Machine Learning
https://doi.org/10.1007/s10994-023-06343-8

NaCL: noise-robust cross-domain contrastive learning for unsupervised domain adaptation

Jingzheng Li[1,3] · **Hailong Sun**[2,3]

Received: 27 November 2022 / Revised: 25 March 2023 / Accepted: 21 April 2023

Abstract
The Unsupervised Domain Adaptation (UDA) methods aim to enhance feature transferability possibly at the expense of feature discriminability. Recently, contrastive representation learning has been applied to UDA as a promising approach. One way is to combine the mainstream domain adaptation method with contrastive self-supervised tasks. The other way uses contrastive learning to align class-conditional distributions according to the semantic structure information of source and target domains. Nevertheless, there are some limitations in two aspects. One is that optimal solutions for the contrastive self-supervised learning and the domain discrepancy minimization may not be consistent. The other is that contrastive learning uses pseudo label information of target domain to align class-conditional distributions, where the pseudo label information contains noise such that false positive and negative pairs would deteriorate the performance of contrastive learning. To address these issues, we propose Noise-robust cross-domain Contrastive Learning (NaCL) to directly realize the domain adaptation task via simultaneously learning the instance-wise discrimination and encoding semantic structures in intra- and inter-domain to the learned representation space. More specifically, we adopt topology-based selection on the target domain to detect and remove false positive and negative pairs in contrastive loss. Theoretically, we demonstrate that not only NaCL can be considered an example of Expectation Maximization (EM), but also accurate pseudo label information is beneficial for reducing the expected error on target domain. NaCL obtains superior results on three public benchmarks. Further, NaCL can also be applied to semi-supervised domain adaptation with only minor modifications, achieving advanced diagnostic performance on COVID-19 dataset. Code is available at https://github.com/jingzhengli/NaCL

Keywords Domain adaptation · Contrastive learning · Clustering

Editors: Fabio Vitale, Tania Cerquitelli, Marcello Restell, Charalampos Tsourakakis.

Extended author information available on the last page of the article

1 Introduction

State-of-the-art deep learning models suffer from significant performance drops when the test distribution shifts from the training distribution (Yao et al., 2022), also known as domain shifts (Wiles et al., 2021). Although some research is dedicated to improving out-of-distribution robustness generally (Hendrycks et al., 2021), Unsupervised Domain Adaptation (UDA) methods (Ganin & Lempitsky, 2015) often achieve better performance where we assume access to the unlabeled target data from the test distribution.

Classical UDA methods have focused on reducing the discrepancy between the *marginal distributions* of source and target domains in representation space via statistics matching (Chen et al., 2020a; El Hamri et al., 2022) or domain adversarial learning (Tzeng et al., 2017; Saito et al., 2018; Xu et al., 2020). Recently advanced methods (Cicek & Soatto, 2019; Xu et al., 2022) consider explicitly aligning the *class-conditional distributions* between source domain and target domain by exploiting the label information on target domain. Although aligning the source and target domain distributions can improve the generalization ability of the learned model over the target domain, it usually comes at the expense of feature discriminability (Chen et al., 2019c; Tang et al., 2020). From this point, some works (Bekkouch et al., 2019; Kang et al., 2019; Thota & Leontidis, 2021) use contrastive learning to address UDA mainly in two ways. One way (Ghifary et al., 2016; Sun et al., 2019) is to perform the self-supervised pretext task (e.g., image rotation prediction or image in-painting) on source and target data together with the mainstream domain adaptation method to align domain distributions and learn instance-wise discrimination simultaneously. The other way (Chen et al., 2021; Wang et al., 2022) adopts cross-domain contrastive learning to pull the instances of same class closer while pushing away the instances of different classes between source domain and target domain by using the pseudo label information of target domain.

Despite using contrastive learning to solve UDA presents potential, there are some limitations that have not been well addressed mainly in two aspects. (1) The optimization direction for the instance-wise discrimination task in contrastive self-supervised learning may be inconsistent with the direction of minimizing domain discrepancy in domain adaptation methods (e.g., DANN (Ganin et al., 2016)). That is, feature discriminability and feature transferability may not be mutually reinforced. For instance, CaCo (Huang et al., 2022) shows experimentally that some instance-wise discrimination tasks do not perform well in UDA. It is still unclear how to define positive and negative pairs in contrastive learning that can enhance the feature transferability to achieve distribution alignment. (2) Recent cross-domain contrastive learning (Park et al., 2020; Wang et al., 2022) aligns the class-conditional distributions between source domain and target domain by using pseudo-labels on the target data. However, the pseudo-labels on the target data contain noisy labels such that false positive and negative pairs would deteriorate the performance of contrastive learning (Li et al., 2022). It is worth noting that some works (Wang & Breckon, 2020; Sharma et al., 2021) use thresholding to remove pseudo-labels with low confidence. Since domain alignment may corrupt the topological structures of target domain and pseudo-labels are involved in model training as a self-training manner, these factors lead to false pseudo-labels that are difficult to be corrected in subsequent training.

To address these issues, we propose a Noise-robust cross-domain Contrastive Learning (NaCL) to directly minimize domain discrepancy, which considers not only the in-domain instance-wise discrimination task but also cross-domain supervised contrastive learning to align class-conditional distributions. More importantly, NaCL considers

the problem of noisy labels in contrastive learning since the pseudo-labels in the target domain contain noisy labels. In detail, NaCL adopts topology-based selection to detect and remove possibly false positive and negative pairs based on the topological structures of target domain, to ensure the efficacy of contrastive learning. In a nutshell, our contributions can be summarized as follows.

- In this paper, we investigate how to design tailored contrastive loss for UDA tasks. Specifically, we propose NaCL to directly solve the UDA without combining the mainstream UDA method. More importantly, NaCL is robust to noisy positive and negative pairs in contrastive learning.
- Theoretically, we demonstrate that NaCL can be optimized as an example of Expectation Maximization. Meanwhile, we explain the importance of accurate target pseudo-labels in NaCL to reduce the expected risk on the target domain.
- NaCL can also be applied to Semi-Supervised Domain Adaptation (SSDA) with only minor modifications, achieving advanced diagnostic performance on COVID-19 dataset.
- We conduct extensive ablation studies and hyper-parameters analysis to verify the efficacy of NaCL. Notably, NaCL achieves state-of-the-art on three UDA benchmarks.

2 Related work

2.1 Unsupervised domain adaptation

The classic UDA methods aim at minimizing the discrepancy of marginal distributions between source and target domains to learn domain-invariant representations. Some works use various statistic distances such as Maximum Mean Discrepancy (MMD) (Long et al., 2017), Jensen-Shannon divergence (Ganin et al., 2016) and Wasserstein distance (Shen et al., 2018). Some other methods adopt an adversarial objective with respect to a domain discriminator to encourage domain confusion (Tzeng et al., 2017; Xu et al., 2020). Marginal distributions alignment may lead to misalignment between different classes even though the global distribution statistics are aligned. Therefore some works (Long et al., 2018; Pei et al., 2018) are devoted to aligning class-conditional distributions between source and target domains by utilizing the label distributions on source and target domains. For instance, these methods (Xie et al., 2018; Cicek & Soatto, 2019; Deng et al., 2019) pull together the class prototypes on the shared classes between source and target domains. However, boosting feature transferability may disrupt the intrinsic structure of the target domain leading to feature discriminability drop. Thus, some works (Xu et al., 2019; Chen et al., 2019a, c; Wang & Breckon, 2020; Jin et al., 2020) improve the discriminative ability of the target data under the assumption of structural domain similarity. For example, representation learning methods such as deep clustering (Ghasedi Dizaji et al., 2017) and contrastive representation learning are adopted in UDA to uncover the intrinsic target discrimination (Wang et al., 2019; Park et al., 2020). On this basis, we further propose noise-robust cross-domain contrastive learning that considers simultaneously feature transferability and feature discriminability, both of which reinforce each other.

2.2 Contrastive learning

Contrastive learning has evolved from early margin-based binary classification (Hadsell et al., 2006), to triplet loss (Schroff et al., 2015) and N-pair loss (Sohn, 2016). Recently the most attractive method for learning representations without manual annotations is contrastive self-supervised learning (Cao et al., 2020; Chen et al., 2020c; Tian et al., 2020; Caron et al., 2021). These methods aim at pulling together similar (or positive) instance pairs while pushing apart dissimilar (or negative) instance pairs to learn instance-wise discrimination via self-supervised pretext task. Besides, some methods such as BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021) use one view of the identical instance to predict another view to learn representations even without negative pairs. The instance-wise discrimination tasks treat two different instances as a negative pair and will result in many negative pairs that share similar semantics but are undesirably pushed apart in the representation space. Leveraging semantic structure information (or labels) in contrastive self-supervised learning has been shown to guide representations towards task-relevant features that improve performance (Henaff, 2020; Khosla et al., 2020; Wu et al., 2018a; Li et al., 2020). In this work, we investigate how to design contrastive loss by considering semantic information between the source and target domains for UDA.

