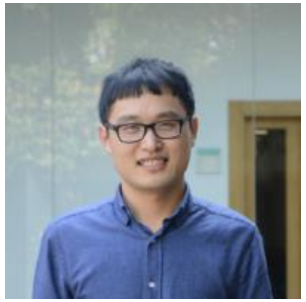
Exploiting the Intrinsic Neighborhood Structure for Source-free Domain Adaptation

Shiqi Yang[†], Yaxing Wang[†‡], Joost van de Weijer[†], Luis Herranz[†], Shangling Jui [*]

† Computer Vision Center UAB, Barcelona, Spain

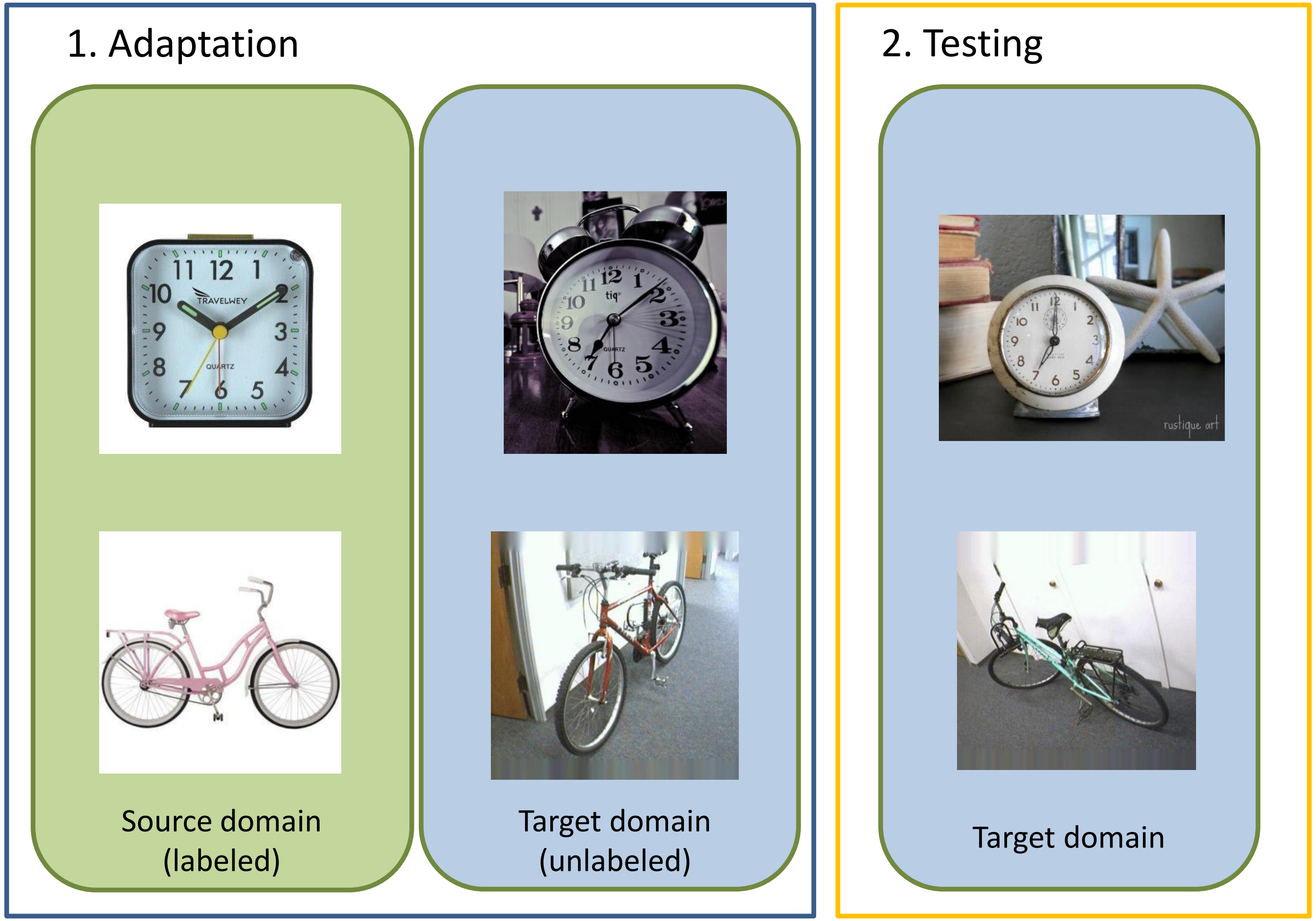
‡ PCALab, Nanjing University of Science and Technology, China

