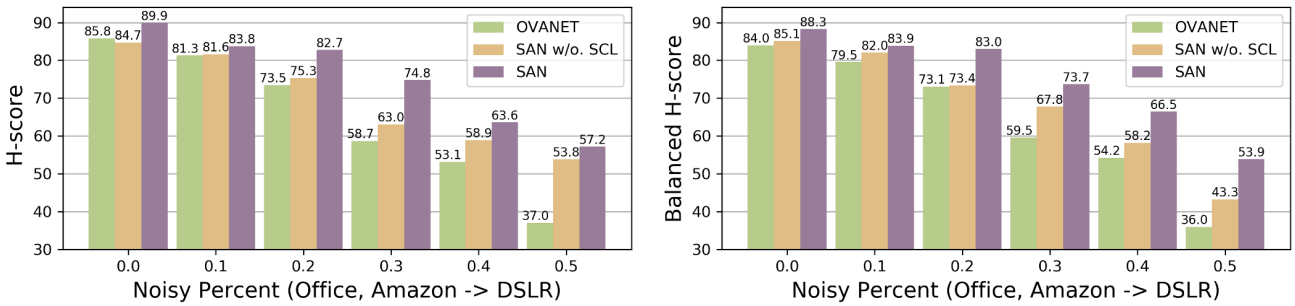

Figure 3. **Ablation study. OVANet v.s. SAN w/o. SCL v.s. SAN.** H-score and Balance H-score Comparisons of Office datasets in the UNDA setting. The horizontal coordinate indicates the addition of a specified percentage of noise to the original domain, and the vertical coordinate indicates the performance of the different methods.

Table 4. **H-score of VisDA datasets in UNDA and ODA setting.**

Method	VisDA ODA (6 / 0 / 6)	VisDA UNDA (6 / 3 / 3)
CMU	54.2	34.6
DANCE	67.5	42.8
DCC	59.6	43.0
OVANet	66.1	53.1
TNT	71.6	55.3
GATE	70.8	56.4
SAN	**72.0**	**60.1**

ple, when the number of samples in the unknown category is much larger than the known category (e.g., the OfficeHome dataset), pairing one more sample from the known category leads A_c to increase more significantly than the unknown category. Moreover, if A_c increases, the H-score will greatly increase, which leads to unfairness about the known and unknown categories. So, to achieve a higher h-score, the model will sacrifice the unknown category's accuracy to exchange for the common category's accuracy, which is unfair and impracticable in the real world. Therefore, inspired by the idea of Weighted Harmonic Means [14], we propose the Balance H-score as a more equitable metric (the proof is shown in Appendix B). For the dataset where the number of unknown categories in the sample is θ times the number of common, we define Balence H-score $= (1+\theta)A_c \cdot A_t/(\theta A_c + A_t)$. This paper selects the Hscore as an evaluation metric for convenient comparison with the baseline approach. Meanwhile, the Balance H-score is used in the more profound analysis of the relative advantages of the proposed method.

Performance comparisons on UNDA setting. From the results in Table 1, Table 2, and Table 4, SAN achieves a new state-of-the-art (SOTA) on all four datasets in the most challenging UNDA setting. Concerning H-score, SAN outperforms the previous SOTA UNDA method on Office by 4.2% and on OfficeHome by 0.3%. On large-scale datasets, SAN also gives more than 0.6% improvement on DomainNet and more than 3.7% on VisDA compared to all other

Table 5. **Ablation study.** H-score comparison on all four datasets in UNDA setting.

Method	Office (10/10/11)	OfficeHome (10/5/50)	DomainNet (150/50/145)	VisDA (6/3/3)
SAN	**91.8**	**75.9**	**52.0**	**60.1**
w/o. AIO	90.5	74.6	51.0	57.2
w/o. SCL	78.9	73.0	50.7	55.2
w. CL	78.2	73.3	50.5	52.7
OVANet	77.9	71.8	50.7	53.1

methods in terms of H-score. In VisDA and DomainNet, the number of samples and/or classes differs greatly from those of Office and OfficeHome.

Performance Comparisons on ODA setting. For the ODA setting, the H-score comparison results are presented in Table 3 and Table 4. Our method performs better than all the UNDA baselines on Office and VisDA datasets, with 1.4% and 1.2% H-score improvement.

Overview of Results. Under these two scenarios with "unknown" samples, SAN shows a more robust capability for separating common and private categories, which benefits from the SCL loss function and AIO classifier. Compared with GATE, a previous SOTA method tailored for the ODA setting, SAN is also superior on all datasets. This evidence shows that SAN gains a better trade-off between common categories classification and private samples identification.