2.3 Contrastive learning for UDA

Recently, adopting contrastive learning to solve UDA task achieves advanced performance. On the one hand, the UDA methods jointly with contrastive self-supervised learning improves the discriminability of the target features so that the model generalizes well over the target domain (Bousmalis et al., 2016; Sun et al., 2019). On the other hand, cross-domain contrastive learning (Kim et al., 2020; Park et al., 2020) utilizes pseudo-labels of target data to align class-conditional distributions between source and target domains via contrastive loss. For instance, Kang et al. (2019) and Huang et al. (2022) propose the *category contrast* to perform semantic-aware alignment. Wang et al. (2022) and Chen et al. (2021) use bidirectional supervised contrastive learning between the source and target domains. Further, Sharma et al. (2021) propose an affinity matrix-based multi-sample contrastive loss together with domain adversarial network. However, the aforementioned methods that use contrastive learning to align class-conditional distributions require the label information of target domain, where pseudo labels containing noise would deteriorate the performance of contrastive learning. Our NaCL is robust to false positive and negative pairs in contrastive learning.

3 Method

3.1 Problem setting

We first present basic notations and the goal of UDA. Formally, given a fully-labeled source domain dataset with n_s instance and label pairs $D_s = (\mathcal{X}_s, \mathcal{Y}_s) = \{\boldsymbol{x}_i^s, y_i^s\}_{i=1}^{n_s}$, and an unlabeled dataset in a target domain with n_t instances $D_t = \mathcal{X}_t = \{\boldsymbol{x}_i^t\}_{i=1}^{n_t}$. Both D_s and D_t denote the input data drawn i.i.d. from two different distributions but belong to the

same label space of C categories. UDA aims to learn a classifier h that performs well over $\mathcal{X}_t$, with the help of *source classification loss* on labeled source data via

$$\mathcal{L}_{\mathrm{S}} = -\frac{1}{n_s}\sum_{i=1}^{n_s} l\left(h\left(\boldsymbol{x}_i^s\right), y_i^s\right) \tag{1}$$

in which l is an accuracy-related loss, e.g., the standard cross-entropy loss.

3.2 Preliminaries of contrastive learning with instance discrimination

The contrastive self-supervised learning learns instance-wise discrimination which can be considered to train an encoder for solving *dictionary look-up* task. The queries and keys represent different *views* of the instances and are extracted from the encoder. The objective aims to learn a representation space where positive pairs, e.g., different views of the same instance, are pulled closer and negative pairs, e.g., views from different instances, are pushed apart. Formally, within a mini-batch I, consider a query $\boldsymbol{q}_i$ and a dictionary $K = \left\{\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_N\right\}$ that consists of a number of keys. $\boldsymbol{k}_j^+$ in the dictionary is positive key for $\boldsymbol{q}_i$, and we treat the other keys as negative instances. A form of a contrastive loss, called InfoNCE, is then formulated as

$$\mathcal{L}_{\mathrm{CL}} = -\sum_{\boldsymbol{q}_i \in I} \log \frac{\exp\left(\boldsymbol{q}_i \cdot \boldsymbol{k}_i^+ / \tau\right)}{\sum_{\boldsymbol{k}_j \in K} \exp\left(\boldsymbol{q}_i \cdot \boldsymbol{k}_j / \tau\right)} \tag{2}$$

in which τ is a temperature hyper-parameter.

3.3 Noise-robust cross-domain contrastive learning

Figure 1 illustrates the framework and visual explanation of our method. Note that the dictionary in Fig. 1 is a *queue* that is generated by a momentum-updated key encoder on-the-fly (He et al., 2020). Our method can also adopt other dictionary generation methods such as a memory bank that stores the keys of all instances in the previous epoch (Wu et al., 2018b), or an end-to-end dictionary that generates keys from instances of the current mini-batch (Chen et al., 2020b). Below we detail the proposed NaCL for UDA task to address the aforementioned limitations.

First, we need to obtain the pseudo-labels of target domain by a certain method which will be discussed later in Sect. 3.4; the labels of the source domain are available. Second, we define tailored contrastive loss so that it encourages distribution alignment between source domain and target domain. In detail, the positive pair is considered as the views of different instances with the same class regardless of whether they come from the source or target domain, which also covers the different views of same instance; the negative pair is considered as the views of instances from different classes across source domain and target domain. Finally, we further propose *topology-based selection* to remove false positive and negative pairs in contrastive loss by setting a binary indicator g_i for each instance which indicates the instance $\boldsymbol{x}_i$ will be selected for contrastive learning if $g_i = 1$ and vice versa. Note that $g_i = 1$ for all source data.

Before introducing topology-based selection, assume that we have g_i for each instance, we define noise-robust cross-domain contrastive loss within source mini-batch I^s and target mini-batch I^t as

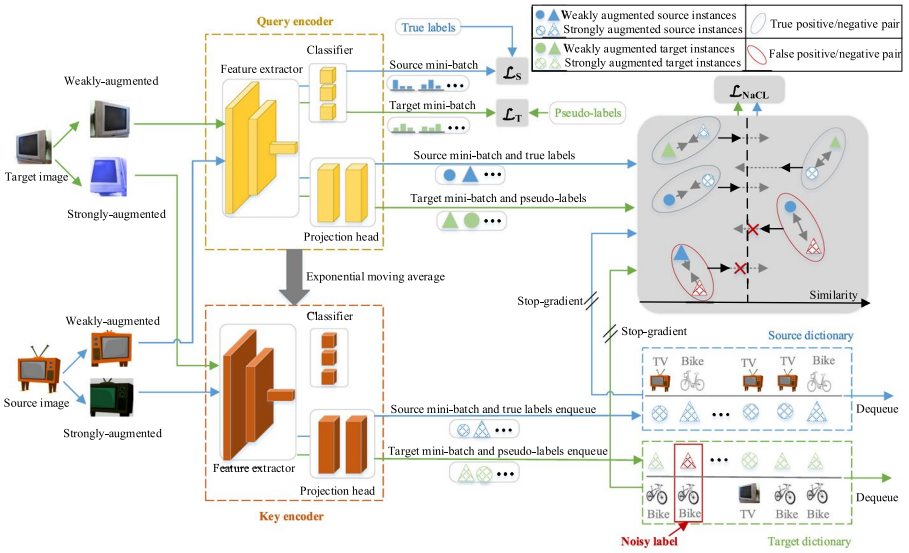

Fig. 1 An illustration shows the overall training process and framework of NaCL, in which as an example the dictionary is a queue generated by a momentum-updated key encoder. The main components of the framework of NaCL include data augmentation, query encoder and key encoder, which are structurally adapted from MOCO (He et al., 2020). Specifically, the query encoder takes the weakly augmented mini-batches as input to learn representations through the feature extractor. The feature extractor is followed by a classifier to optimize source and target classification losses, and a projection head to output queries used in the contrastive loss. The key encoder takes strongly augmented mini-batches as input to dynamically maintain queues (i.e., dictionary) of keys for source and target data, respectively. The key encoder has the same architecture as the query encoder, but the parameters of the key encoder are updated by exponential moving average of the query encoder rather than back-propagation. NaCL considers not only in-domain instance-wise discrimination task but also cross-domain class-conditional distribution alignment task. In particular, NaCL uses topology-based selection to remove false positive and negative pairs in the contrastive loss

$$\mathcal{L}_{\text{NaCL}} = -\sum_{\boldsymbol{q}_i \in \hat{I}} \frac{\mathbb{I}[g_i = 1]}{\mid P(i) \mid} \sum_{\boldsymbol{k}_i^+ \in P(i)} \log \frac{\exp\left(\boldsymbol{q}_i \cdot \boldsymbol{k}_i^+ / \tau\right)}{\sum_{\boldsymbol{k}_j \in A} \exp\left(\boldsymbol{q}_i \cdot \boldsymbol{k}_j / \tau\right)} \tag{3}$$

where $\hat{I} \equiv \{\boldsymbol{q}_i \mid \boldsymbol{q}_i \in I^s \cup I^t\}$ denotes source and target domain-mixed mini-batches, $\mathbb{I}[\cdot]$ is an indicator function to ensure that incorrectly labeled instances are excluded from training, $A \equiv \left\{\boldsymbol{k}_j \mid \boldsymbol{k}_j \in K^s \cup K^t\right\}$ denotes the domain-mixed dictionary, $P(i) \equiv \left\{\boldsymbol{k}_a \mid \boldsymbol{k}_a \in K^s \cup K^t, y_a = y_i\right\}$ denotes the positive instances in domain-mixed dictionary that share the same label with query $\boldsymbol{q}_i$. In doing so, feature discriminability and feature transferability will be mutually enhanced. As the training proceeds, the pseudo-labels become more accurate at each epoch, then more target data will be selected in contrastive loss.

