NSCL: NOISE-RESISTANT SOFT CONTRASTIVE LEARNING FOR UNIVERSAL DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

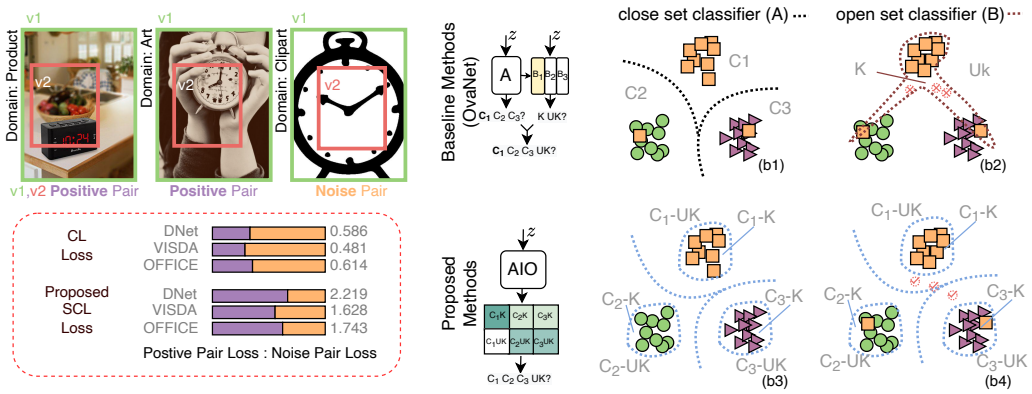
Domain adaptation (DA) transfers knowledge from label-rich domains to new domains where labels are scarce to address the problem of generalization of deep neural networks in new domains. Universal domain adaptation (UNDA) assumes the label distributions of labeled and unlabeled data are different and unknowable. In this paper, we concentrate on solving the noise problem on the UNDA problem based on contrastive learning (CL), which includes view noise in data augmentation and label noise in the classifier training. The domain differences in UNDA amplify the noise in the view of data augmentation, resulting in data augmentation schemes that apply to all domains being challenging to find. In addition, the mainstream UNDA classifiers combine closed-set classifiers with open-set classifiers; insufficient competition among open-set classifiers leads to overconfidence, which results in incredible sensitivity to noise in labeled data. Therefore, we propose Noise-Resistant Soft Contrastive Learning (NSCL) addresses the above issues. Firstly, we propose a soft contrast learning loss to avoid the over-response of typical CL loss to noisy samples, thus enabling data augmentation to improve the performance of UNDA further. Secondly, we design an all-in-one (AIO) classifier to improve the robustness of noisy labels while introducing multi-category unknown class competition. Extensive experimental results on UNDA and open-set DA demonstrate the advantages of NSCL over existing methods, especially in downstream tasks such as classification and visualization. [1]

1 INTRODUCTION

Domain adaptation (DA) transfers knowledge from label-rich training domains to new domains where labels are scarce (Ben-David et al., 2010), 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) (Busto & Gall, 2017; Saito et al., 2018); the source domain includes classes absent in the target (source private categories); i.e., partial DA (PDA) (Cao et al., 2018); 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 (You et al., 2019; Saito et al., 2020) 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 cannot know the difference in advance. The UNDA is a uniform and practical setup, since estimating the label distributions of unlabeled data is very hard in real applications.

Recently, Contrastive Learning (CL) has been used for unsupervised pre-training and has yielded excellent results in various downstream tasks. Furthermore, several studies (Yu et al., 2022; Chen et al., 2022a) have attempted to enhance UNDA by introducing CL. However, the widely used data augmentation-based CL schemes have not been introduced into UNDA. It is attributed to two noise-related factors (problems) that have limited the development of data-augmentation based CL on UNDA.

[1] code of the paper: https://anonymous.4open.science/r/nscl-A543/



(a) Noise in Data Augmentation for CL (b) Noise in UNDA Classifier Training

Figure 1: **The noise problem in UNDA and the solution proposed in 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) Label-noise problem in the classifier training. The dashed circle with a tick/cross means the test samples are classified correctly/incorrectly.

View-noise problem in data augmentation for CL. Although the view-noise in CL has started to receive attention in network pre-training (Chuang et al., 2022), we found that it causes more severe damage due to domain differences of UNDA. As shown at the top 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.

Label-noise problem in UNDA classifier training. Recently, OVANet and its variants (Saito & Saenko, 2021; Wang et al., 2022) have received much attention. These methods combine closed-set classifiers and open-set classifiers to identify known and unknown classes. However, these methods contain multiple open-set classifiers, each corresponding to a single known class. These methods combine closed-set classifier and open-set classifiers to identify known and unknown classes. However, each of the open-set classifiers only corresponds to a single known class. As a result, inadequate competition between open-set classifiers causes the classifier to be easier to fall into over-fitting and over-confidence. This eventually leads to failure in target domain recognition. When label-noise is present in the source domain, such imperfect data is almost inevitable, and the detrimental effects of overconfidence are magnified. As shown in 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 phenomenon, which eventually causes the target domain samples to be misclassified.