4.1. Analysis in Universal Domain Adaptation

Ablation study, the effect of SCL Loss. We conduct controlled experiments to verify the necessity of the soft contrastive learning (SCL) Loss on all four datasets in UNDA settings, and the results are shown in Table 5. The *SAN* shows the performance of the proposed method. The *w/o. SCL* shows the performance of the SCL loss with $\mathcal{L}_{SCL}(\mathbf{x}_i^t, {y_i}^t)$ removed from the overall loss of SAN. The *w. CL* shows the performance of the SCL loss with $\mathcal{L}_{SCL}(\mathbf{x}_i^t, {y_i}^t)$ replaced by the CL loss L_{CL} in Eq. (1). The *OVANet* shows the performance of OVANet. The above experiences indicate that the SCL Loss significantly outper-

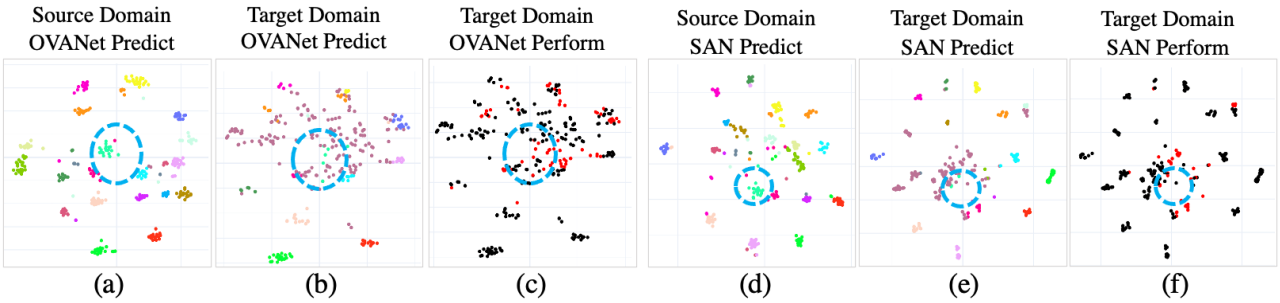

Figure 4. **Feature Visualization, OVANet V.S. SAN.** The (a) to (f) show the t-SNE visualization of embeddings produced by the backbone network of OVANet and SAN. The (a), (b), (d) and (e) are coloured by the predicted value label. In (c) and (f), the error samples (concentrate on know/unknown classification) are shown in red. The blue circle marks a private cluster. We find that the OVANet make more mistakes, and SAN is better. We attribute this improvement to the fact that SAN overcame the problem of overconfidence.

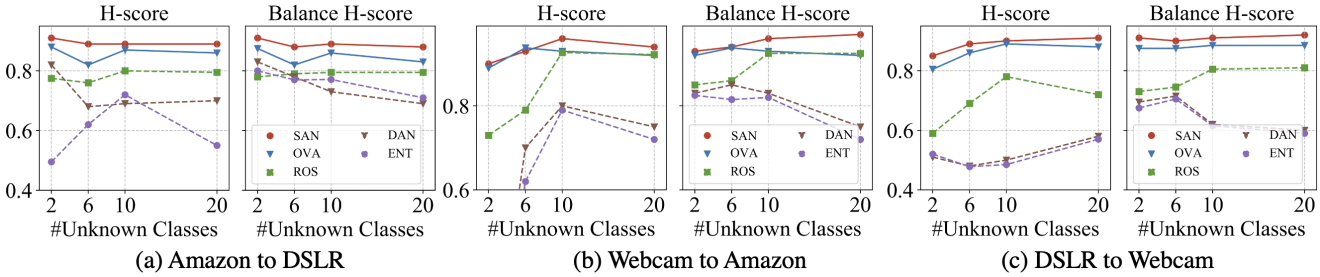

Figure 5. **H-score and Balance H-score comparison of Office dataset in ODA.** We vary the number of unknown classes using Office ($L_s \cap L_t| = 10$, $|L_s - L_t| = 0$). The left and right parts, respectively, show H-score and Balance H-score. SAN shows stable performance across different unknown class numbers, while baselines degrade performance in some settings.

forms typical CL loss. We attribute the failure of L_{CL} to the fact that the view noise caused by domain bias cannot be ignored. In addition, SCL loss can better alleviate this problem.

Ablation study, the effect of AIO classifier. We further conduct controlled experiments to verify the necessity of the All in one (AIO) classifier. In Table. 5, the *w/o. AIO* shows the performance of the AIO classifier replaced by the open-set and closed-set classifier. The control experiments on all four datasets indicate that the AIO Classifier brings improvements. The improvement from the AIO classifier is not as significant as that from SCL, probably because the label noise in the dataset is not significant. We further verify this idea by manually adding some label noise, and the experiment results are shown in Fig. 3. The results show that the *SAN* and *SAN w/o. SCL* exceed the baseline more significantly as the proportion of noise increases.