3.3.1 Topology-based selection

We argue that the instances with noisy labels are usually far away from clean instances with the same labels in the representation space (Bahri et al., 2020; Wu et al., 2020). If we connect instances over target data to construct graph according to their k-nearest neighbors,

then we can remove instances that may be noisy labels based on the topological structure of this graph (Gao et al., 2016; Wu et al., 2021). Figure 2 depicts the proposed topology-based selection which can be subdivided into two steps: Largest Connected Component (LCC)-based selection and k-nearest neighbor (KNN)-based selection. We first detail the LCC-based selection. The representation vectors are obtained by feeding the feature extractor with the target data. We construct a k-nearest neighbor graph G over the target data based on the similarity between the representation vectors. For each class c, we first derive the subgraph $G(c)$ on G by removing the vertices belonging to other classes and their associated edges; then we find the LCC $G(c)_{lcc}$ on $G(c)$ and consider the instances belonging to $G(c)_{lcc}$ as clean; finally we obtain a collection of potentially clean instances $\mathcal{S} = \bigcup_{c=1}^{C} G(c)_{lcc}$. However, the instances distributed in the outer layer of the LCC are likely to be corrupted. We adopt KNN-based selection to further improve the purity of the subgraph $\mathcal{S}$. Concretely, for each instance $\boldsymbol{x}_i$ belonging to subgraph $\mathcal{S}$, we find its k-nearest neighbors $KNN(\boldsymbol{x}_i)$ in subgraph $\mathcal{S}$, and then this instance is treated as clean if labels of $KNN(\boldsymbol{x}_i)$ are all the same as the label of $\boldsymbol{x}_i$. Thus, for each instance $\boldsymbol{x}_i$ in target data, we have

$$g_i = \underbrace{\mathbb{I}\left[\boldsymbol{x}_i \in \mathcal{S}\right]}_{\text{LCC-based}} \wedge \underbrace{\mathbb{I}\left[KNN\left(\boldsymbol{x}_i\right) = E\left(\boldsymbol{x}_i\right)\right]}_{\text{KNN-based}} \tag{4}$$

where $E(\boldsymbol{x}_i) \equiv \left\{\boldsymbol{x}_j \mid \boldsymbol{x}_j \in KNN(\boldsymbol{x}_i), y_j = y_i\right\}$ denotes instances in $KNN(\boldsymbol{x}_i)$ have same label as instance $\boldsymbol{x}_i$.

3.3.2 Overall objective

Inspired by advanced semi-supervised methods such as FixMatch (Sohn & Berthelot, 2020), we use the cross-entropy loss between the pseudo-label y_i^t of target instance $\boldsymbol{x}_i$ with topology-based selection and its output $h\left(\boldsymbol{x}_i^t\right)$ on the classifier to optimize *target classification loss*, iff. the output confidence exceeds a threshold ρ (ρ=0.95), formalized as

$$\mathcal{L}_{\mathrm{T}} = -\frac{1}{n_t} \sum_{i=1}^{n_t} \mathbb{I}[h\left(\boldsymbol{x}_i^t\right)_{y_i^t} > \rho] \wedge \mathbb{I}[g_i = 1] l\left(h\left(\boldsymbol{x}_i^t\right), y_i^t\right). \tag{5}$$

where $h\left(\boldsymbol{x}_i^t\right)_{y_i^t}$ denotes the y_i^t-th component of the vector-valued function $h\left(\boldsymbol{x}_i^t\right)$ to obtain the confidence of pseudo-label y_i^t for target instance $\boldsymbol{x}_i$.

In summary, within mini-batches from source and target domains, combining source and target classification losses $\mathcal{L}_{\mathrm{S}}$, $\mathcal{L}_{\mathrm{T}}$ and noise-robust cross-domain contrastive loss $\mathcal{L}_{\mathrm{NaCL}}$, the overall objective of NaCL is

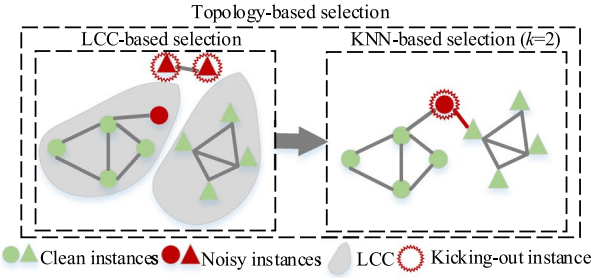

Fig. 2 An illustration of topology-based selection

$$\min \mathcal{L}_{\mathrm{S}} + \mathcal{L}_{\mathrm{T}} + \gamma \mathcal{L}_{\mathrm{NaCL}} \tag{6}$$

where γ controls the strength of contrastive loss.

3.4 Iterative pseudo-label assignment for the target domain

The UDA methods to obtain pseudo-labels on the target domain are mainly based on model predictions (Xie et al., 2018; Chen et al., 2019b), the nearest class prototype of source domain (Chen et al., 2019a; Lin et al., 2022), and clustering on target domain (Kang et al., 2019; Xu et al., 2022). We here use clustering to obtain pseudo-labels and later discuss the choice of pseudo-labels in Sect. 4.6.

Below we detail the process of obtaining pseudo-labels by using clustering. At each epoch in training process, we update pseudo-labels via clustering. Firstly (representation extraction), we collect current representation vectors of source and target instances without augmentation via the feature extractor. The representation vectors are normalized to the unit hypersphere, which enables using an inner product to measure distances in clustering. Secondly (center initialization), the number of clusters in clustering is set to the number of classes. For each class c, the cluster center is initialized as the mean representation of class c from source domain. Lastly (pseudo-label acquisition), we perform spherical k-means clustering on target representation vectors to obtain pseudo-labels, according to the class index of cluster center to which each instance's cluster assignment belongs. Notably, the collected representation vectors of the target domain are also used for topology-based selection to detect noisy labels.

3.5 Extension to semi-supervised domain adaptation (SSDA)

SSDA (Berthelot et al., 2021) differs from UDA in that partial target domain data has labels. NaCL can be used to address SSDA with minor modifications. (1) We concatenate mini-batches from the labeled source data and labeled target data to minimize the classification loss as Eq.(1). (2) In the process of obtaining pseudo-labels by using clustering, the initialized cluster centers can be set to the centroids of the partially labeled target data.

3.6 Theoretical insights

Proposition 1 *At each epoch in the training process, we alternately update pseudo labels of target data via clustering and optimize contrastive loss to learn representations. The proposed noise-robust cross-domain contrastive learning can be considered as a maximum likelihood problem optimized by Expectation Maximization (EM).*

The proof is provided "in the appendix". We further quantitatively report the behavior of noise-robust cross-domain contrastive loss in Fig. 5.

Previous work (Ben-David et al., 2010; Mansour et al., 2009) gives theoretical guarantee of the expected risk over the target domain with a similar form summarized as Theorem 1.

Theorem 1 *Given source domain distribution P_S and target domain distribution P_T, let $h \in \mathcal{H}$ be a hypothesis, the following inequality on the expected error of target domain holds,*

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(P_S, P_T\right) + \lambda \tag{7}$$

in which $\epsilon_S(h)$ depicts the expected error on source domain, $\mathcal{H}\Delta\mathcal{H}\left(P_S, P_T\right)$ is $\mathcal{H}\Delta\mathcal{H}$-divergence which is measured as the discrepancy between two hypotheses, and $\lambda = \epsilon_T(h^*) + \epsilon_S(h^*)$ is the combined error of ideal joint hypothesis with $h^* = \arg\min_{h\in\mathcal{H}} \epsilon_T(h) + \epsilon_S(h)$. λ depends on the capacity of the hypothesis space $\mathcal{H}$.

The following expected error analysis of NaCL over target domain is based on the previous theoretical bound of domain adaptation.

Lemma 1 *Inspired by the previous theoretical bound, given source domain distribution P_S and target domain distribution P_T, let $h \in \mathcal{H}$ be a hypothesis, f_S and f_T be the true labeling function for source and target data respectively. Given pseudo-labels $\hat{\mathcal{Y}} = \left\{\hat{y}_i^t\right\}_{i=1}^{n_t}$, $\hat{f}_T$ is the pseudo-labeling function for target domain. The expected error $\epsilon_T(h)$ of target domain w.r.t pseudo-labels in the deduction of the upper bound holds*

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(P_S, P_T\right) + \epsilon_T\left(\hat{f}_T, f_T\right) + \beta \tag{8}$$

where $\epsilon_T\left(\hat{f}_T, f_T\right) = \mathbb{E}_{(\boldsymbol{x}^t, \hat{y}^t)\sim P_t}\left[\left|\, \hat{f}_T(\boldsymbol{x}^t) - f_T(\boldsymbol{x}^t) \,\right|\right]$ depicts a disagreement between target pseudo-labels and true labels and $\beta = \epsilon_T(h^*, \hat{f}_T) + \epsilon_S(h^*)$ is a combined error of ideal joint hypothesis with $h^* = \arg\min_{h\in\mathcal{H}} \epsilon_T(h, \hat{f}_T) + \epsilon_S(h)$.

Remark 1 The proof is provided "in the appendix". For NaCL, the first term on the right-hand side of Eq.(8), i.e., $\epsilon_S(h)$, is minimized by source classification loss; the second term, i.e., $\frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(P_S, P_T\right)$, indicates the domain discrepancy which can be minimized by the cross-domain contrastive loss; the topology-based selection to select accurate target pseudo-labels in contrastive loss implicitly minimizes the disagreement between target pseudo-labels and true labels, i.e., $\epsilon_T\left(\hat{f}_T, f_T\right)$; and β can generally be viewed as a constant w.r.t h.