To solve the above view-noise and label-noise problem, we propose the noise-resistant soft contrastive learning (NSCL), a new UNDA method. **To solve the view-noise problem**, we design a soft contrastive learning (SCL) loss. The proposed SCL loss incorporates the idea of self-distillation, drawing partial knowledge from the training network to counteract the detrimental effects of the 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 label-noise problem**, we try to prevent the overconfidence problem of independent classifiers. Therefore, an all-in-one (AIO) classifier is designed to solve the UNDA uniformly. Furthermore, The AIO classifier is started from common sense. It assumes that *identifying a class as an unknown class requires determining that it does not belong to all known classes*. Underlying the above assumptions, we design a new loss function 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 UNDA and open-set DA benchmarks and vary the proportion of unknown classes. As a result, the proposed NSCL outperforms various baselines that explicitly or implicitly employ the ratio of unknown samples.

2 RELATED WORK

Domain Adaptation (DA). Unsupervised domain adaptation (UDA) (Saenko et al., 2010) 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 (Ganin & Lempitsky, 2014; Tzeng et al., 2017; Long et al., 2015). For ODA (Panareda Busto & Gall, 2017; Saito et al., 2018), 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, $|L_s - L_t| > 0, |L_t - L_s| > 0$. Saito et al. (2020) 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. You et al. (2019); Fu et al. (2020); Saito et al. (2020) compute a confidence score for known classes, and samples with a score lower than a threshold are regarded as unknown. Fu et al. (2020) seems to validate the threshold using labeled data, which is not a realistic solution. Bucci et al. (2020) set the mean of the confidence score as the threshold, which implicitly rejects about half of the target data as unknown. Saito et al. (2020) sets a threshold decided by the number of classes in the source, which does not always work well. Wang et al. (2022) 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 (Xiao et al., 2020), has achieved impressively superior performance in many computer vision tasks (Chen et al., 2020). 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 (Chen & He, 2020; Wang et al., 2021). Chen et al. (2022a) proposes to utilize mutual nearest neighbors as positive pairs to achieve feature alignment between the two domains. Chen et al. (2022b) 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 UNDA, we are given a source domain dataset $\mathcal{D}_s = \{(\mathbf{x}_i^s, \widehat{y}_i^s)\}_{i=1}^{N_s}$ and an 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 NSCL. 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 CONTRAST 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

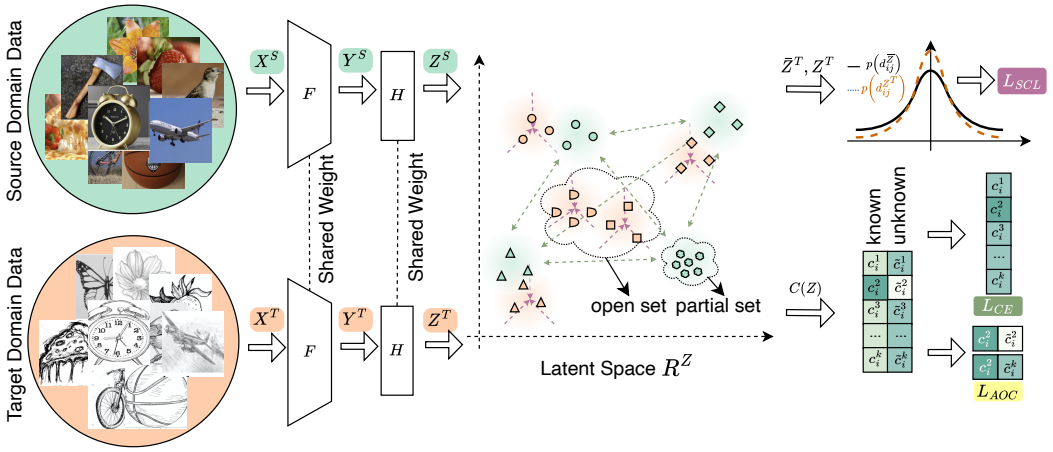


Figure 2: **Framework of NSCL.** The proposed method includes a backbone network $F(\cdot)$, a projection head network $H(\cdot)$, and an all-in-one (AIO) classifier $C(\cdot)$. The backbone network $F(\cdot)$ and projection head network $H(\cdot)$ map the source domain data $\mathbf{x}^s$ and the target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

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 (van den Oord et al., 2018):

$$L_{\text{CL}}\left(v_i, v_j, \{v_k\}_{k\in\{1,\cdots,N_K\}}\right) = -\log \frac{e^{z_i^T z_j}}{\sum_{k=1}^{N_K} e^{z_i^T z_k}} = -\log \frac{S(z_i, z_j)}{\sum_{k=1}^{N_K} S(z_i, z_k)} \tag{1}$$

where (v_i, v_j) is positive pair and (v_i, v_k) is negative pair, and z_i, z_j, z_k are 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 defended by cosine similarity $S_{ij} = \exp(z_i \cdot z_j)$. The latest approach (Li et al., 2021a; Zang et al., 2022), which redefines similarity based on long-tail kernel functions, aims to improve the discriminative performance of the potential space. Thus the t-distribution kernel function is used to calculate the pairwise similarity.