The overconfidence problem and its mitigation by SAN. Many current approaches are based on a combination of open-set classifiers and closed-set classifiers. We consider that they fail to achieve further improvements because the strategy of open-set classifiers leads to overconfidence. One direct evidence is that SAN achieves a more significant advantage in datasets with fewer samples (e.g., Office). To explore the adverse effects of overconfidence, we perform a visual analysis of the A2D setting of the Office dataset in Fig. 4. We find that the open set classifier of OVANet make mistake in some private class (e.g. the orange scatters). It can be attributed to the overconfidence classifier causing the over-sharp classification boundaries thus wrong testing results are output. We consider this is the result of overconfidence. Contrastingly, the same class is handled well by SAN.

The effect of the proportion of unknown samples on H-score, and the advantage of SAN on Balance H-score. H-score introduces fairness bias if there is a large quantitative difference between the sample size of unknown and known. To explore the fairness of the H-score, we changed the number of unknown classes in the target domain and then tested the performance of the H-score and balance H-score (in Fig. 5). The results show that changing the number of unknown classes dramatically changes the H-score. In contrast, the balance H-score exhibits higher stability. This suggests that the balance H-score is a more stable indicator for the proportion of unknown class samples. Its fairness is demonstrated in Appendix B. In addition, Fig.3 and Fig.5 show that the proposed SAN has more evident advantages in both the H-score and Balance H-score.

5. Conclusion

ODA and UNDA tasks aim to transfer the knowledge learned from a label-rich source domain to a label-scarce target domain without any constraints on the label space. In this paper, to solve the view-noise problem of augmentation-baed CL and the overconfidence problem of

novel category classifier, a Soft-contrastive All-in-one Network (SAN) is proposed. SAN includes SCL loss which can avoid the over-response of typical CL loss and enable data augmentation to improve the performance of ODA and UNDA. In addition, SAN includes an all-in-one (AIO) classifier to improve the robustness of the network. Extensive experimental results on UNDA and ODA demonstrate the advantages of SAN over existing methods.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010. 1

[2] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation. *Eur. Conf. Comput. Vis.*, pages 422–438, 2020. 3, 5, 6

[3] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. *Int. Conf. Comput. Vis.*, 2017. 1

[4] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. *Eur. Conf. Comput. Vis.*, 2018. 1

[5] Liang Chen, Yihang Lou, Jianzhong He, Tao Bai, and Minghua Deng. Evidential neighborhood contrastive learning for universal domain adaptation. *AAAI*, 2022. 1, 3, 6, 17

[6] Liang Chen, Yihang Lou, Jianzhong He, Tao Bai, and Minghua Deng. Geometric anchor correspondence mining with uncertainty modeling for universal domain adaptation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 3, 6, 17

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *international conference on machine learning*, 2020. 1, 3

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *computer vision and pattern recognition*, 2020. 3

[9] Ching-Yao Chuang, R Devon Hjelm, Xin Wang, Vibhav Vineet, Neel Joshi, Antonio Torralba, Stefanie Jegelka, and Yale Song. Robust contrastive learning against noisy views. *CVPR*, 2022. 1

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009. 6, 17

[11] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. *Eur. Conf. Comput. Vis.*, 2020. 3, 5, 17

[12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *Int. Conf. Mach Learn.*, 2014. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 6, 17

[14] S Kanas. Weighted harmonic means. *Complex Analysis and Operator Theory*, 11(8):1715–1728, 2017. 7

[15] S Sathiya Keerthi and Chih-Jen Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003. 4

[16] Jogendra Nath Kundu, Suvaansh Bhambri, Akshay Kulkarni, Hiran Sarkar, Varun Jampani, and R. Venkatesh Babu. Subsidiary prototype alignment for universal domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 6, 17

[17] Bolian Li, Zongbo Han, Haining Li, Huazhu Fu, and Changqing Zhang. Trustworthy long-tailed classification. *arXiv: Learning*, 2021. 4

[18] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. *computer vision and pattern recognition*, 2021. 5, 6, 17

[19] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *Int. Conf. Mach Learn.*, 2015. 3

[20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11):2579–2605, 2008. 4

[21] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. *Int. Conf. Comput. Vis.*, 2017. 3

[22] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. *Int. Conf. Comput. Vis.*, 2019. 5, 17

[23] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 5, 17

[24] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. *Eur. Conf. Comput. Vis.*, 2010. 3, 5, 17

[25] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *arXiv preprint arXiv:2002.07953*, 2020. 1, 3, 5, 6, 17

[26] Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. *arXiv: Computer Vision and Pattern Recognition*, 2021. 2, 5, 6, 17

[27] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. *Eur. Conf. Comput. Vis.*, 2018. 1, 3

[28] Huazhu Song, Zichun Ding, Cuicui Guo, Zhe Li, and Hongxia Xia. Research on combination kernel function of support vector machine. In *2008 International conference on computer science and software engineering*, volume 1, pages 838–841. IEEE, 2008. 4