4 Experiments

4.1 Datasets

Office31 (Saenko et al., 2010) is a popular benchmark for evaluation on domain adaptation. It contains 4110 images of 31 classes in total, which are collected from three domains, including Amazon (**A**) comprising 2817 images downloaded from online merchants, Webcam (**W**) involving 795 low resolution images acquired from webcams, and DSLR (**D**) containing 498 high resolution images of digital SLRs. We evaluate all methods on all the 6 transfer tasks.

OfficeHome (Venkateswara et al., 2017) is a medium-scale benchmark dataset, with 15.5K images of 65 classes shared by four extremely distinct domains: Artistic images (**A**), Clip Art (**C**), Product images (**P**), and Real-World images (**R**). We try all 12 combinations of 4 domains for evaluation.

VisDA-2017 (Peng et al., 2017) is a more challenging simulation-to-real dataset with two extremely distinct domains: Synthetic renderings of 3D models and Real are collected from photo-realistic or real image datasets. With 280K images in 12 classes, the scale of VisDA-2017 brings challenges to domain adaptation.

ImageNet-scale dataset is a ImageNet to ImageNet-Rendition (Hendrycks et al., 2021) transfer task, where ImageNet-Rendition has renditions of 200 ImageNet classes resulting in 30K images.

COVID-19 (Zhang et al., 2020) is a medical X-ray image analysis dataset containing 11663 images from two domains with 3 classes for the diagnosis of COVID-19. The source domain consists of normal and pneumonia cases, while the target domain consists of normal and COVID-19 cases. The statistics of COVID-19 dataset are summarized in Table 1, in which about 30% of target domain data are labeled in training set. Such semi-supervised domain adaptation setting would be more practical in real-world scenario.

4.2 Data processing

We first pre-process each image by resizing its shorter size to 256 while keeping the aspect ratio unchanged. Then we randomly crop a region of 224×224 for training and the central 224×224 region for testing. We normalize all images with the mean and standard deviation calculated from the ImageNet dataset.

For data augmentation, our method uses one "weak" augmentation and one "strong" augmentation. The weak augmentation is the standard random crop and flip, which is the same as UDA baselines in the public library (Jiang et al., 2020). For strong augmentation, we follow the augmentation strategy in SimCLR (Chen et al., 2020b) which applies random color jittering and grayscale conversion, etc.

4.3 Networks and hyper-parameters

NaCL adopts two dictionary implementations as described in Sect. 3.3, i.e., dictionary obtained by momentum-updated key encoder and end-to-end dictionary, denoted **NaCL(momentum)** and **NaCL(end-to-end)** respectively. If not otherwise specified, NaCL refers to NaCL(momentum) in experiments.

We use ResNet-50 (He et al., 2016a) for Office31, OfficeHome and ImageNet-scale datasets, ResNet-101 (He et al., 2016b) for VisDA-2017 dataset, Resnet-18 (He et al., 2016a) for COVID-19 dataset. We adopt ImageNet pre-trained network as the feature extractor, and add task-specific FC layer as classifier and a two-layer multilayer perceptron with 128-dimensional output as projection head where the contrastive loss is applied. We normalize the output of projection head to ensure that inner product can be used in contrastive loss.

Table 1 Statistic of the COVID-19 dataset under SSDA

Set	Domain	Category		
		#Normal	#Pneumonia	#COVID-19
Training	Pneumonia	5613	2306	0
	COVID-19	2541	0	258
Test	COVID-19	885	0	60

Joining previous practices (Ganin et al., 2016; Long et al., 2017), the learning rate of the newly added layers are set 10 times to that of pre-trained networks. We utilize mini-batch SGD with momentum of 0.9 and employ an annealing strategy of learning rate: the learning rate (lr) is adjusted during the SGD by $lr_p = \frac{lr_0}{(1+\alpha p)^\beta}$, where p is increased linearly from 0 to 1 as training proceeds, we set $lr_0 = 0.001$, $\alpha = 10$, and $\beta = 0.75$. For all transfer tasks in experiments, the k is set to 3 to construct the k-nearest neighbor graph; the contrastive loss strength γ is set to 1; the temperature τ is set to 0.07 (Wu et al., 2018a); the batch size is set to 32. We take 2,000, 10,000, 10,000 and 10,000 iterations for Office31, OfficeHome, VisDA-2017 and COVID-19 respectively. We take 100 iterations to complete 1 epoch during training. For NaCL(momentum), the size of source dictionary is the same as the size of target dictionary. The size of dictionary is set to an integer multiple of the number of classes, which are 20, 20, 200, and 200 on the four datasets, respectively.

4.4 Unsupervised domain adaptation (UDA)

We conduct experiments on Office31, OfficeHome and VisDA-2017 for UDA task. All labeled source data and all unlabeled target data are used for training, and the test results on target data are reported in the *transductive* UDA following the standard protocol (Ganin & Lempitsky, 2015; Long et al., 2018). We report average classification results over three random trials.

4.4.1 Baselines

We adopt the public library (Jiang et al., 2020) to implement the baselines including **Source Only**, **DANN** (Ganin et al., 2016), **Self-ensembling** (French et al., 2018), **CDAN** (Long et al., 2018), **MCD** (Saito et al., 2018), and **MDD** (Zhang et al., 2019b), in which the models are well-tuned and achieve superior results over official implementations. An advanced UDA method Dynamic Weighted Learning (**DWL**) (Xiao & Zhang, 2021) are also used for comparison. Moreover, we also compare some UDA methods that adopt contrastive learning including Joint Contrastive Learning (**JCL**) (Park et al., 2020), Contrastive Adaptation Network (**CAN**) (Kang et al., 2019) and Category Contrast (**CaCo**) (Huang et al., 2022). Except for some baselines that we reproduce, the remaining experimental results are directly reported from their respective papers.

4.4.2 Results

All the results on Office31 and OfficeHome based on network backbone adapted from a ResNet-50 are reported in Tables 2 and 3. Table 4 lists the test accuracy over 12 classes on VisDA-2017 based on a network backbone adapted from a ResNet-101. We draw some observations. Overall, NaCL based on two types of dictionary achieves advanced performance on three benchmarks with different scales, respectively. For Office31, both NaCL(momentum) and NaCL(end-to-end) obtain average accuracy above 90%, exceeding the 87.6% achieved by CaCo that also uses contrastive learning to align class-conditional distribution. These gains are mainly contributed to the fact that we take into account the false positive and negative pairs in the contrastive loss. Further, NaCL also achieves advanced performance on OfficeHome with more classes, yielding average accuracy of 68.7% and 69.5% for NaCL(momentum) and NaCL(end-to-end), respectively. For the more challenging VisDA-2017 dataset, NaCL(momentum) and NaCL(end-to-end) obtain

Table 2 Test accuracy (%) on the small-scale Office31

Methods	A → W	D → W	W → D	A → D	D → A	W → A	Avg.
Source only	74.3	95.8	99.0	72.9	64.6	64.1	78.5
DANN	82.0	96.9	99.1	79.7	68.2	67.4	82.2
Self-ensembling	86.1	98.3	100.0	89.2	68.8	67.5	85.0
MCD	88.1	98.3	100.0	87.5	71.6	68.1	85.6
MDD	94.5	98.4	100.0	93.5	74.6	72.2	88.9
CDAN	89.3	98.6	100.0	94.1	75.5	71.9	88.2
CAN⋆	94.5	99.1	99.8	95.0	78.0	77.0	90.6
JCL	87.4	96.0	97.0	90.4	77.3	77.7	87.6
DWL⋆	89.2	99.2	100.0	91.2	73.1	69.8	87.1
CaCo⋆	89.7	98.4	100.0	91.7	73.1	72.8	87.6
NaCL(momentum)	93.1	99.2	100.0	93.6	77.4	76.5	90.0
NaCL(end-to-end)	94.0	99.1	100.0	93.4	77.9	76.7	90.2

The star⋆ indicates the results are copied from respective papers

average accuracy of 88.3% and 87.2%, respectively, while most other baselines do not exceed 80%, except that CAN reaches 87.2%.

It is worth noting that NaCL performs better on the large-scale VisDA-2017 datasets than the baselines on the other two relatively small-scale datasets. To explore whether NaCL is more applicable to large-scale data or if this result is specific to VisDA-2017 dataset, we additionally evaluated the performance of NaCL on ImageNet to ImageNet-Rendition transfer task. The experimental results are shown in Table 5, where the experimental results of the baselines are copied from open-sourced libariry (Jiang et al., 2020) and the baselines are well-tuned. The experimental results show that the performance of NaCL is significantly superior to that of the baselines on the ImageNet-scale dataset. We conjecture that a large amount of data will contribute to intro-domain contrastive self-supervised learning and cross-domain contrastive learning in NaCL to learn transferable representations for domain alignment.

4.5 Semi-supervised domain adaptation (SSDA)

We evaluate the diagnostic performance of NaCL on COVID-19 for SSDA task. All labeled source data and target data containing partially labeled target data are used for training, and the results are reported on held out target data that are not seen during training.