$$S(z_i, z_j, \nu) = S(d_{ij}^z, \nu) = \frac{\text{Gam}\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\text{Gam}\left(\frac{\nu}{2}\right)} \left(1 + \frac{d_{ij}^2}{\nu}\right)^{-\frac{\nu+1}{2}}, d_{ij}^z = d(z_i, z_j) \tag{2}$$

where $\text{Gam}(\cdot)$ is the Gamma function, d_{ij}^z is the euclidean distance between z^i and z^i. The degrees of freedom ν control the shape of the kernel function. A formal presentation of positive and negative sample pairs is considered to analyze the view noise problem. The details of the transformation from Eq. (1) to Eq. (3) are in appendix A.

$$L_{\text{CL}}(x_i, \{x_j\}_{k\in\{1,\cdots,N_J\}}) = -\sum_{j=1}\{(1-\mathcal{H}_{ij})\log(1 - S(z_i, z_j)) + \mathcal{H}_{ij}\log(S(z_i, z_j))\}, \tag{3}$$

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. 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, data augmentation schemes suitable 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 resort to the idea of self-distillation to obtain information from the neural network under training to predict the confidence level of the sample to the information provided. We propose soft contrast learning (SCL) loss,

$$L_{\text{SCL}}(x_i, \{x_j\}_{k\in\{1,\cdots,N_J\}}) = -\sum_{j=1}\{(1-\tilde{S}_{ij}^y(d_{ij}^y, \nu^y))\log\left(1-S(d_{ij}^z, \nu^z)\right)+\tilde{S}_{ij}^y(d_{ij}^y, \nu^y)\log\left(S(d_{ij}^z, \nu^z)\right)\} \tag{4}$$

where ν^y is the width of t-distribution in the latent space $\mathcal{R}^y$, where ν^z is the width of t-distribution in the latent space $\mathcal{R}^z$. The soften similarity $\tilde{S}(z_i, z_j, \nu)$ is calculated by the $\mathcal{H}_{ij}$ and the trustworthiness estimation,

$$\tilde{S}(d_{ij}^y, \nu) = \begin{cases} S(\alpha d_{ij}^y, \nu) & \text{if } \mathcal{H}(x_i, x_j) = 1 \\ S(d_{ij}^y, \nu) & \text{otherwise} \end{cases}, \tag{5}$$

where $\alpha \in [0, 1]$ is a soften hyper-parameters. We discuss the differences between SCL and CL loss in the appendix A.3, including that SCL is a smoother CL loss and that SCL loss can maintain a higher signal-to-noise ratio in the noise-view.

3.2 Label-noise and All in One (AIO) Classifier

Current advanced UNDA (Saito & Saenko, 2021; Wang et al., 2022) 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 $\mathcal{K} \in \{1, \cdots, N_K\}$ 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 over-confident due to it focusing on labels consisting of information from single classes and ignoring the competing relationships of different known classes (Wang et al., 2022). Above over-confidence is manifested in sharp categorical 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 not to belong to all known classes before they can identify new classes.

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

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

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 category k, $\sum_k \left\{c_{x_i}^k + \tilde{c}_{x_i}^k\right\} = 1$. The $\sigma(\cdot)$ is a 'top_n softmax' active 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 class and the unknown class, 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 data x_i to a known class y^s, it needs to make sure that it 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 data x_i to a unknown class, it needs to confirm that it 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 the above inequalities, we design the loss function as,

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

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 them have sufficiently positive predictions and 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 soft contrastive learning loss and AIO classifier to learn both open-set and closed-set categorization. The overall training loss is computed as follows:

$$\mathcal{L}_{all} = E_{(\mathbf{x}_i^s, {y_i}^s)\sim\mathcal{D}_s}\mathcal{L}_{src}(\mathbf{x}_i^s, {y_i}^s) + \lambda E_{\mathbf{x}_i^t\sim\mathcal{D}_t}\mathcal{L}_{scl}(\mathbf{x}_i^t), \tag{9}$$

$$\mathcal{L}_{src}(\mathbf{x}_i^s, {y_i}^s) = \mathcal{L}_{ce}(\mathbf{x}_i^s, {y_i}^s) + \beta\mathcal{L}_{AIO}(\mathbf{x}_i^s, {y_i}^s). \tag{10}$$

The parameters of networks are optimized to minimize the loss. Note that λ and β are the weighted parameters. This method is much more straightforward than existing ODA and UNDA methods (Saito et al., 2020; Fu et al., 2020; Bucci et al., 2020), all of which require setting the threshold manually and/or multiple training phases.

Inference. If the AIO classifier's $c_{x_i}^k$ 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 UNDA settings along with ablation studies. To evaluate the robustness of the change in the number of unknown target samples, we vary the number and compare it with other baselines.

Datasets. We utilize popular datasets in DA: Office (Saenko et al., 2010), Office-Home (Venkateswara et al., 2017), VisDA (Peng et al., 2017), and DomainNet (Peng et al., 2019). Unless otherwise noted, we follow existing protocols (Saito & Saenko, 2021) 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 (Fu et al., 2020), DANCE (Saito et al., 2020), DCC (Li et al., 2021b), OVANet (Saito & Saenko, 2021), TNT (Chen et al., 2022a),GATE (Chen et al., 2022b), KUADA (Wang et al., 2022). Instead of reproducing the results of these papers, we directly used the results reported in the papers with the same configuration. We focused on contemporaneous work, although the results are not included in the table because these articles were not peer-reviewed.