[29] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 3

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv: Learning*, 2018. 3, 4

[31] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 5, 17

[32] Xudong Wang, Ziwei Liu, and Stella X. Yu. Unsupervised feature learning by cross-level instance-group discrimination. *computer vision and pattern recognition*, 2021. 3

[33] Yunyun Wang, Yao Liu, and Songcan Chen. Towards adaptive unknown authentication for universal domain adaptation by classifier paradox. 2022. 6, 17

[34] Yifan Wang, Lin Zhang, Ran Song, Lin Ma, and Wei Zhang. Exploiting inter-sample affinity for knowability-aware universal domain adaptation. 2022. 2, 3, 5, 6, 17

[35] Yifan Wang, Lin Zhang, Ran Song, Lin Ma, and Wei Zhang. A novel framework based on unknown estimation via principal sub-space for universal domain adaption. 2022. 6, 17

[36] Xiao, Liu, Fanjin, Zhang, Zhenyu, Hou, Zhaoyu, Wang, Li, Mian, Jing, Jie, and Tang. Self-supervised learning: Generative or contrastive. 2020. 3

[37] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1, 3

[38] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Self-labeling framework for novel category discovery over domains. *AAAI*, 2022. 1

[39] Zelin Zang, Siyuan Li, Di Wu, Ge Wang, Lei Shang, Baigui Sun, Hao Li, and Stan Z. Li. Dlme: Deep local-flatness manifold embedding. 2022. 1, 4

[40] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 1

Supplementary Materials: Boosting Novel Category Discovery Over Domains with Soft Contrastive Learning and All-in-One Classifier

Anonymous CVPR submission

Paper ID 5598

A. Details of SCL loss

A.1. Details of the transformation from Eq. (1) to Eq. (2)

We start with $L_{\text{CL}} = -\log \frac{\exp(S(z_i,z_j))}{\sum_{k=1}^{N_K}\exp(S(z_i,z_k))}$ (Eq. (1)), then

$$L_{\text{CL}} = \log N_K - \log \frac{\exp(S(z_i,z_j))}{\frac{1}{N_K}\sum_{k=1}^{N_K}\exp(S(z_i,z_k))}.$$

We are only concerned with the second term that has the gradient. Let (i,j) are positive pair and $(i,k_1),\cdots,(i,k_N)$ are negative pairs. The overall loss associated with point i is:

$$\begin{aligned}
&-\log \frac{\exp(S(z_i,z_j))}{\frac{1}{N_K}\sum_{k=1}^{N_K}\exp(S(z_i,z_k))} \\
=&-\left[\log\exp(S(z_i,z_j)) - \log\frac{1}{N_K}\sum_{k=1}^{N_K}\exp(S(z_i,z_k))\right] \\
=&-\left[\log\exp(S(z_i,z_j)) - \sum_{k=1}^{N_K}\log\exp(S(z_i,z_k)) + \sum_{k=1}^{N_K}\log\exp(S(z_i,z_k)) - \log\frac{1}{N_K}\sum_{k=1}^{N_K}\exp(S(z_i,z_k))\right] \\
=&-\left[\log\exp(S(z_i,z_j)) - \sum_{k=1}^{N_K}\log\exp(S(z_i,z_k)) + \log\Pi_{k=1}^{N_K}\exp(S(z_i,z_k)) - \log\frac{1}{N_K}\sum_{k=1}^{N_K}\exp(S(z_i,z_k))\right] \\
=&-\left[\log\exp(S(z_i,z_j)) - \sum_{k=1}^{N_K}\log\exp(S(z_i,z_k)) + \log\frac{\Pi_{k=1}^{N_K}\exp(S(z_i,z_k))}{\frac{1}{N_K}\sum_{k=1}^{N_K}\exp(S(z_i,z_k))}\right]
\end{aligned}$$

We focus on the case where the similarity is normalized, $S(z_i,z_k)\in[0,1]$. The data i and data k is the negative samples, then $S(z_i,z_k)$ is near to 0, $\exp(S(z_i,z_k))$ is near to 1, thus the $\frac{\Pi_{k=1}^{N_K}\exp(S(z_i,z_k))}{\frac{1}{N}\sum_{k=1}^{N_K}\exp(S(z_i,z_k))}$ is near to 1, and $\log\frac{\Pi_{k=1}^{N_K}\exp(S(z_i,z_k))}{\frac{1}{N}\sum_{k=1}^{N_K}\exp(S(z_i,z_k))}$ near to 0. We have

$$L_{\text{CL}} \approx -\left[\log\exp(S(z_i,z_j)) - \sum_{k=1}^{N_K}\log\exp(S(z_i,z_k))\right]$$

We denote ij and ik by a uniform index and use $\mathcal{H}_{ij}$ to denote the homology relation of ij.