4.5.1 Baselines

NaCL is compared with (1) **Source Only**: training the model on labeled source data only; (2) **Target Only**: training the model on partially labeled target data only; (3) **Fine-tuning**: training the model on labeled source data and then fine-tuning it on the partially labeled target data. Moreover, we also compare some SSDA methods including **SDT** (Tzeng et al., 2015), **Semi-DMAN** extended from DMAN (Zhang et al., 2019a), and **COVID-DA** (Zhang et al., 2020) tailored for medical diagnosis.

Table 3 Test accuracy (%) on the medium-scale OfficeHome dataset

Methods	A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	Avg.
Source Only	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
Self-Ensembling	52.1	72.0	75.8	62.2	70.3	69.7	60.9	52.3	77.5	71.9	57.6	80.7	66.9
MCD	51.7	72.2	78.2	63.7	69.5	70.8	61.5	52.8	78.0	74.5	58.4	81.8	67.8
MDD	56.2	75.4	79.6	63.5	72.1	73.8	62.5	54.8	79.9	73.5	60.9	84.5	69.7
CDAN	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
NaCL(momentum)	50.3	78.2	79.8	64.6	73.8	76.0	65.9	47.0	82.0	70.9	52.3	83.4	68.7
NaCL(end-to-end)	52.7	79.9	80.9	65.4	74.0	76.9	66.3	49.0	81.8	71.0	53.0	83.3	69.5

Table 4 Test accuracy (%) on the large-scale VisDA-2017

Methods	Airplane	Bicycle	Bus	Car	Horse	Knife	Motor	Person	Plant	Skate	Train	Truck	Avg.
Source only	63.6	35.3	50.6	78.2	74.6	18.7	82.1	16.0	84.2	35.5	77.4	4.7	56.9
DANN	93.5	74.3	83.4	50.7	87.2	90.2	89.9	76.1	88.1	91.4	89.7	39.8	74.9
Self-Ensembling	95.6	66.8	84.0	88.5	96.6	90.8	93.7	37.0	95.4	81.5	91.8	26.3	79.1
MCD	87.8	75.7	84.2	78.1	91.6	95.3	88.1	78.3	83.4	64.5	84.8	20.9	76.7
MDD	89.8	70.9	83.7	61.2	89.3	93.8	90.6	75.2	88.2	88.0	83.4	48.5	77.0
CDAN	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
JCL	97.3	87.2	79.6	65.8	95.9	92.0	88.8	72.9	90.5	91.8	82.3	69.7	84.5
CAN*	97.0	87.2	82.5	74.3	97.8	96.2	90.8	80.7	96.6	96.3	87.5	59.9	87.2
DWL*	90.7	80.2	86.1	67.6	92.4	81.5	86.8	78.0	90.6	57.1	85.6	28.7	77.1
CaCo*	90.4	80.7	78.8	57.0	88.9	87.0	81.3	79.4	88.7	88.1	86.8	63.9	80.9
NaCL(momentum)	98.0	90.2	84.8	88.1	96.8	93.4	93.6	81.5	96.6	93.8	87.7	55.4	88.3
NaCL(end-to-end)	98.4	88.4	88.4	81.7	96.6	94.7	94.4	82.5	96.6	93.1	84.4	46.7	87.2

Table 5 Test accuracy (%) on ImageNet to ImageNet-Rendition transfer task

Methods	ImageNet→ ImageNet-Rendition
Source only	35.6
DANN	52.7
MCD	46.7
MDD	56.2
CDAN	53.9
NaCL	62.0

4.5.2 Results

We evaluate NaCL in terms of 4 metrics and report the experimental results in Table 6. NaCL obtains competitive results compared to the advanced COVID-DA, performing best on Recall and AUC metrics. The Recall of 95% demonstrates the superiority of NaCL for computer-aided diagnosis of COVID-19.

4.6 Ablation study

We conduct ablation studies to examine the effect of different components in NaCL, by evaluating some variants of NaCL. (1) **NaCL (w/o topology.)** denotes that all pseudo-labeled target data are used in contrastive loss, i.e., Eq.(3), without using topology-based selection. That is, there are false positive and negative pairs in contrastive loss. To analyze the topology-based selection more specifically, we perform ablation studies for LCC-based selection and kNN-based selection, respectively. (2) **NaCL (w/o LCC.)** indicates that LCC-based selection is not used in topology-based selection for NaCL. (3) **NaCL (w/o kNN.)** indicates that kNN-based selection is not used in topology-based selection for NaCL. (4) **NaCL (model pred.)** means that the pseudo-labels on target domain in NaCL are derived by model predictions instead of clustering. (5) **NaCL (w/o $\mathcal{L}_{\mathrm{T}}$)** indicates that $\mathcal{L}_{\mathrm{T}}$, i.e., Eq.(5), is not used in the overall objective to assess the effect of target classification loss.

The average results for all transfer tasks on three benchmarks are reported in Table 7. The ablation studies illustrate that when any of components is replaced, the performance

Table 6 Comparisons on COVID-19 dataset diagnosis with respect to F1 Score (%), Recall (%), Precision (%) and AUC

Methods	F1 Score	Recall	Precision	AUC
Source Only	62.16	69.42	71.69	0.910
Target Only	70.46	63.42	92.70	0.959
Fine-tuning	75.39	87.59	72.49	0.948
SDT*	79.69	85.00	75.00	0.962
Semi-DMAN*	77.27	85.00	70.83	0.978
COVID-DA*	**92.98**	88.33	**98.15**	0.985
NaCL	90.68	**95.00**	86.73	**0.989**

The star* indicates that the results are copied from Zhang et al. (2020). The best results are shown in bold

Table 7 Ablation studies to test different components in NaCL

Methods	Office31	OfficeHome	VisDA-2017
NaCL (w/o topology.)	87.4	66.3	81.1
NaCL (w/o LCC.)	88.3	66.4	84.6
NaCL (w/o kNN.)	88.9	67.8	86.7
NaCL (model pred.)	87.6	66.9	82.4
NaCL (w/o $\mathcal{L}_{\mathrm{T}}$)	88.9	67.9	87.1
NaCL	90.0	68.7	88.3

The average test accuracy (%) of all transfer tasks on three benchmarks are reported, respectively

of NaCL degrades. In detail, NaCL (w/o topology.) does not remove false positive and negative pairs in contrastive loss, which deteriorates the performance of contrastive learning. NaCL (model pred.) uses model predictions to provide pseudo-labels such that it performs worse than NaCL, probably because the use of model predictions as a self-training manner could lead to error accumulation. By contrast, in NaCL, the use of clustering to obtain pseudo-labels is not directly involved in the model training such that the topological structure of target data is not damaged, and in turn the model training promotes the clustering assumption. Inspired by the consistence regularization in FixMatch, NaCL further improves performance, compared to NaCL (w/o $\mathcal{L}_{\mathrm{T}}$), by training the model with target classification loss, although the improvement is slight.

4.7 Hyper-parameters analysis

We evaluate and analyze the effect of several important hyper-parameters in NaCL on two transfer tasks. The default hyper-parameters setting is: k is 3 in KNN graph, the contrastive loss strength γ is 1.

Density of k-nearest neighbor graph The value of k indicates the number of adjacent vertices in the k-nearest neighbor graph, which determines the density of the graph. We first analyze the effect of the size of k on the LCC-based selection, including the number of selected instances and the accuracy of the corresponding pseudo-labels, on two transfer tasks, as shown in Fig. 3. We observe that as the value of k increases, the number of instances in the LCC also increases, but the accuracy of the corresponding pseudo-labels

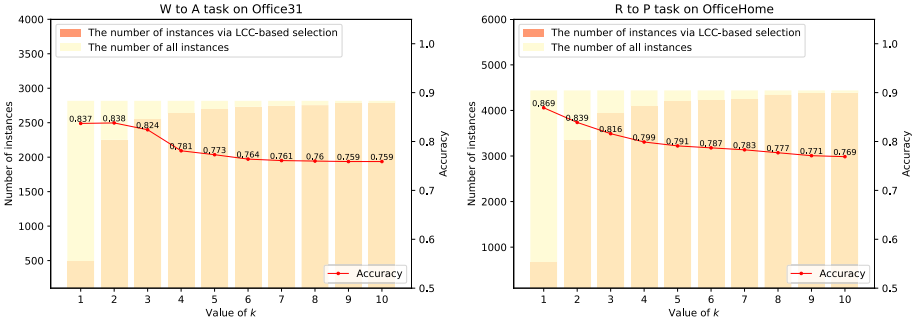

Fig. 3 The effect of different sizes of hyper-parameter k on LCC-based selection. **Left:** W to A task on Office31. **Right:** R to P task on OfficeHome