Implementation. Following previous works, such as OVANet (Saito & Saenko, 2021) and GATE(Chen et al., 2022b), we employ ResNet50 (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009) as our backbone network. We train our models with inverse learning rate decay scheduling. The performance of the proposed NSCL in uniform settings is listed in the penultimate row of the table. A grid search is performed for each setup, and the optimal values obtained are marked with *. The selected parameters 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

Method	Office (10 / 10 / 11)						Avg	DomainNet (150 / 50 / 145))						Avg
	A2D	A2W	D2A	D2W	W2D	W2A		P2R	R2P	P2S	S2P	R2S	S2R	
CMU	68.1	67.3	71.4	79.3	80.4	72.2	73.1	50.8	52.2	45.1	44.8	45.6	51.0	48.3
DANCE	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	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
ROS	71.4	71.3	81.0	94.6	95.3	79.2	82.1	—	—	—	—	—	—	—
USFDA	85.5	79.8	83.2	90.6	88.7	81.2	84.8	—	—	—	—	—	—	—
OVANet	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	85.7	80.4	83.8	92.0	91.2	79.1	85.4	—	—	—	—	—	—	—
GATE	87.7	81.6	84.2	94.8	94.1	83.4	87.6	—	—	—	—	—	—	—
KAUDA	88.9	83.0	81.1	94.5	**98.3**	85.2	88.5	**59.1**	52.4	47.5	48.1	45.1	**58.6**	51.8
NSCL	**89.9**	**87.6**	**87.6**	**98.4**	**98.3**	83.4	**91.8**	57.4	**52.9**	**47.9**	**48.2**	**47.0**	57.9	**52.0**
NSCL*	**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 1: **H-score comparison of Office and DomainNet datasets in the UNDA setting.** Single NSCL indicates that consistent settings are used, and NSCL* indicates selecting the best setting for each setting using the grid search. **Bolded** means best performance, underlined means 2% better than other methods.

Method	OfficeHome (10 / 5 / 50)												
	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg
CMU	56.0	56.9	59.1	66.9	64.2	67.8	54.7	51.0	66.3	68.2	57.8	69.7	61.6
DANCE	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	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	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	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	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
KAUDA	63.9	79.0	83.5	70.4	72.4	77.6	71.7	61.3	**83.6**	78.8	64.3	83.0	73.8
NSCL	**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**
NSCL*	**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**

Table 2: **H-score comparison of OfficeHome datasets in the UNDA setting. Single NSCL indicates that consistent settings are used, and NSCL* indicates selecting the best setting for each setting using the grid search. Bolded means best performance.**

outperforms or is comparable to the baseline method in all different settings. More details of the implementation are in the Appendix.

Method	Office (10 / 0 / 11)						Avg
	A2D	A2W	D2A	D2W	W2D	W2A	
CMU	52.6	55.7	76.5	75.9	64.7	65.8	73.1
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
ROS	82.4	82.1	77.9	96.0	99.7	77.2	85.9
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
KAUDA	89.4	85.6	92.4	94.5	90.5	92.2	90.8
NSCL	**90.5**	**93.5**	91.7	**98.9**	**100**	**92.8**	**94.6**
NSCL*	**90.5**	**93.8**	**92.7**	**99.3**	**100**	**93.7**	**95.0**

Table 3: **H-score of Office datasets in the 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
KAUDA	—	54.7
NSCL	**72.0**	**60.1**

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

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 (Bucci et al., 2020). 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 example, when the number of samples in the unknown category is much larger than the number of the known (e.g., the Office-Home dataset), pairing one more sample from the known category leads to a significant increase in A_c, which further leads to the H-score increase significantly. 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 (Kanas, 2017), we propose the Balance H-score as a more equitable metric (the proof is shown in the 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, NSCL achieves a new state-of-the-art (SOTA) on all four datasets in the most challenging UNDA setting. Concerning H-score, NSCL outperforms the previous SOTA UNDA method on Office by 4.2% and on OfficeHome by 0.3%. On large-scale datasets, NSCL also gives more than 0.2% improvement on DomainNet and more than 3.7% on VisDA compared to all other methods in terms of

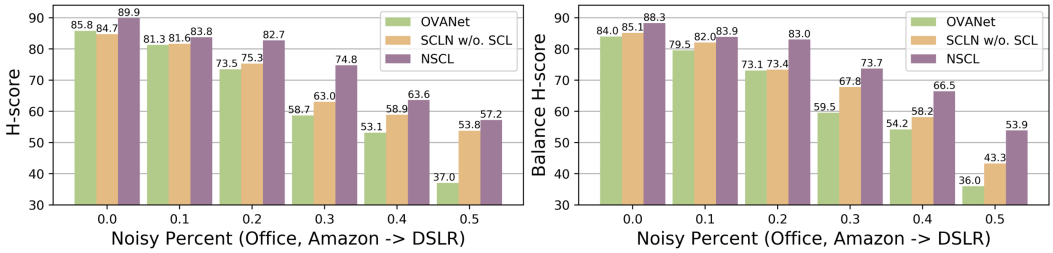


Figure 3: **Ablation study. OVANet v.s. OVANet+AIO v.s. NSCL.** H-score and Balance H-score Comparisons of Office datasets in the OPDA setting. The horizontal coordinate indicates the addition of a specified percentage of noise to the original domain, and the vertical coordinate indicates the performance of the different methods.