$$
\begin{aligned}
L_{\mathrm{CL}} &\approx -\left[\log \exp(S(z_i, z_j)) - \sum_{k=1}^{N_K} \log \exp(S(z_i, z_k))\right] \\
&\approx -\left[\mathcal{H}_{ij} \log \exp(S(z_i, z_j)) - \sum_{j=1}^{N_K} (1 - \mathcal{H}_{ij}) \log \exp(S(z_i, z_j))\right] \\
&\approx -\left[\sum_{j=1}^{N_K+1} \{\mathcal{H}_{ij} \log \exp(S(z_i, z_j)) + (1 - \mathcal{H}_{ij}) \log\{\exp(-S(z_i, z_j))\}\}\right]
\end{aligned}
$$

we define the similarity of data i and data j as $Q_{ij} = \exp(S(z_i, z_j))$ and the dissimilarity of data i and data j as $\dot{Q}_{ij} = \exp(-S(z_i, z_j))$.

$$
L_{\mathrm{CL}} \approx -\left[\sum_{j=1}^{N_K+1} \left\{\mathcal{H}_{ij} \log Q_{ij} + (1 - \mathcal{H}_{ij}) \log \dot{Q}_{ij}\right\}\right]
$$

A.2. The proposed SCL loss is a smoother CL loss

This proof tries to indicate that the proposed SCL loss is a smoother CL loss. We discuss the differences by comparing the two losses to prove this point. the forward propagation of the network is, $z_i = H(\hat{z}_i), \hat{z}_i = F(x_i)$, $z_j = H(\hat{z}_j), \hat{z}_j = F(x_j)$. We found that we mix y and $\hat{z}$ in the main text, and we will correct this in the new version. So, in this section $z_i = H(y_i), y_i = F(x_i)$, $z_j = H(y_j), y_j = F(x_j)$ is also correct.

Let $H(\cdot)$ satisfy K-Lipschitz continuity, then $d^z_{ij} = k^* d^y_{ij}, k^* \in [1/K, K]$, where k^* is a Lipschitz constant. The difference between L_{SCL} loss and L_{CL} loss is,

$$
L_{\mathrm{CL}} - L_{\mathrm{SCL}} \approx \sum_j \left[\left(\mathcal{H}_{ij} - [1 + (e^{\alpha} - 1)\mathcal{H}_{ij}]\kappa\left(d^y_{ij}\right)\right) \log\left(\frac{1}{\kappa\left(d^z_{ij}\right)} - 1\right)\right]. \tag{10}
$$

Because the $\alpha > 0$, the proposed SCL loss is the soft version of the CL loss. if $\mathcal{H}_{ij} = 1$, we have:

$$
(L_{\mathrm{CL}} - L_{\mathrm{SCL}})|_{\mathcal{H}_{ij}=1} = \sum\left[\left((1 - e^{\alpha})\kappa\left(k^* d^z_{ij}\right)\right) \log\left(\frac{1}{\kappa\left(d^z_{ij}\right)} - 1\right)\right] \tag{11}
$$

then:

$$
\lim_{\alpha \to 0} (L_{\mathrm{CL}} - L_{\mathrm{SCL}})|_{\mathcal{H}_{ij}=1} = \lim_{\alpha \to 0} \sum\left[\left((1 - e^{\alpha})\kappa\left(k^* d^z_{ij}\right)\right) \log\left(\frac{1}{\kappa\left(d^z_{ij}\right)} - 1\right)\right] = 0 \tag{12}
$$

Based on Eq.(12), we find that if i, j is neighbor ($\mathcal{H}_{ij} = 1$) and $\alpha \to 0$, there is no difference between the CL loss L_{CL} and SCL loss L_{SCL}. When if $\mathcal{H}_{ij} = 0$, the difference between the loss functions will be the function of d^z_{ij}. The CL loss L_{CL} only minimizes the distance between adjacent nodes and does not maintain any structural information. The proposed SCL loss considers the knowledge both comes from the output of the current bottleneck and data augmentation, thus less affected by view noise.

Details of Eq. (10). Due to the very similar gradient direction, we assume $\dot{Q}_{ij} = 1 - Q_{ij}$. The contrastive learning loss is written as,

$$
L_{\mathrm{CL}} \approx -\sum \{\mathcal{H}_{ij} \log Q_{ij} + (1 - \mathcal{H}_{ij}) \log(1 - Q_{ij})\} \tag{13}
$$

where $\mathcal{H}_{ij}$ indicates whether i and j are augmented from the same original data.