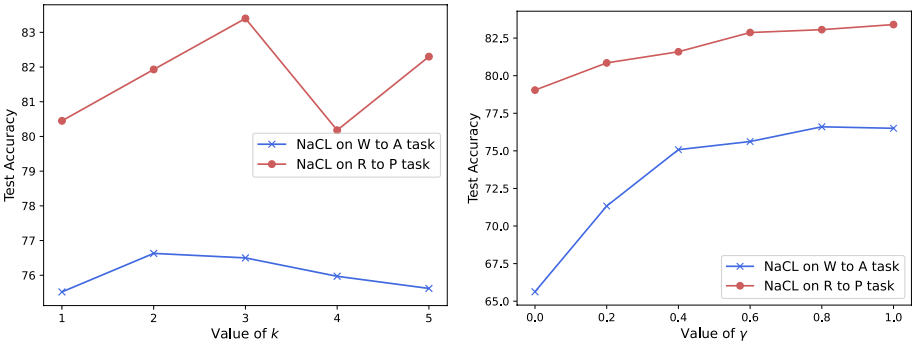

Fig. 4 Plot of various hyper-parameters analysis on NaCL with respect to two transfer tasks. **Left:** Varying the value of k which controls the density of k-nearest neighbor graph. **Right:** Varying γ, the strength of noise-robust cross-domain contrastive loss. $\gamma = 0$ can be roughly viewed as a semi-supervised method FixMatch

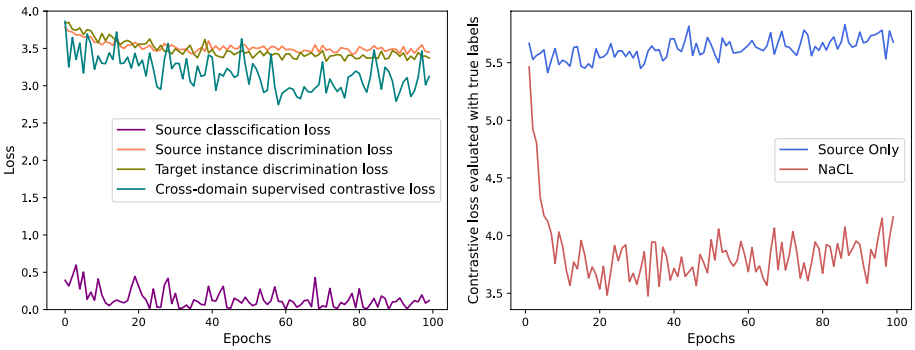

Fig. 5 **Left:** The variation of each component in the contrastive loss and the source classification loss as training proceeds on **VisDA-2017**. **Right:** The curve of contrastive loss evaluated with true labels during training on **P→R** task of OfficeHome

decreases. Further, the left panel of Fig. 4 presents the effect by varying the value of k on the test accuracy. The test accuracy of NaCL increases with the increase of k, and then declines. The larger k means that more instances in the LCC are obtained via LCC-based selection while more instances are removed via KNN-based selection. Additionally, large k would make the topology meaningless since the k-nearest neighbor graph on target data becomes too dense.

Strength of contrastive loss. Further, we study the importance of contrastive loss in NaCL by varying the value of γ in Eq.(6) and report the result in the right panel of Fig. 4. When $\gamma = 0$, NaCL without contrastive loss can be roughly viewed as semi-supervised method FixMatch. As γ increases, the performance of NaCL progressively grows, which reflects the efficacy of the proposed contrastive loss in NaCL.

4.8 Behavior analysis

Behavior of each component in contrastive loss The noise-robust cross-domain contrastive loss consists of intra-domain instance discrimination contrastive loss on source

and target data, respectively, and cross-domain supervised contrastive loss. The left panel of Fig. 5 reports the behavior of each component in noise-robust cross-domain contrastive loss during training on VisDA-2017 dataset. As can be seen, in addition to the source classification loss, each component in the noise-robust cross-domain contrastive loss gradually decreases as training proceeds. Although the cross-domain supervised contrastive loss oscillates due to the presence of noisy positive and negative pairs, it also decreases overall.

Contrastive loss evaluated with target true labels NaCL uses the pseudo-labels of target data to compute the contrastive loss as an optimization objective. In fact, we expect NaCL can reduce the contrastive loss evaluated with the true labels until convergence during training. Thus, we evaluate the contrastive loss with true labels on both source and target domains for NaCL and Source Only during training. The trend of contrastive loss evaluated with true labels is plotted in the right panel of Fig. 5. As seen, although we can only compute the contrastive loss evaluated with the pseudo-labels hypothesis on target domain during training, NaCL can effectively reduce the contrastive loss evaluated with true labels until convergence.

5 Conclusion and limitations

In this paper, we investigate how to design contrastive loss to solve UDA tasks. We propose NaCL that considers not only in-domain contrastive self-supervised learning but also cross-domain supervised contrastive learning to align class-conditional distribution between the source domain and target domain. In particular, NaCL adopts topology-based selection to detect and remove possibly false positive and negative pairs in contrastive loss. NaCL achieves state-of-the-art on three UDA benchmarks with different scales. Moreover, NaCL can also be applied to the diagnosis of COVID-19 in a semi-supervised domain adaptation scenario.

NaCL has the following potential limitations that would provide promising directions for future research, including: 1) we merely focus on using the ResNet-like models as the backbone in experiments, e.g., ResNet-50 and ResNet-101, yielding superior performance. Recently some works (Xu et al., 2022; Sun et al., 2022) have shown that large-scale pre-trained Vision Transformer (ViT) and its variants can also achieve advanced performance on the UDA task. Thus, how to design contrastive loss on ViT-like models to solve UDA task deserves further research; 2) we verify the efficacy of NaCL on image classification task. NaCL is extended to structured prediction tasks such as object detection, semantic segmentation, etc., as future work.

Appendix

Proof of Proposition 1

Proof Maximum likelihood is initially proposed to model clustering tasks. For unsupervised domain adaptation tasks, the objective of noise-robust cross-domain contrastive learning can be seen as adapting the model parameters θ trained on source data to maximize the log-likelihood function of the target domain data

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{n_t} \log p\left(\boldsymbol{x}_i \mid \theta\right) \tag{9}$$

We assume that the observed target domain data $\left\{\boldsymbol{x}_i\right\}_{i=1}^{n_t}$ are related to latent variable $\mathcal{C} = \left\{y_c\right\}_{c=1}^{|\mathcal{C}|}$ which denotes the true labels of data within the label space of $|\mathcal{C}|$ categories. We can re-write the log-likelihood function as

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{n_t} \log \sum_{y_c \in \mathcal{C}} p\left(\boldsymbol{x}_i, y_c \mid \theta\right). \tag{10}$$

It is difficult to optimize Eq.(10) directly, so we utilize a surrogate function to lower-bound the log-likelihood function

$$\begin{aligned} &\sum_{i=1}^{n_t} \log \sum_{y_c \in \mathcal{C}} p\left(\boldsymbol{x}_i, y_c \mid \theta\right) = \sum_{i=1}^{n_t} \log \sum_{y_c \in \mathcal{C}} \mathcal{Q}\left(y_c\right) \frac{p\left(\boldsymbol{x}_i, y_c \mid \theta\right)}{\mathcal{Q}\left(y_c\right)} \\ &\geq \sum_{i=1}^{n_t} \sum_{y_c \in \mathcal{C}} \mathcal{Q}\left(y_c\right) \log \frac{p\left(\boldsymbol{x}_i, y_c \mid \theta\right)}{\mathcal{Q}\left(y_c\right)} \end{aligned} \tag{11}$$

where $\mathcal{Q}\left(y_c\right)$ indicates a certain probability distribution for y_c. To make the inequality hold with equality, we require $\frac{p\left(\boldsymbol{x}_i, y_c|\theta\right)}{\mathcal{Q}\left(y_c\right)}$ to be a constant, based on which we have

$$\mathcal{Q}\left(y_c\right) = \frac{p\left(\boldsymbol{x}_i, y_c \mid \theta\right)}{\sum_{y_c \in \mathcal{C}} p\left(\boldsymbol{x}_i, y_c \mid \theta\right)} = \frac{p\left(\boldsymbol{x}_i, y_c \mid \theta\right)}{p\left(\boldsymbol{x}_i \mid \theta\right)} = p\left(y_c \mid \boldsymbol{x}_i, \theta\right) \tag{12}$$

Combining Eq.(11) and Eq.(12), and then ignoring the constant term, we should maximize the following equation, i.e., the expectation of the complete-data log-likelihood,

$$\sum_{i=1}^{n_t} \sum_{y_c \in \mathcal{C}} \mathcal{Q}\left(y_c\right) \log p\left(\boldsymbol{x}_i, y_c \mid \theta\right) \tag{13}$$

E-step. In this step, we use the current parameter θ^{old} to estimate $p\left(y_c \mid \boldsymbol{x}_i, \theta\right)$, i.e., pseudo-labels. To this end, we perform spherical k-means clustering on the features via encoder parameterized by θ^{old} to obtain $|\mathcal{C}|$ cluster assignments. As described in Section 3.4 of the manuscript, we compute $p\left(y_c \mid \boldsymbol{x}_i, \theta^{\text{old}}\right) = \mathbb{I}[\boldsymbol{x}_i \in \boldsymbol{o}_c]$ in which $\mathbb{I}[\boldsymbol{x}_i \in \boldsymbol{o}_c] = 1$ if $\boldsymbol{x}_i$ belongs to this cluster where $\boldsymbol{o}_c$ is its cluster center, otherwise $\mathbb{I}[\boldsymbol{x}_i \in \boldsymbol{o}_c] = 0$.