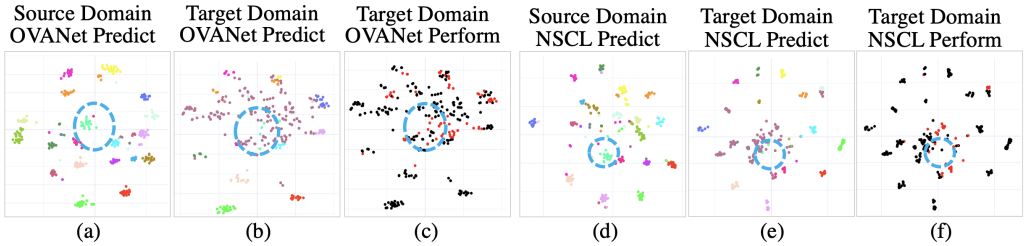


Figure 4: **Feature Visualization, OVANet V.S. NSCL.** (a), (b), (d), and (e) show the t-SNE visualization of embeddings produced by the backbone network. In (c) and (f), error samples (concentrate on know/unknown classification) are shown in blue. The blue circle marks a private cluster (in orange) where the OVANet sub-classifier is overconfident, and NSCL handles the same case better.

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 2.9% and 1.2% H-score improvement.

Overview of Results. Under these two scenarios with "unknown" samples, NSCL shows a more robust capability for separating common and private categories, which benefits from the global joint local feature alignment paradigm and adaptive energy uncertainty calibration strategy. Compared with GATE, a previous SOTA method tailed for the ODA setting, NSCL is also superior on all datasets. This evidence shows that NSCL 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 designed controlled experiments to verify the necessity of the soft contrastive learning (SCL) Loss. In Table. 5, the *NSCL w/o SCL* means the SCL loss $\mathcal{L}_{src}(\mathbf{x}_i^t, {y_i}^t)$ is removed in the overall loss. The *NSCL with CL* means the SCL loss $\mathcal{L}_{src}(\mathbf{x}_i^t, {y_i}^t)$ is replaced by the CL loss L_{INCE} in Eq. (1). The control experiments on all four datasets in UNDA settings are shown in Table 5. It indicates that the SCL Loss significantly outperforms typical CL loss (*NSCL with CL*) or without CL loss (*NSCL w/o. CL*). We attribute the failure of CL loss to the fact that the view noise caused by domain bias cannot be ignored to improve performance further. And our SCL loss can better deal with the above problem, thus improving performance.

Ablation study: the effect of AIO classifier. We further designed controlled experiments to verify the necessity of the All in one (AIO) classifier. In *NSCL w/o AIO* of Table. 5, the AIO classifier is replaced by open-set and closed-set classifier. The control experiments on all four datasets indicate that the AIO Classifier brings consistency 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

Method	Office Avg (10 / 10 / 11)	OfficeHome Avg (10 / 5 / 50)	DomainNet Avg (150 / 50 / 145)	VisDA (6 / 3 / 3)
NSCLw/o. SCL & AIO (OVANet)	77.9	71.8	50.7	66.1
NSCLwith CL	78.2	73.3	50.5	69.9
NSCLw/o. SCL	78.9	73.0	50.7	69.4
NSCLw/o. AIO	90.5	74.6	51.0	70.1
NSCL	**91.8**	**76.3**	**51.6**	**72.0**

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

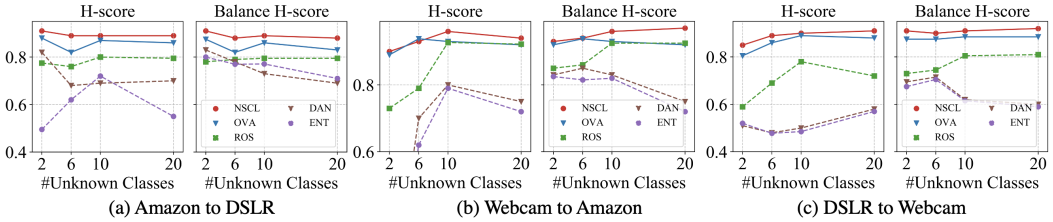


Figure 5: **H-score and accuracy comparison of Office dataset in ODA.** We vary the number of unknown classes using Office ($|L_s \cap L_t| = 10$, $|L_s - L_t| = 0$). The left and right parts, respectively, show H-score and accuracy. OVANet shows stable performance across different openness, while baselines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

are shown in Fig. 3. The results show that the NSCL and NSCL w/o RCL exceed the baseline more significantly as the proportion of noise increases.