The SCL loss is written as:

$$
L_{\mathrm{SCL}} = -\sum \{P_{ij} \log Q_{ij} + (1 - P_{ij}) \log(1 - Q_{ij})\} \tag{14}
$$

According to Eq. (4) and Eq. (5), we have

$$
\begin{aligned}
&P_{ij} = R_{ij}\kappa(d_{ij}^y) = R_{ij}\kappa(y_i, y_j), R_{ij} = \begin{cases} e^{\alpha} & \text{if } \mathcal{H}(x_i, x_j) = 1 \\ 1 & \text{otherwise} \end{cases}, \\
&Q_{ij} = \kappa(d_{ij}^z) = \kappa(z_i, z_j),
\end{aligned} \tag{15}
$$

For ease of writing, we use distance as the independent variable, $d_{ij}^y = \|y_i - y_j\|_2$, $d_{ij}^z = \|z_i - z_j\|_2$.
The difference between the two loss functions is:

$$
\begin{aligned}
&L_{\text{CL}} - L_{\text{SCL}} \\
&= -\sum \left[\mathcal{H}_{ij} \log \kappa\left(d_{ij}^z\right) + (1 - \mathcal{H}_{ij}) \log\left(1 - \kappa\left(d_{ij}^z\right)\right) - R_{ij}\kappa\left(d_{ij}^y\right) \log \kappa\left(d_{ij}^z\right) - \left(1 - R_{ij}\kappa\left(d_{ij}^y\right)\right) \log\left(1 - \kappa\left(d_{ij}^z\right)\right) \right] \\
&= -\sum \left[\left(\mathcal{H}_{ij} - R_{ij}\kappa\left(d_{ij}^y\right)\right) \log \kappa\left(d_{ij}^z\right) + \left(1 - \mathcal{H}_{ij} - 1 + R_{ij}\kappa\left(d_{ij}^y\right)\right) \log\left(1 - \kappa\left(d_{ij}^z\right)\right) \right] \\
&= -\sum \left[\left(\mathcal{H}_{ij} - R_{ij}\kappa\left(d_{ij}^y\right)\right) \log \kappa\left(d_{ij}^z\right) + \left(R_{ij}\kappa\left(d_{ij}^y\right) - \mathcal{H}_{ij}\right) \log\left(1 - \kappa\left(d_{ij}^z\right)\right) \right] \\
&= -\sum \left[\left(\mathcal{H}_{ij} - R_{ij}\kappa\left(d_{ij}^y\right)\right) \left(\log \kappa\left(d_{ij}^z\right) - \log\left(1 - \kappa\left(d_{ij}^z\right)\right)\right) \right] \\
&= \sum \left[\left(\mathcal{H}_{ij} - R_{ij}\kappa\left(d_{ij}^y\right)\right) \log\left(\frac{1}{\kappa\left(d_{ij}^z\right)} - 1\right) \right]
\end{aligned} \tag{16}
$$

Substituting the relationship between $\mathcal{H}_{ij}$ and R_{ij}, $R_{ij} = 1 + (e^{\alpha} - 1)\mathcal{H}_{ij}$, we have

$$
L_{\text{CL}} - L_{\text{SCL}} = \sum \left[\left(\mathcal{H}_{ij} - [1 + (e^{\alpha} - 1)\mathcal{H}_{ij}]\kappa\left(d_{ij}^y\right)\right) \log\left(\frac{1}{\kappa\left(d_{ij}^z\right)} - 1\right) \right] \tag{17}
$$

We assume that network $H(\cdot)$ to be a Lipschitz continuity function, then

$$
\frac{1}{K} H(d_{ij}^z) \leq d_{ij}^y \leq K H(d_{ij}^z) \quad \forall i, j \in \{1, 2, \cdots, N\} \tag{18}
$$

We construct the inverse mapping of $H(\cdot)$ to $H^{-1}(\cdot)$,

$$
\frac{1}{K} d_{ij}^z \leq d_{ij}^y \leq K d_{ij}^z \quad \forall i, j \in \{1, 2, \cdots, N\} \tag{19}
$$

and then there exists k^*:

$$
d_{ij}^y = k^* d_{ij}^z \quad k^* \in [1/K, K] \quad \forall i, j \in \{1, 2, \cdots, N\} \tag{20}
$$

Substituting the Eq.(20) into Eq.(17).

$$
L_{\text{CL}} - L_{\text{SCL}} = \sum \left[\left(\mathcal{H}_{ij} - [1 + (e^{\alpha} - 1)\mathcal{H}_{ij}]\kappa\left(k^* d_{ij}^z\right)\right) \log\left(\frac{1}{\kappa\left(d_{ij}^z\right)} - 1\right) \right] \tag{21}
$$

A.3. SCL is better than CL in view-noise

To demonstrate that compared to contrastive learning, the proposed SCL Loss has better results, we first define the signal-to-noise ratio (SNR) as an evaluation metric.