** Huawei Kirin Solution, Shanghai, China*

NeurIPS 2021

Introduction

(Unsupervised) Domain Adaptation

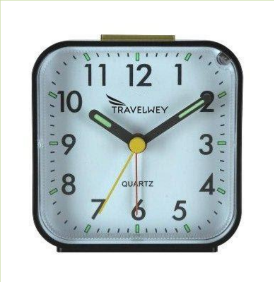


Problem: source data restricted due to data privacy, intellectual property

Introduction

Source-free Domain Adaptation (SFDA)

1. Pre-training



Source domain
(labeled)

2. Adaptation

Target domain
(unlabeled)

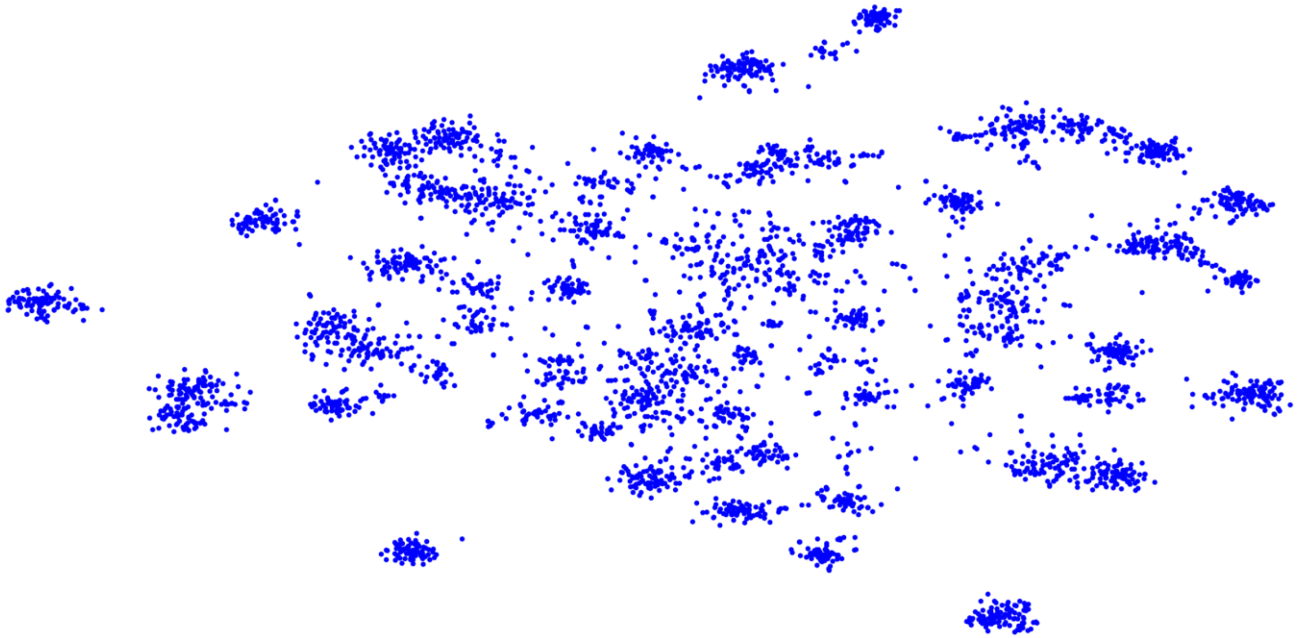
3. Testing

Target domain

Motivation

Premise: We already have the source-pretrained model

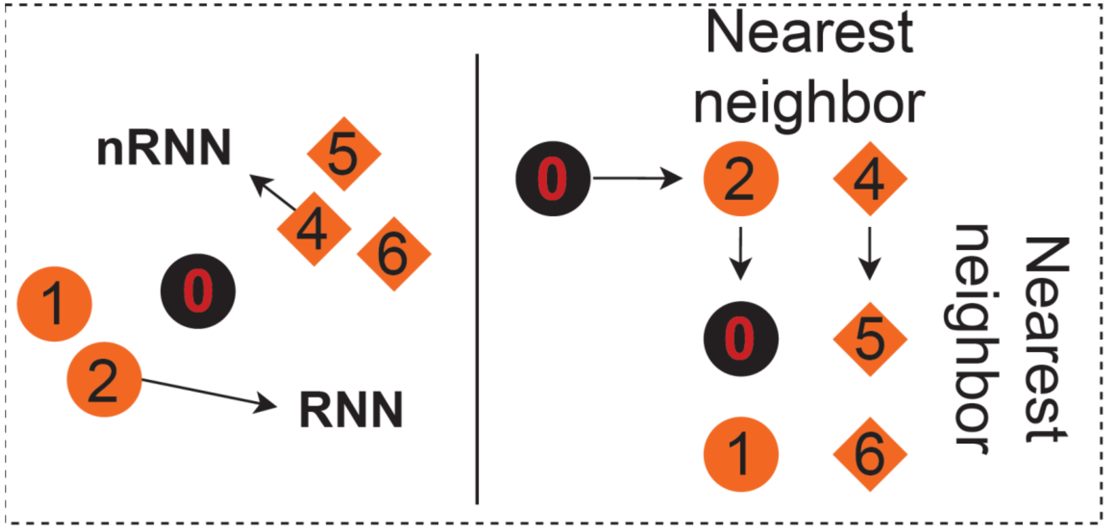
t-SNE visualization

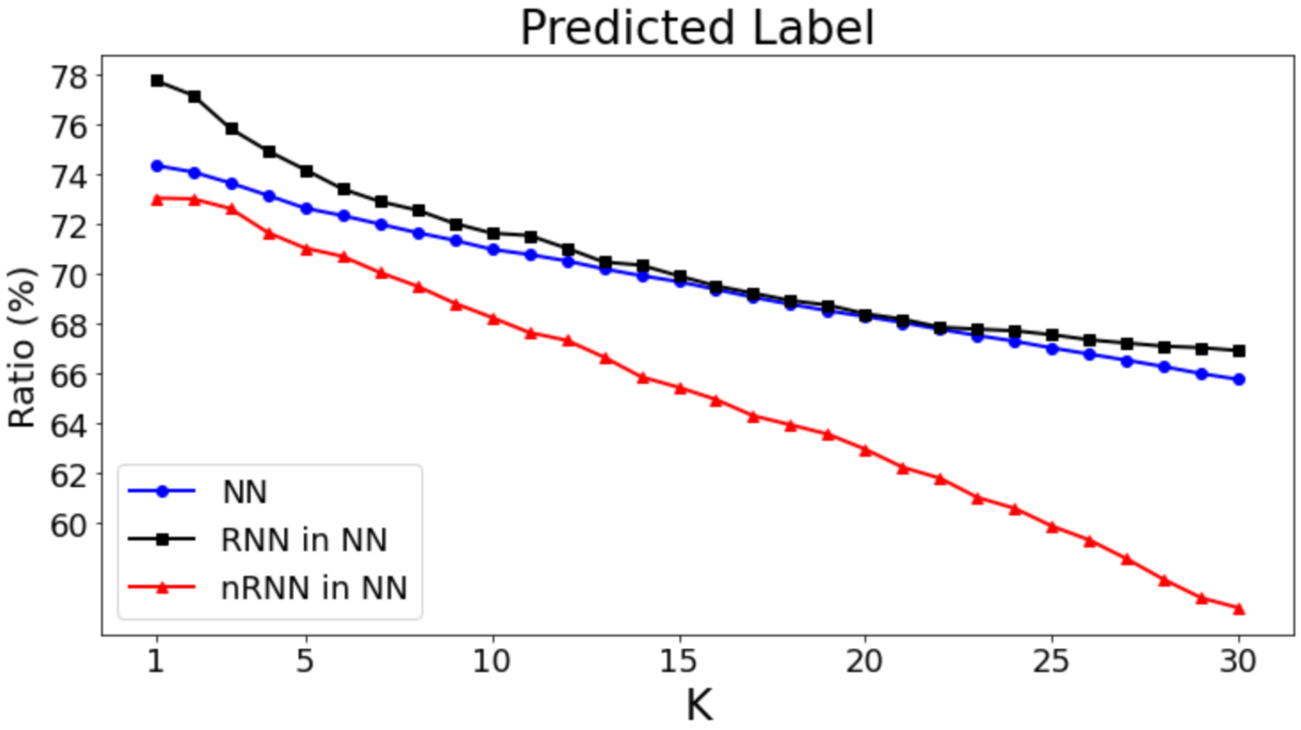


Observation 1: Target features from source pretrained model already form some clusters