The overconfidence problem and its mitigation by NSCL. 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 NSCL achieves a more significant advantage in datasets with fewer samples (e.g., Office). To explore the adverse effects of overconfidence, we performed a visual analysis of the A2D setting of the Office dataset in Fig. 4. We found that the open set classifier in OVANet corresponding to the private class (the orange scatters) had very sharp classification boundaries, as disjoint sample points are incorrectly classified to this class. We believe this is the result of overconfidence. Contrastingly, the same class is handled well by NSCL.

The effect of the proportion of unknown samples on H-score, and the advantage of NSCL on Balance H-score. For H-score, some fairness bias occurs if there is a large quantitative difference between 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 figure shows that changing the number of unknown classes causes drastic changes in the ordering of the H-score. Correspondingly, the Balance H-score exhibits higher stability. This indicates that the Balance H-score is a more stable metric for the proportion of unknown class samples, and its fairness is demonstrated in the appendix. In addition, the results in Fig.3 and Fig.5 show that the proposed NSCL has more evident advantages in both the H-score and Balance H-score.

5 Conclusion

Universal domain adaptation (UNDA) aims 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 noise problem of UNDA based on Contrastive Learning (CL), we propose Noise-Resistant Soft Contrastive Learning (NSCL). NSCL includes soft contrastive learning loss funtion to avoid the over-response of typical CL loss to noisy samples. The SCL loss enables data augmentation to improve the performance of UNDA. In addition, NSCL includes an all-in-one (AIO) classifier to improve the robustness of noisy labels. Extensive experimental results on UNDA and open-set DA demonstrate the advantages of NSCL over existing methods, especially in downstream tasks such as classification and visualization.

REFERENCES

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.

Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation. *Eur. Conf. Comput. Vis.*, pp. 422–438, 2020.

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

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

Liang Chen, Yihang Lou, Jianzhong He, Tao Bai, and Minghua Deng. Evidential neighborhood contrastive learning for universal domain adaptation. *AAAI*, 2022a.

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.*, 2022b.

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.

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

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.

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.

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

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

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

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

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

Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. *computer vision and pattern recognition*, 2021b.

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

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

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.

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.

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

Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. *arXiv: Computer Vision and Pattern Recognition*, 2021.

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

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *arXiv preprint arXiv:2002.07953*, 2020.

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

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

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

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

Yifan Wang, Lin Zhang, Ran Song, Lin Ma, and Wei Zhang. Exploiting inter-sample affinity for knowability-aware universal domain adaptation. 2022.

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

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

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

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.

A DETAILS OF SCL LOSS

A.1 DETAILS OF THE TRANSFORMATION FROM EQ. (1) TO EQ. (3)

We start with $L_{i,j} = -[\mathcal{H}_{ij} \log \kappa(d_{ij}^z) + (1-\mathcal{H}_{ij}) \log(1-\kappa(d_{ij}^z))]$ (Eq. (3)). 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:

$$\begin{aligned} L =& -1[L_{i,j} + \sum_{k=k_1}^{k_N} [L_{i,k}]] \\ =& -1[\log \kappa(d_{ij}^z) + \log \sum_{k=k_1}^{k_N} [1-\kappa(d_{ik}^z))]] \\ =& -1[N + \log \kappa(d_{ij}^z) - \log \sum_{k=k_1}^{k_N} [\kappa(d_{ik}^z)]] \\ =& -1[N + \log \frac{\kappa(d_{ij}^z)}{\sum_{k=k_1}^{k_N} [\kappa(d_{ik}^z)]}] \sim -\log \frac{\kappa(d_{ij}^z)}{\sum_{k=k_1}^{k_N} \kappa(d_{ik}^z)} \end{aligned}$$

After neglecting the in constant N, the loss L differs from InfoNCE loss

$$l_{\text{InfoNCE}} = -\log \frac{\exp\left(\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_k\right)/\tau\right)}$$

only in the method of measuring the similarity.

A.2 THE PROPOSED LOSS AS A SMOOTHER CL LOSS

the proposed loss is considered a smoother CL loss. We discuss the differences by comparing the two losses to prove this point. Let $g_\phi(\cdot)$ satisfies K-Lipschitz continuity, then $d_{ij}^z = k^* d_{ij}^y, k^* \in [1/K, K]$, where k^* is a Lipschitz constant. The difference between L_{SCL} loss and L_{INCE} loss is

$$|L_{\text{INCE}}(x_i) - L_{\text{SCL}}(x_i)| = \sum_j \{\mathcal{H}_{ij} - \kappa\left((1+(\alpha-1)\mathcal{H}_{ij}) k^* d_{ij}^z\right) \log(\frac{1}{\kappa(d_{ij}^z)} - 1)\}. \tag{11}$$

When $\alpha > 0$, the SCL loss is the soft version of the CL loss, which causes a minor collapse of local structures. where $\kappa(\cdot)$ is a t-distribution kernel as described in Eq. (2).