M-step. Based on E-step, we are ready to maximize Eq.(13) as follows.

$$\begin{aligned} \theta^{\text{new}} &= \arg\max_{\theta} \sum_{i=1}^{n_t} \sum_{y_c \in \mathcal{C}} p\left(y_c \mid \boldsymbol{x}_i, \theta^{\text{old}}\right) \log p\left(\boldsymbol{x}_i, y_c \mid \theta\right) \\ &= \arg\max_{\theta} \sum_{i=1}^{n_t} \sum_{y_c \in \mathcal{C}} \mathbb{I}[\boldsymbol{x}_i \in \boldsymbol{o}_c] \log p\left(\boldsymbol{x}_i, y_c \mid \theta\right). \end{aligned} \tag{14}$$

Under the assumption that the class prior obeys a uniform distribution, we have

$$p\left(\boldsymbol{x}_i, y_c \mid \theta\right) = p\left(\boldsymbol{x}_i \mid y_c, \theta\right) p\left(y_c \mid \theta\right) = \frac{1}{|\mathcal{C}|} p\left(\boldsymbol{x}_i \mid y_c, \theta\right) \tag{15}$$

Likewise, we assume that the distribution around each class center is an isotropic Gaussian, which lead to

$$p(\boldsymbol{x}_i \mid y_c, \theta) = \exp\left(\frac{-\left(\boldsymbol{q}_i - \boldsymbol{o}_c\right)^2}{2\sigma_c^2}\right) / \sum_{j=1}^{|C|} \exp\left(\frac{-\left(\boldsymbol{q}_i - \boldsymbol{o}_j\right)^2}{2\sigma_j^2}\right) \tag{16}$$

where the query $\boldsymbol{q}_i$ is the output of instance $\boldsymbol{x}_i$ in projection head, and the class center $\boldsymbol{o}_c$ can be regarded as the cluster center of instance $\boldsymbol{x}_i$. We use ℓ_2-normalization for vectors $\boldsymbol{q}$ and $\boldsymbol{o}$, and then we get $(\boldsymbol{q} - \boldsymbol{o})^2 = 2 - 2\boldsymbol{q} \cdot \boldsymbol{o}$. Combining this with Eqs.(10)(11)(13)(14)(15)(16), we can re-write maximum log-likelihood estimation as

$$\theta^{\text{new}} = \arg\min_{\theta} \sum_{i=1}^{n_t} -\log \frac{\exp\left(\boldsymbol{q}_i \cdot \boldsymbol{o}_c / \tau\right)}{\sum_{j=1}^{|C|} \exp\left(\boldsymbol{q}_i \cdot \boldsymbol{o}_j / \tau\right)} \tag{17}$$

where $\tau \propto \sigma^2$ stands for the density of the feature distribution around class center. We can see that the goal of Eq.(17) is to pull the query $\boldsymbol{q}_i$ closer to its class center, while staying away from other class centers.

Next we elaborate that the contrastive loss in our method can be empirically interpreted as optimizing Eq.(17). Specifically, given a query $\boldsymbol{q}_i$, the contrastive loss in our method can be written as

$$\mathcal{L}_{\text{NaCL}} = \min -\frac{1}{\mid P(i) \mid} \sum_{\boldsymbol{k}_i^+ \in P(i)} \log \frac{\exp\left(\boldsymbol{q}_i \cdot \boldsymbol{k}_i^+ / \tau\right)}{\sum_{\boldsymbol{k}_j \in A} \exp\left(\boldsymbol{q}_i \cdot \boldsymbol{k}_j / \tau\right)} \tag{18}$$

According to the previous assumptions, the positive keys in the set $P(i)$ should be distributed around the class center $\boldsymbol{o}_c$ of the query $\boldsymbol{q}_i$. Thus, we can derive an approximation w.r.t the positive keys of query $\boldsymbol{q}_i$ as follows,

$$\frac{1}{\mid P(i) \mid}\left(\boldsymbol{k}_{i_1}^+ + \boldsymbol{k}_{i_2}^+ + \cdots + \boldsymbol{k}_{i_{|P(i)|}}^+\right) \approx \boldsymbol{o}_c \tag{19}$$

By plugging Eq.(19) into Eq.(18), Eq.(18) can be re-written as

$$\mathcal{L}_{\text{NaCL}} \approx \min -\log \frac{\exp\left(\boldsymbol{q}_i \cdot \boldsymbol{o}_c / \tau\right)}{\sum_{\boldsymbol{k}_j \in A} \exp\left(\boldsymbol{q}_i \cdot \boldsymbol{k}_j / \tau\right)} \tag{20}$$

which has a similar form to the maximum log-likelihood estimation in Eq.(17) and both of them aim to pull the query $\boldsymbol{q}_i$ closer to its class center, while staying away from other class centers.

In summary, the optimization process of contrastive loss in our method can be considered an example of EM: At each epoch in training process, E-step aims to estimate the posterior probability of latent true labels via clustering, M-step aims to maximize the lower-bound of log-likelihood. □

Proof of Lemma 1

Proof Recall that the triangle inequality for classification error (Ben-David et al., 2007). Let $h \in \mathcal{H}$ be a hypothesis and $\mathcal{D}$ be any distribution over input space $\mathcal{X}$. Then $\forall h_1, h_2, h_3 \in \mathcal{H}$, the following triangle inequality holds

$$\epsilon_{\mathcal{D}}\left(h_1, h_2\right) \leq \epsilon_{\mathcal{D}}\left(h_1, h_3\right) + \epsilon_{\mathcal{D}}\left(h_3, h_2\right) \tag{21}$$

Theorem 1 shows that

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(P_S, P_T\right) + \lambda \tag{22}$$

in which $\lambda = \epsilon_T(h^*) + \epsilon_S(h^*)$ with $h^* = \arg\min_{h\in\mathcal{H}} \epsilon_T(h) + \epsilon_S(h)$. According to triangle inequality, we have

$$\epsilon_T(h^*) = \epsilon_T\left(h^*, f_T\right) \leq \epsilon_T\left(h^*, \hat{f}_T\right) + \epsilon_T\left(\hat{f}_T, f_T\right) \tag{23}$$

Combining Eq.(22) and Eq.(23), we derive

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(P_S, P_T\right) + \epsilon_T\left(\hat{f}_T, f_T\right) + \beta \tag{24}$$

in which $\beta = \epsilon_T(h^*, \hat{f}_T) + \epsilon_S(h^*)$ with $h^* = \arg\min_{h\in\mathcal{H}} \epsilon_T(h, \hat{f}_T) + \epsilon_S(h)$. □

Author contributions JL wrote and revised the manuscript, designed and implemented the research. HS contributed to the revision of the manuscript and the analysis of the results.

Funding This work was supported partly by National Natural Science Foundation under Grant Nos. (61972013, 61932007, 62141209) and partly by Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing.

Data availability The datasets are the benchmark datasets available online (Data Source available in manuscript).

Code availability https://github.com/jingzhengli/NaCL.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Bahri, D., Jiang, H., & Gupta, M. (2020). Deep k-nn for noisy labels. In *Proceedings of ICML*.

Bekkouch, I. E. I., Youssry, Y., & Gafarov, R., et al. (2019). Triplet loss network for unsupervised domain adaptation. *Algorithms*.

Ben-David, S., Blitzer, J., & Crammer, K., et al. (2007). Analysis of representations for domain adaptation. In *Proceedings of NeurIPS*.

Ben-David, S., Blitzer, J., & Crammer, K., et al. (2010). A theory of learning from different domains. *Machine Learning*.

Berthelot, D., Roelofs, R., & Sohn, K., et al. (2021). Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *Proceedings of ICLR*.