Motivation 1: We can adopt neighborhood clustering for target adaptation

Motivation

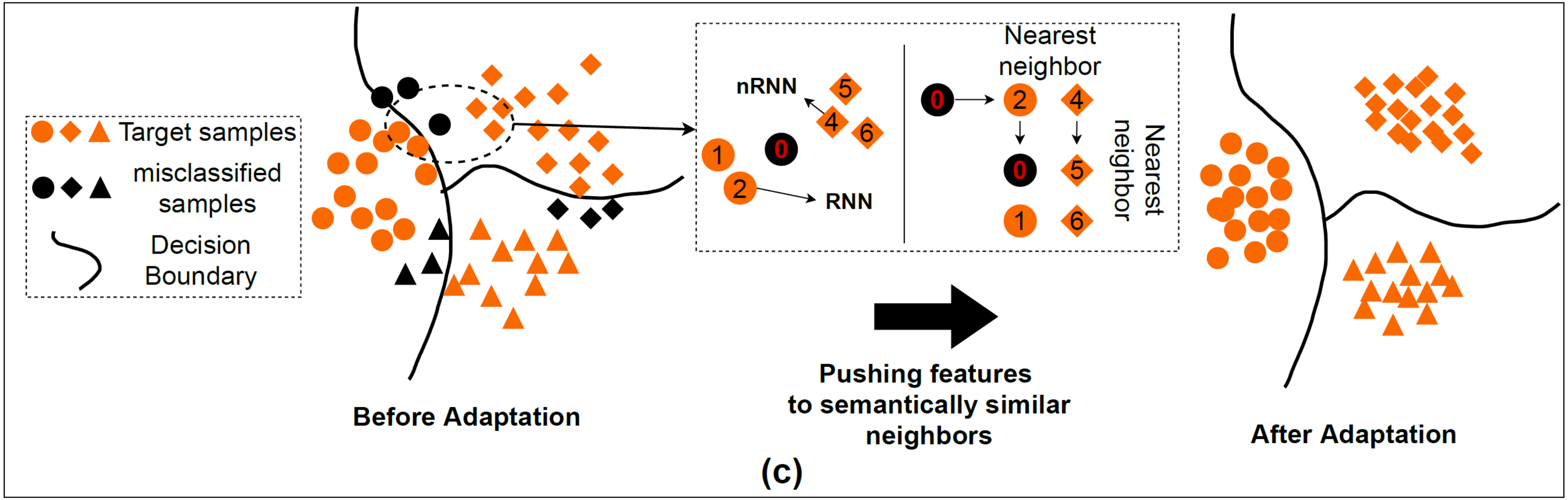




Observation 2: Reciprocal neighbors are more likely to have the **correct** predicted label

Motivation 2: We should assign higher credit to reciprocal neighbors.