Let backbone network $F(\cdot)$ be Lipschitz continuity mapping. $d_{ij}^z = k^* d_{ij}^y, k^* \in [1/K, K]$, where K is a Lipschitz constant. For neighborhoods points i and j, The difference between SCL loss L_{SCL} and contrastive learning loss L_{CL} is

$$|L_{\text{SCL}} - L_{\text{CL}}| = -\sum \left\{ \mathcal{H}_{ij} - \kappa\left((1+(\alpha-1)\mathcal{H}_{ij}) k^* d_{ij}^z\right) \log(\frac{1}{\kappa(d_{ij}^z)} - 1) \right\}, \tag{12}$$

Detail. The contrastive learning loss is written as, For notational simplicity, we omit redundant symbols:

$$\begin{aligned} L_{CL} =& \sum \left\{ \mathcal{H}_{ij} \log \kappa\left(d_{ij}^z, \nu\right) + (1-\mathcal{H}_{ij}) \log\left(1 - \kappa\left(d_{ij}^z, \nu\right)\right) \right\} \\ =& -\mathbb{E}_{x_j, x_j} \left\{ \mathcal{H}_{ij} \log \kappa\left(d_{ij}^z\right) + (1-\mathcal{H}_{ij}) \log\left(1 - \kappa\left(d_{ij}^z\right)\right) \right\} \end{aligned} \tag{13}$$

where $\mathcal{H}_{ij} = \pi[(x_i, x_j) \in E]$ shows weath i and j are neighborhoods in graph $G(X, E)$. The SCL loss is written as:

$$L_{\text{SCL}} = -\sum \left\{ \kappa\left(R_{ij} d_{ij}^y\right) \log \kappa\left(d_{ij}^z\right) + \left(1 - \kappa\left(R_{ij} d_{ij}^y\right)\right) \log\left(1 - \kappa\left(d_{ij}^z\right)\right) \right\} \tag{14}$$

The difference between the two loss functions is:

$$
\begin{aligned}
& L_{CL}-L_{\mathrm{SCL}} \\
=&-\sum\left\{\mathcal{H}_{ij}\log\kappa\left(d_{ij}^{z}\right)+\left(1-\mathcal{H}_{ij}\right)\log\left(1-\kappa\left(d_{ij}^{z}\right)\right)\right\}- \\
&-\sum\left\{\kappa\left(R_{ij}d_{ij}^{y}\right)\log\kappa\left(d_{ij}^{z}\right)+\left(1-\kappa\left(R_{ij}d_{ij}^{y}\right)\right)\log\left(1-\kappa\left(d_{ij}^{z}\right)\right)\right\} \\
=&-\sum\left[\left(\mathcal{H}_{ij}-\kappa\left(R_{ij}d_{ij}^{y}\right)\right)\log\kappa\left(d_{ij}^{z}\right)+\left(1-\mathcal{H}_{ij}-1+\kappa\left(R_{ij}d_{ij}^{y}\right)\right)\log\left(1-\kappa\left(R_{ij}d_{ij}^{z}\right)\right)\right] \\
=&-\sum\left[\left(\mathcal{H}_{ij}-\kappa\left(R_{ij}d_{ij}^{y}\right)\right)\log\kappa\left(d_{ij}^{z}\right)+\left(\kappa\left(R_{ij}d_{ij}^{y}\right)-\mathcal{H}_{ij}\right)\log\left(1-\kappa\left(d_{ij}^{z}\right)\right)\right] \\
=&-\sum\left[\left(\mathcal{H}_{ij}-\kappa\left(R_{ij}d_{ij}^{y}\right)\right)\left(\log\kappa\left(d_{ij}^{z}\right)-\log\left(1-\kappa\left(d_{ij}^{z}\right)\right)\right)\right] \\
=&\sum\left[\left(\mathcal{H}_{ij}-\kappa\left(R_{ij}d_{ij}^{y}\right)\right)\log\left(\frac{1}{\kappa\left(d_{ij}^{z}\right)}-1\right)\right]
\end{aligned}
\tag{15}
$$

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

$$
L_{CL}-L_{\mathrm{SCL}}=\sum\left[\left(\mathcal{H}_{ij}-\kappa\left((1+(\alpha-1)\mathcal{H}_{ij})d_{ij}^{y}\right)\right)\log\left(\frac{1}{\kappa\left(d_{ij}^{z}\right)}-1\right)\right] \tag{16}
$$

We assume that network $g_{\phi}(\cdot)$ to be a Lipschitz continuity function, then

$$
\begin{aligned}
\frac{1}{K}g_{\phi}(d_{ij}^{y})\leq d_{ij}^{y}\leq Kg_{\phi}(d_{ij}^{y}) \quad \forall i,j\in\{1,2,\cdots,N\} \\
\frac{1}{K}d_{ij}^{z}\leq d_{ij}^{y}\leq Kd_{ij}^{z} \quad \forall i,j\in\{1,2,\cdots,N\}
\end{aligned}
\tag{17}
$$

We construct the inverse mapping of $g_{\phi}(\cdot):g_{\phi}^{-1}(\cdot)$

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

then exit k^{*}, let:

$$
d_{ij}^{y}=k^{*}d_{ij}^{z} \quad k^{*}\in[1/K,K] \quad \forall i,j\in\{1,2,\cdots,N\} \tag{19}
$$

Substituting the Eq.(19) into Eq.(16).

