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

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.

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

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 $\dot{Q}_{ij} = \exp(-S(z_i, z_j))$.

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

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,

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

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

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

then:

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

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,

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

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

The SCL loss is written as:

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

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

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

For ease of writing, we use distance as the independent variable, $d^y_{ij} = \|y_i - y_j\|_2$, $d^z_{ij} = \|z_i - z_j\|_2$.

The difference between the two loss functions is:

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

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

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

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

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

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

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

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).

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

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 [26], DCC [19], OVANet [27], TNT [6], GATE [5] and D+SPA [17]. We are looking at some contemporaneous work such as KUADA [35], UACP [34] and UEPS [36], 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 [25], OfficeHome [32], VisDA [24], and DomainNet [23]. Unless otherwise noted, we follow existing protocols [27] 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. 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 $\sum_{k \in \mathcal{T}^N} \{c_{x_i}^k + \tilde{c}_{x_i}^k\} = 1$, $\mathcal{T}^N$ 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.