Method



Method overview:

$$\mathcal{L} = -\frac{1}{n_t}\sum_{x_i \in \mathcal{D}_t}\sum_{x_j \in \mathrm{Neigh}(x_i)} \frac{D_{sim}(p_i, p_j)}{D_{dis}(x_i, x_j)}$$

Method

Premise: We already have the source-pretrained model

- **Nearest Neighbor Retrieving**

$$\mathcal{F} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{n_t}] \text{and } \mathcal{S} = [p_1, p_2, \ldots, p_{n_t}]$$

$\mathcal{F}$ stores all target features, and $\mathcal{S}$ stores corresponding prediction scores

- **Defining affinity by reciprocity**

$$A_{i,j} = \begin{cases} 1 & \text{if } j \in \mathcal{N}_K^i \wedge i \in \mathcal{N}_M^j \\ r & \text{otherwise.} \end{cases}$$

reciprocal

where r = 0.1

A_{ik} is the affinity value of k-th nearest neighbors of feature $\boldsymbol{z}_i$

Method

- **Neighborhood clustering by prediction-consistency**

$$\mathcal{L}_{\mathcal{N}} = -\frac{1}{n_t}\sum_{i}\sum_{k\in\mathcal{N}_K^i} A_{ik}\mathcal{S}_k^\top p_i$$

$\mathcal{N}_K^i$: index set of *K nearest neighbor* of feature i

To achieve target clustering with affinity as weight.

- **Self regularization**

constant!

$$\mathcal{L}_{self} = -\frac{1}{n_t}\sum_{i}^{n_t}\mathcal{S}_i^\top p_i$$

It aims to avoid the negative impact of potential neighbors.

Method

- **Diversity loss**

$$\mathcal{L}_{div} = \sum_{c=1}^{C} \mathrm{KL}(\bar{p}_c || q_c), \text{ with } \bar{p}_c = \frac{1}{n_t} \sum_{i} p_i^{(c)} \text{ , and } q_{\{c=1,..,C\}} = \frac{1}{C}$$

Encouraging the prediction to be balanced to avoid degeneration solution.

- **Expanded neighborhood**

$$E_M(\boldsymbol{z}_i) = \mathcal{N}_M(\boldsymbol{z}_j)\ \forall j \in \mathcal{N}_K(\boldsymbol{z}_i)$$

where E_M^k contain the M-nearest neighbors of neighbor k in $\mathcal{N}_K$.

$$\mathcal{L}_E = -\frac{1}{n_t} \sum_{i} \sum_{k \in \mathcal{N}_K^i} \sum_{m \in E_M^k} r \mathcal{S}_m^\top p_i$$

Method

Algorithm 1 Neighborhood Reciprocity Clustering for Source-free Domain Adaptation

```
Require: 𝒟_s (only for source model training), 𝒟_t
Pre-train model on 𝒟_s
Build feature bank ℱ and score bank 𝒮 for 𝒟_t
while Adaptation do
    Sample batch 𝒯 from 𝒟_t
    Update ℱ and 𝒮 corresponding to current batch 𝒯
    Retrieve nearest neighbors 𝒩 for each of 𝒯
    Compute affinity value A
    Retrieve expanded neighborhoods E for each of 𝒩
    Compute loss and update the model
end while
```

- **Final objective**

$$\mathcal{L} = \mathcal{L}_{div} + \mathcal{L}_{\mathcal{N}} + \mathcal{L}_E + \mathcal{L}_{self}$$

Experiment

- Results on Office-Home with ResNet50 as backbone

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	**Avg**
MCD [35]	✗	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN [24]	✗	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
SAFN [52]	✗	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
Symnets [58]	✗	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	74.5	52.6	82.7	67.6
MDD [59]	✗	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	**60.2**	82.3	68.1
TADA [47]	✗	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
BNM [4]	✗	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
BDG [53]	✗	51.5	73.4	78.7	65.3	71.5	73.7	65.1	49.7	81.1	74.6	55.1	84.8	68.7
SRDC [42]	✗	52.3	76.3	81.0	**69.5**	76.2	78.0	**68.7**	53.8	81.7	**76.3**	57.1	85.0	71.3
RSDA-MSTN [10]	✗	53.2	77.7	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	85.4	70.9
SHOT [21]	✓	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
NRC	✓	**57.7**	**80.3**	**82.0**	68.1	**79.8**	**78.6**	65.3	**56.4**	**83.0**	71.0	58.6	**85.6**	**72.2**