$$SNR = \frac{PL}{NL} \tag{22}$$

where PL means the expectation of positive pair loss, NL means the expectation of noisy pair loss.
This metric indicates the noise-robust of the model, and obviously, the bigger this metric is, the better.
In order to prove the soft contrastive learning's SNR is larger than contrastive learning's, we should prove:

$$\frac{PL_{cl}}{NL_{cl}} < \frac{PL_{scl}}{NL_{scl}} \tag{23}$$

Obviously, when it is the positive pair case, $S\left(z_i, z_j\right)$ is large if $H\left(x_i, x_j\right) = 1$ and small if $H\left(x_i, x_j\right) = 0$. Anyway, when it is the noisy pair case, $S\left(z_i, z_j\right)$ is small if $H\left(x_i, x_j\right) = 1$ and large if $H\left(x_i, x_j\right) = 0$.
First, we organize the PL_{scl} and PL_{cl} into 2 cases, $H\left(x_i, x_j\right) = 1$ and $H\left(x_i, x_j\right) = 0$, for writing convenience, we write $S\left(z_i, z_j\right)$ as S and S', respectively.

$$PL_{scl} = -M\left\{(1-S')\log\left(1-S'\right) + S'\log S'\right\} - \left\{(1-e^{\alpha}S)\log(1-S) + e^{\alpha}S\log S\right\} \tag{24}$$

$$PL_{cl} = -M\log\left(1-S'\right) - \log S \tag{25}$$

M is the ratio of the number of occurrences of $H = 1$ to $H = 0$. So, we could get:

$$\begin{aligned}
&PL_{scl} - PL_{cl} \\
&= -M\left\{(1-S'-1)\log\left(1-S'\right) + S'\log S'\right\} - \left\{(1-e^{\alpha}S)\log(1-S) + (e^{\alpha}S-1)\log S\right\} \\
&= -M\left\{S'\left(\log S' - \log\left(1-S'\right)\right)\right\} - \left\{(e^{\alpha}S-1)\left(\log S - \log(1-S)\right)\right\} \\
&= -M\left\{S' log\frac{S'}{(1-S')}\right\} - \left\{(e^{\alpha}S-1) log\frac{S}{(1-S)}\right\}
\end{aligned} \tag{26}$$

In the case of positive pair, S converges to 1 and S' converges to 0.
Because we have bounded that $e^{\alpha}S <= 1$, so we could easily get:

$$(e^{\alpha}S-1) log\frac{S}{(1-S)} <= 0 \tag{27}$$

Also, we could get:

$$-M\left\{S' log\frac{S'}{(1-S')}\right\} > 0 \tag{28}$$

So we get:

$$PL_{scl} - PL_{cl} > 0 \tag{29}$$

And for the case of noise pair, the values of S and S' are of opposite magnitude, so obviously, there is $NL_{scl} - NL_{cl} < 0$.
So the formula Eq. (23) has been proved.

B. Details of Balance Hscore

Inspired by the idea of Weighted Harmonic Means, the proposed Balance Hscore is,

$$\text{Balance Hscore} = B = \frac{1+\theta}{\frac{1}{A_c} + \frac{\theta}{A_t}} = \frac{A_t A_c}{A_t + \theta A_c}(1+\theta) \tag{30}$$

where θ is the ratio of unknown and known samples, The A_c is the accuracy of known classes, and A_t is the accuracy of unknown classes.

Why Balance Hscore is balance for known classes and unknown classes. To avoid sacrificing a category's accuracy in exchange for another category's accuracy, we assume that the change in the number of the correct categories and the number of the unknown categories has the same impact on the evaluation metric.

Let M be the number of the samples of known classes, and N_c be the number of the correct samples of known classes, with $A_c = N_c/M$. The impact of Balance Hscore from the known class is,

$$\begin{aligned} \frac{\partial B}{\partial N_c} &= \frac{\partial B}{\partial A_c} \cdot \frac{\partial A_c}{\partial N_c} \\ &= A_t(1+\theta) \cdot \frac{\theta A_c + A_t - \theta A_c}{(\theta A_c + A_t)^2} \cdot \frac{1}{M} \\ &= \frac{(1+\theta)A_t^2}{M(\theta A_c + A_t)^2} \end{aligned} \tag{31}$$

Let M_t be the number of the samples of known classes, and N_t be the number of the correct samples of known classes, with $A_t = N_t/M_t = N_t/(\theta M)$. The impact of a Balance Hscore from the unknown class is,

$$\begin{aligned} \frac{\partial B}{\partial N_t} &= \frac{\partial B}{\partial A_t} \cdot \frac{\partial A_t}{\partial N_t} \\ &= A_c(1+\theta) \cdot \frac{(\theta A_c + A_t) - A_t}{(\theta A_c + A_t)^2} \cdot \frac{1}{\theta M} = \frac{(1+\theta)A_c^2}{M(\theta A_c + At)^2} \end{aligned} \tag{32}$$

So if $A_c = A_t$, we have

$$\frac{\partial B}{\partial N_c} = \frac{\partial B}{\partial N_t},$$

it indicates that the metric gets the same influence as the correct classification. Thus the Balance Hscore is balance for known and unknown classes.