Bousmalis, K., Trigeorgis, G., & Silberman, N., et al. (2016). Domain separation networks. In *Proceedings of NeurIPS*.
Cao, Y., Xie, Z., & Liu, B., et al. (2020). Parametric instance classification for unsupervised visual feature learning. In *Proceedings of NeurIPS*.
Caron, M., Touvron, H., & Misra, I., et al. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of ICCV*.
Chen, C., Chen, Z., & Jiang, B., et al. (2019a). Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of AAAI*.
Chen, C., Fu, Z., & Chen, Z., et al (2020a). Homm: Higher-order moment matching for unsupervised domain adaptation. In *Proceedings of AAAI*.
Chen, C., Xie, W., & Huang, W., et al. (2019b). Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of CVPR*.
Chen, T., Kornblith, S., & Norouzi, M., et al. (2020b). A simple framework for contrastive learning of visual representations. In *Proceedings of ICML*.
Chen, X., Fan, H., & Girshick, R., et al. (2020c). Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297.
Chen, X., & He, K. (2021). Exploring simple siamese representation learning. In *Proceedings of CVPR*.
Chen, X., Wang, S., & Long, M., et al. (2019c). Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *Proceedings of ICML*.
Chen, Y., Pan, Y., & Wang, Y., et al. (2021). Transferrable contrastive learning for visual domain adaptation. In *Proceedings of ACM MM*.
Cicek, S., & Soatto, S. (2019). Unsupervised domain adaptation via regularized conditional alignment. In *Proceedings of ICCV*.
Deng, Z., Luo, Y., & Zhu, J. (2019). Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of ICCV*.
El Hamri, M., Bennani, Y., & Falih, I. (2022). Hierarchical optimal transport for unsupervised domain adaptation. *Machine Learning* 1–24.
French, G., Mackiewicz M., & Fisher, M. (2018). Self-ensembling for visual domain adaptation. In *Proceedings of ICLR*.
Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *Proceedings of ICML*.
Ganin, Y., Ustinova, E., & Ajakan, H., et al. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*.
Gao, W., Yang, B. B., & Zhou, Z. H. (2016). On the resistance of nearest neighbor to random noisy labels. arXiv preprint arXiv:1607.07526.
Ghasedi Dizaji, K., Herandi, A., & Deng, C., et al. (2017). Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of ICCV*.
Ghifary, M., Kleijn, W. B., & Zhang, M., et al. (2016). Deep reconstruction-classification networks for unsupervised domain adaptation. In *Proceedings of ECCV*.
Grill, J. B., Strub, F., & Altché, F., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. In *Proceedings of NeurIPS*.
Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *Proceedings of CVPR*.
He, K., Fan, H., & Wu, Y., et al. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of CVPR*.
He, K., Zhang, X., & Ren, S., et al. (2016a). Deep residual learning for image recognition. In *Proceedings of CVPR*.
He, K., Zhang, X., & Ren, S., et al. (2016b). Identity mappings in deep residual networks. In *Proceedings of ECCV*.
Henaff, O. (2020). Data-efficient image recognition with contrastive predictive coding. In *Proceedings of ICML*.
Hendrycks, D., Basart, S., & Mu, N., et al. (2021). The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of ICCV*.
Huang, J., Guan, D., & Xiao, A., et al. (2022). Category contrast for unsupervised domain adaptation in visual tasks. In *Proceedings of CVPR*.
Jiang, J., Fu, B., & Long, M. (2020). Transfer-learning-library.
Jin, Y., Wang, X., & Long, M., et al. (2020). Minimum class confusion for versatile domain adaptation. In *Proceedings of ECCV*.
Kang, G., Jiang, L., & Yang, Y., et al. (2019). Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of CVPR*.

Khosla, P., Teterwak, P., & Wang, C., et al. (2020). Supervised contrastive learning. In *Proceedings of NeurIPS*.
Kim, D., Saito, K., & Oh, T.H., et al. (2020). Cross-domain self-supervised learning for domain adaptation with few source labels. arXiv preprint arXiv:2003.08264.
Li, J., Zhou, P., & Xiong, C., et al. (2020). Prototypical contrastive learning of unsupervised representations. In *Proceedings of ICLR*.
Li, S., Xia, X., & Ge, S., et al. (2022). Selective-supervised contrastive learning with noisy labels. In *Proceedings of CVPR* (pp. 316–325).
Lin, H., Zhang, Y., & Qiu, Z., et al. (2022). Prototype-guided continual adaptation for class-incremental unsupervised domain adaptation. In *Proceedings of ECCV*.
Long, M., Cao, Z., & Wang, J., et al. (2018). Conditional adversarial domain adaptation. In *Proceedings of NeurIPS*.
Long, M., Zhu, H., & Wang, J., et al. (2017). Deep transfer learning with joint adaptation networks. In *Proceedings of ICML*.
Mansour, Y., Mohri, M., & Rostamizadeh, A. (2009). Domain adaptation: Learning bounds and algorithms. arXiv preprint arXiv:0902.3430.
Park, C., Lee, J., & Yoo, J., et al. (2020). Joint contrastive learning for unsupervised domain adaptation. arXiv preprint arXiv:2006.10297.
Pei, Z., Cao, Z., & Long, M., et al. (2018). Multi-adversarial domain adaptation. In *Proceedings of AAAI*.
Peng, X., Usman, B., & Kaushik, N., et al. (2017). Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924.
Saenko, K., Kulis, B., & Fritz, M., et al. (2010). Adapting visual category models to new domains. In *Proceedings of ECCV*.
Saito, K., Watanabe, K., & Ushiku, Y., et al. (2018). Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of CVPR*.
Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of CVPR*.
Sharma, A., Kalluri, T., & Chandraker, M. (2021). Instance level affinity-based transfer for unsupervised domain adaptation. In *Proceedings of CVPR*.
Shen, J., Qu, Y., & Zhang, W., et al. (2018). Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of AAAI*.
Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of NeurIPS*.
Sohn, K., & Berthelot, D. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Proceedings of NeurIPS*.
Sun, T., Lu, C., & Zhang, T., et al. (2022). Safe self-refinement for transformer-based domain adaptation. In *Proceedings of CVPR* (pp. 7191–7200).
Sun, Y., Tzeng, E., & Darrell, T., et al. (2019). Unsupervised domain adaptation through self-supervision. arXiv preprint arXiv:1909.11825.
Tang, H., Chen, K., & Jia, K. (2020). Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of CVPR*.
Thota, M., & Leontidis, G. (2021). Contrastive domain adaptation. In *Proceedings of CVPR*.
Tian, Y., Krishnan, D., & Isola, P. (2020). Contrastive multiview coding. In *Proceedings of ECCV*.
Tzeng, E., Hoffman, J., & Darrell, T., et al. (2015). Simultaneous deep transfer across domains and tasks. In *Proceedings of ICCV*.
Tzeng, E., Hoffman, J., & Saenko, K., et al. (2017). Adversarial discriminative domain adaptation. In *Proceedings of CVPR*.
Venkateswara, H., Eusebio, J., & Chakraborty, S., et al. (2017). Deep hashing network for unsupervised domain adaptation. In *Proceedings of CVPR*.
Wang, Q., & Breckon, T. (2020). Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. In *Proceedings of AAAI*.
Wang, R., Wang, G., & Henao, R. (2019). Discriminative clustering for robust unsupervised domain adaptation. arXiv preprint arXiv:1905.13331.
Wang, R., Wu, Z., & Weng, Z., et al. (2022). Cross-domain contrastive learning for unsupervised domain adaptation. *IEEE Transactions on Multimedia*.
Wiles, O., Gowal, S., & Stimberg, F., et al. (2021). A fine-grained analysis on distribution shift. In *Proceedings of ICLR*.
Wu, P., Zheng, S., & Goswami, M., et al. (2020). A topological filter for learning with label noise. In *Proceedings of NeurIPS*.

Wu, Z., Efros, A. A., & Yu, S. X. (2018a). Improving generalization via scalable neighborhood component analysis. In *Proceedings of ECCV*.
Wu, Z., Xiong, Y., & Yu, S. X., et al. (2018b). Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of CVPR*.
Wu, Z. F., Wei, T., & Jiang, J., et al. (2021). Ngc: A unified framework for learning with open-world noisy data. In *Proceedings of ICCV*.
Xiao, N., & Zhang, L. (2021). Dynamic weighted learning for unsupervised domain adaptation. In *Proceedings of CVPR*.
Xie, S., Zheng, Z., & Chen, L., et al. (2018). Learning semantic representations for unsupervised domain adaptation. In *Proceedings of ICML*.
Xu, M., Zhang, J., & Ni, B., et al. (2020). Adversarial domain adaptation with domain mixup. In *Proceedings of AAAI*.
Xu, R., Li, G., & Yang, J., et al. (2019). Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of ICCV*.
Xu, T., Chen, W., & Pichao, W., et al. (2022). Cdtrans: Cross-domain transformer for unsupervised domain adaptation. In *Proceedings of ICLR*.
Yao, H., Wang, Y., & Li, S., et al. (2022). Improving out-of-distribution robustness via selective augmentation. arXiv preprint arXiv:2201.00299.
Zhang, Y., Chen, H., & Wei, Y., et al. (2019a). From whole slide imaging to microscopy: Deep microscopy adaptation network for histopathology cancer image classification. In *International conference on medical image computing and computer-assisted intervention*.
Zhang, Y., Liu, T., & Long, M., et al. (2019b). Bridging theory and algorithm for domain adaptation. In *Proceedings of ICML*.
Zhang, Y., Niu, S., & Qiu, Z., et al. (2020). Covid-da: deep domain adaptation from typical pneumonia to covid-19. arXiv preprint arXiv:2005.01577.

Authors and Affiliations

Jingzheng Li[1,3] · Hailong Sun[2,3]

✉ Hailong Sun
sunhl@buaa.edu.cn

Jingzheng Li
jingzhengli@buaa.edu.cn

[1] SKLSDE Lab, School of Computer Science and Engineering, Beihang University, XueYuan Road No. 37, Beijing 100191, China

[2] SKLSDE Lab, School of Software, Beihang University, XueYuan Road No. 37, Beijing 100191, China

[3] Beijing Advanced Innovation Center for Big Data and Brain Computing, XueYuan Road No. 37, Beijing 100191, China