- Results on VisDA-C with ResNet101 as backbone

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
ADR [34]	✗	94.2	48.5	84.0	72.9	90.1	74.2	92.6	72.5	80.8	61.8	82.2	28.8	73.5
CDAN [24]	✗	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
CDAN+BSP [2]	✗	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
SAFN [52]	✗	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
SWD [19]	✗	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
MDD [59]	✗	-	-	-	-	-	-	-	-	-	-	-	-	74.6
DMRL [49]	✗	-	-	-	-	-	-	-	-	-	-	-	-	75.5
MCC [15]	✗	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
STAR [26]	✗	95.0	84.0	**84.6**	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
RWOT [51]	✗	95.1	80.3	83.7	**90.0**	92.4	68.0	**92.5**	82.2	87.9	78.4	**90.4**	**68.2**	84.0
3C-GAN [20]	✓	94.8	73.4	68.8	74.8	93.1	95.4	88.6	**84.7**	89.1	84.7	83.5	48.1	81.6
SHOT [21]	✓	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
NRC	✓	**96.8**	**91.3**	82.4	62.4	**96.2**	**95.9**	86.1	80.6	**94.8**	**94.1**	90.4	59.7	**85.9**

Experiment

- Results on PointDA (3D point-cloud dataset)

	SF	Model→Shape	Model→Scan	Shape→Model	Shape→Scan	Scan→Model	Scan→Shape	Avg
MMD [25]	✗	57.5	27.9	40.7	26.7	47.3	54.8	42.5
DANN [6]	✗	58.7	29.4	42.3	30.5	48.1	56.7	44.2
ADDA [44]	✗	61.0	30.5	40.4	29.3	48.9	51.1	43.5
MCD [35]	✗	62.0	31.0	41.4	31.3	46.8	59.3	45.3
PointDAN [30]	✗	64.2	**33.0**	47.6	**33.9**	49.1	64.1	48.7
Source-only		43.1	17.3	40.0	15.0	33.9	47.1	32.7
NRC	✓	**64.8**	25.8	**59.8**	26.9	**70.1**	**68.1**	**52.6**

Experiment

- Ablation study on Office-Home and VisDA

$\mathcal{L}_{div}$	$\mathcal{L}_{\mathcal{N}}$	$\mathcal{L}_{E}$	$\mathcal{L}_{\hat{\boldsymbol{E}}}$	A	Avg
					59.5
✓					62.1
✓	✓				67.1
✓	✓			✓	69.1
✓	✓	✓			65.2
✓	✓	✓		✓	**72.2**
✓	✓		✓	✓	69.1

Office-Home

$\mathcal{L}_{div}$	$\mathcal{L}_{\mathcal{N}}$	$\mathcal{L}_{E}$	$\mathcal{L}_{\hat{\boldsymbol{E}}}$	A	Acc
					44.6
✓					47.8
✓	✓				74.6
✓	✓			✓	81.5
✓	✓	✓			61.2
✓	✓	✓		✓	**85.9**
✓	✓		✓	✓	82.0