Why Hscore is unbalance for known classes and unknown classes. However, for the

$$\text{Hscore} = (2 \cdot A_c \cdot A_t)/(A_c + A_t).$$

The impact of the Hscore by the known class is

$$\begin{aligned} \frac{\partial H}{\partial N_c} &= \frac{\partial H}{\partial A_c} \cdot \frac{\partial A_c}{\partial N_c} \\ &= 2A_t \cdot \frac{A_t + A_c - A_c}{(A_c + A_t)^2} \cdot \frac{1}{M} \\ &= \frac{2A_t^2}{M(A_c + A_t)^2} \end{aligned} \tag{33}$$

The impact of the Hscore by the unknown class is

$$\begin{aligned} \frac{\partial H}{\partial N_t} &= \frac{\partial H}{\partial A_t} \cdot \frac{\partial A_t}{\partial N_t} \\ &= 2A_c \cdot \frac{A_c + A_t - A_t}{(A_c + A_t)^2} \cdot \frac{1}{\theta M} \\ &= \frac{2A_c^2}{\theta M(A_c t A_t)^2} \end{aligned} \tag{34}$$

So when $A_c = A_t$, we could get $\frac{\partial B}{\partial N_c} \neq \frac{\partial B}{\partial N_t}$, we think it's not balance.

C. Experimental setups

C.1. Baseline Methods

We aim to compare methods of universal domain adaptation (UNDA), which can reject unknown samples, such as, CMU [11], DANCE [25], DCC [18], OVANet [26], TNT [5], GATE [6] and D+SPA [16]. We are looking at some contemporaneous work such as KUADA [34], UACP [33] and UEPS [35], which we did not include in the comparison because the source code was not available and some of these works were not peer-reviewed. Instead of reproducing the results of these papers, we directly used the results reported in the papers with the same configuration.

C.2. Datasets

We utilize popular datasets in DA: Office [24], OfficeHome [31], VisDA [23], and DomainNet [22]. Unless otherwise noted, we follow existing protocols [26] to split the datasets into source-private ($|L_s - L_t|$), target-private ($|L_t - L_s|$) and shared categories ($|L_s \cap L_t|$).

Table 6. The division on label sets in each setting

Tasks	Datasets	$\|L_s \cap L_t\|$	$\|L_s - L_t\|$	$\|L_t - L_s\|$
ODA	Office-31	10	0	11
	VisDA	6	0	6
UNDA	Office-31	10	10	11
	Office-Home	10	5	50
	VisDA	6	3	3
	DomainNet	150	50	145

C.3. Hyperparameters

Following previous works, such as OVANet [26] and GATE [6], we employ ResNet50 [13] pre-trained on ImageNet [10] as our backbone network. We train our models with inverse learning rate decay scheduling. The performance of the proposed SAN in uniform settings is listed in the penultimate row of tables in the main paper. At the same time, we use a grid search to select the optimal hyperparameters for each setup, and the corresponding optimal results are marked with a *. The selected hyperparameters for searching include λ, β, α and learning rate. The grid search illustrates the proposed method without optimizing the hyperparameters for each setup. In summary, our method outperforms or is comparable to the baseline method in all different settings. For all experiments, $\nu^y = 100$ and $\nu^z = 10$. The network $H(\cdot)$ uses a two-layer MLP network with 2048 neurons.

The search space of hyperparameters in the grid search method are as follows.

Table 7. Hyperparameter search space

Hyperparameters	Search Space
α	[0.01, 0.02, 0.05, 0.1, 0.2]
λ	[0.1, 0.2, 0.3]
β	[0.1, 0.2, 0.3]
learning rate	[1e-2, 2e-2, 5e-2]

C.4. Top_n softmax in AIO

The forward propagation of $C^{\text{AIO}}(\cdot)$ is

$$\mathcal{C}_{x_i} = \left\{c_{x_i}^k, \tilde{c}_{x_i}^k | k \in \mathcal{K}\right\} = \sigma\left(C^{\text{AIO}}\left(z_{x_i}\right)\right), \tag{35}$$

The $c_{x_i}^k$ and $\tilde{c}_{x_i}^k$ are the probability of x_i being identified as a known and unknown class by kth category, $\sum_k \left\{c_{x_i}^k + \tilde{c}_{x_i}^k\right\} = 1$. The $\sigma(\cdot)$ is a 'top_n softmax' function to ensure $\sum_{k\in\mathcal{T}^N}\left\{c_{x_i}^k + \tilde{c}_{x_i}^k\right\} = 1$, $\mathcal{T}^N$ is the top $N = 20$ item of $\mathcal{C}_{x_i}$. We deploy

'top_n softmax' to balance the loss scale of different category numbers. For example, in UNDA setting, there are 200 known categories in DomainNet, while only 20 known categories in Office. If deploying a simple softmax, the loss scale will vary over a wide range with different datasets.