$$
L_{CL}-L_{\mathrm{SCL}}=\sum\left[\left(\mathcal{H}_{ij}-\kappa\left((1+(\alpha-1)\mathcal{H}_{ij})k^{*}d_{ij}^{z}\right)\right)\log\left(\frac{1}{\kappa\left(d_{ij}^{z}\right)}-1\right)\right] \tag{20}
$$

if $\mathcal{H}_{ij}=1$, we have:

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

then:

$$
\begin{aligned}
&\lim_{\alpha\to0}L_{CL}-L_{\mathrm{SCL}} \quad |_{\mathcal{H}_{ij}=1} \\
=&\sum\left[\left(1-\kappa\left(\alpha k^{*}d_{ij}^{z}\right)\right)\log\left(\frac{1}{\kappa\left(d_{ij}^{z}\right)}-1\right)\right] \\
=&0
\end{aligned}
\tag{22}
$$

Based on Eq.(22), we find that if i, j is neighbor and $\alpha \to 0$, there is no difference between the two loss functions L_{CL} and L_{SCL}. When $\alpha \nrightarrow 0$, the difference between the loss functions will be the function of d_{ij}^{z}. Because the contrastive learning loss L_{CL} only minimizes the distance between adjacent nodes and does not maintain any structural information. We believe that the loss of SCL will better preserve the structural information based on contrastive loss.

A.3 SCL IS BETTER THAN CL IN VIEW-NOISE

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

$$SNR = \frac{PL}{NL} \tag{23}$$

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 contrast learning's SNR is larger than contrast learning's, we should prove:

$$\frac{PL_{cl}}{NL_{cl}} < \frac{PL_{scl}}{NL_{scl}} \tag{24}$$

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

$$PL_{scl} = -M\left\{(1-S')\log(1-S') + S'\log S'\right\} - \left\{(1-e^{\alpha}S)\log(1-S) + e^{\alpha}S\log S\right\} \tag{25}$$

$$PL_{cl} = -M\log(1-S') - \log S \tag{26}$$

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

$$\begin{aligned}
&PL_{scl} - PL_{cl} \\
&= -M\left\{(1-S'-1)\log(1-S') + S'\log S'\right\} - \left\{(1-e^{\alpha}S)\log(1-S) + (e^{\alpha}S-1)\log S\right\} \\
&= -M\left\{S'(\log S' - \log(1-S'))\right\} - \left\{(e^{\alpha}S-1)(\log S - \log(1-S))\right\} \\
&= -M\left\{S' log\frac{S'}{(1-S')}\right\} - \left\{(e^{\alpha}S-1) log\frac{S}{(1-S)}\right\}
\end{aligned} \tag{27}$$

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:

$$(e^{\alpha}S-1) log\frac{S}{(1-S)} <= 0 \tag{28}$$

Also, we could get:

$$-M\left\{S' log\frac{S'}{(1-S')}\right\} > 0 \tag{29}$$

So we get:

$$PL_{scl} - PL_{cl} > 0 \tag{30}$$

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. (24) has been proved.

B DETAILS OF BALANCE HSCORE

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

$$\text{Balance Hscore} = B = \frac{1+\theta}{\frac{1}{A_c} + \frac{\theta}{A_t}} = \frac{A_t A_c}{A_t + \theta A_c}(1+\theta) \tag{31}$$

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,

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

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,

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

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

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

The impact of the Hscore by the unknown class is

$$\begin{aligned} \frac{\partial H}{\partial N_t} &= \frac{\partial H}{\partial A_t} \cdot \frac{\partial A_t}{\partial N_t} \\ &= 2A_c \cdot \frac{A_c + A_t - A_t}{(A_c + A_t)^2} \cdot \frac{1}{\theta M} \\ &= \frac{2A_c^2}{\theta M(A_c t A_t)^2} \end{aligned} \tag{35}$$

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 (Fu et al., 2020), DANCE (Saito et al., 2020), DCC (Li et al., 2021b), OVANet (Saito & Saenko, 2021), TNT (Chen et al., 2022a),GATE (Chen et al., 2022b), KUADA (Wang et al., 2022). Instead of reproducing the results of these papers, we directly used the results reported in the papers with the same configuration. We focused on contemporaneous work, although the results are not included in the table because these articles were not peer-reviewed.

C.2 DATASETS

We utilize popular datasets in DA: Office (Saenko et al., 2010), OfficeHome (Venkateswara et al., 2017), VisDA (Peng et al., 2017), and DomainNet (Peng et al., 2019). Unless otherwise noted, we follow existing protocols (Saito & Saenko, 2021) 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	$\lvert L_s \cap L_t \rvert$	$\lvert L_s - L_t \rvert$	$\lvert L_t - L_s \rvert$
ODA	Office-31	10	0	11
	Office-Home	25	0	40
	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 (Saito & Saenko, 2021) and GATE(Chen et al., 2022b), we employ ResNet50 (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009) as our backbone network. We train our models with inverse learning rate decay scheduling. We use a grid search to select the optimal model parameters; the selected parameters include λ and β. The performance of the proposed NSCL in uniform settings is listed in the penultimate row of the table. At the same time, a grid search is performed for each setup, and the optimal values obtained are marked with a *. 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 H-network uses a two-layer MLP network with 2048 neurons.

We use the grid search method to determine the best super parameters. The super parameters adjusted 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]