VisDA

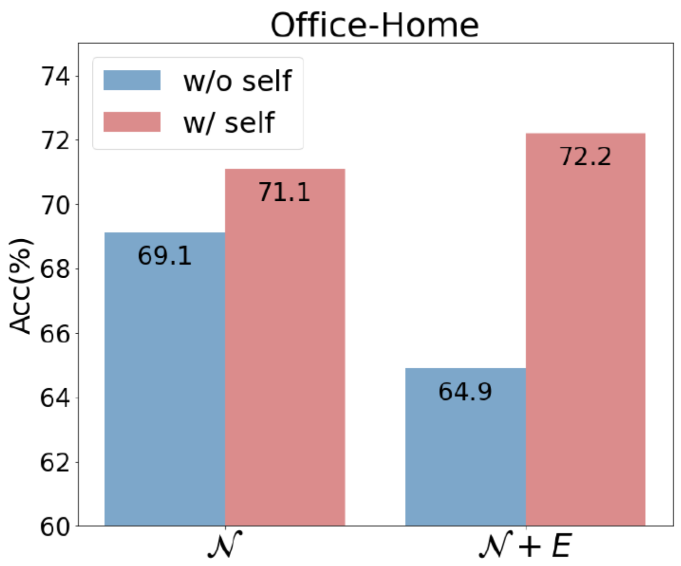
Method&Dataset	Acc
VisDA (K=M=5)	**85.9**
VisDA w/o E (K=30)	84.0
OH (K=3,M=2)	**72.2**
OH w/o E (K=9)	69.5

VisDA & Office-Home

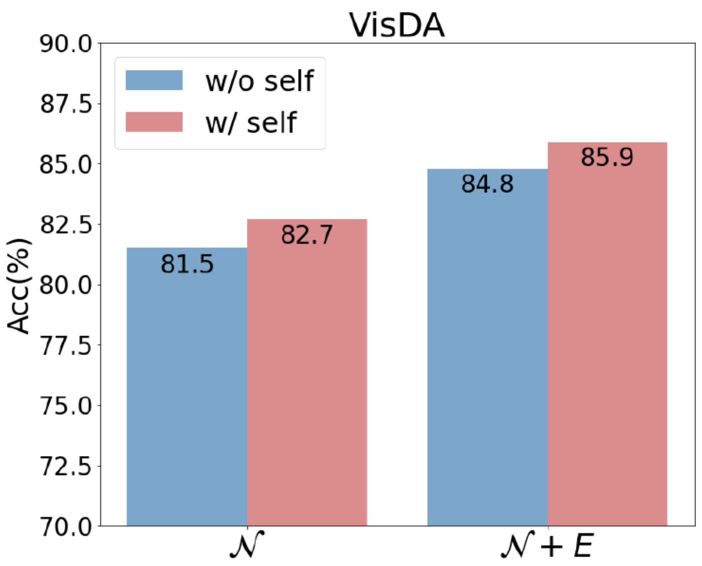
$\mathcal{L}_{\hat{\boldsymbol{E}}}$: Removing duplication in expanded neighbors

Experiment

- Ablation study on Office-Home and VisDA

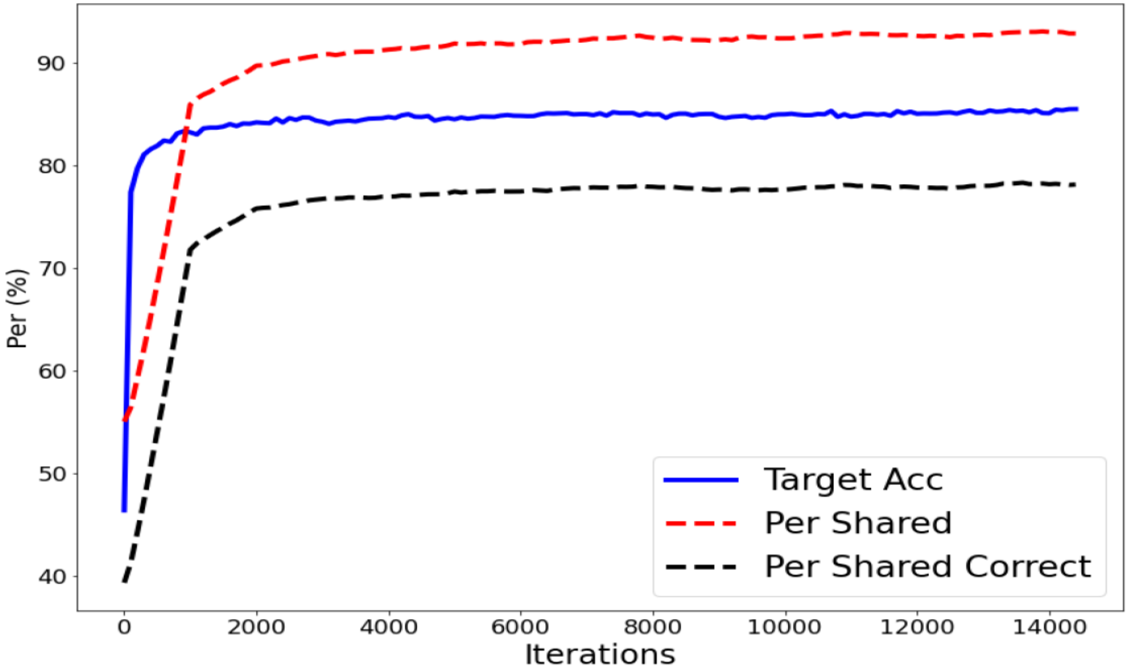


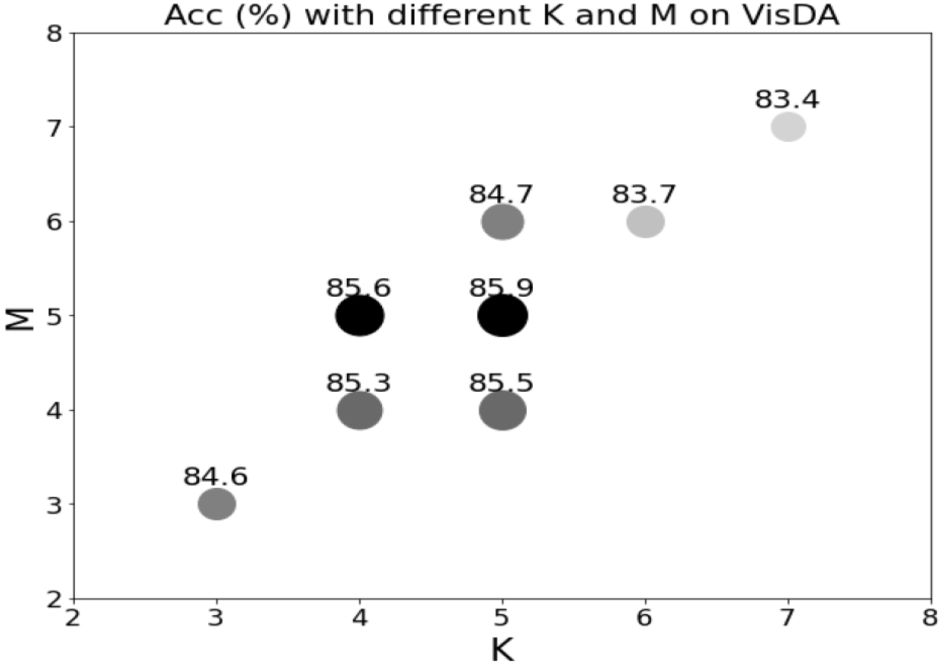
(a)



(b)

Experiment





Per Shared : ratio of features which have 5-nearest neighbors all sharing the same predicted label

Per Shared Correct : ratio of features which have 5-nearest neighbors all sharing the same **correct** predicted label

Experiment

- Runtime experiment on VisDA (with one TITAN-Xp)

VisDA	Runtime (s/epoch)	Per-class (%)
SHOT	618.82	82.9
NRC	540.89	85.9
NRC(20% for memory bank)	507.15	85.3
NRC(10% for memory bank)	499.49	85.2
NRC(5% for memory bank)	499.28	85.1

Instead of storing all features, we store a fixed number of target features and prediction scores (as a queue, first in first out).

Conclusion

- We propose to use neighborhood clustering to tackle source-free domain adaptation problem:
    - Reciprocal and non-reciprocal neighbors
    - Self regularization
    - Expanded neighbors

- State-of-the-art performance on several 2D and 3D point cloud datasets, without access to source data.

Thank you

paper

code