SF(DA)²: Source-free Domain Adaptation Through the Lens of Data Augmentation

Uiwon Hwang[1*], Jonghyun Lee[2], Juhyeon Shin[2], Sungroh Yoon[2]

[1] Yonsei University
[2] Seoul National University

ICLR

BDSL Biomedical Data Science Lab.

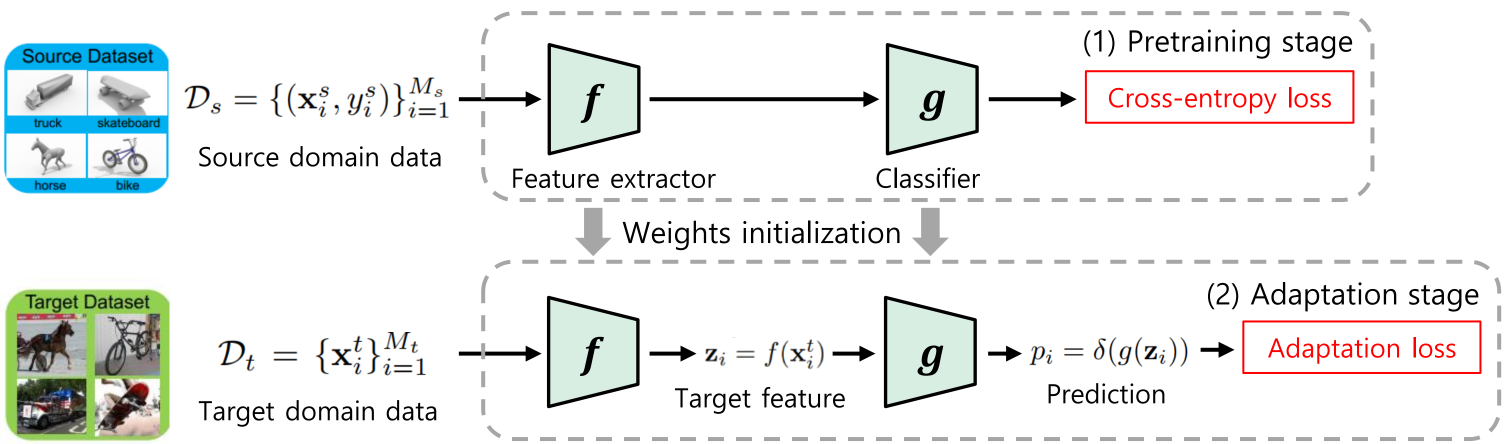
DSAIL Data Science & AI Laboratory

*speaker

Domain Adaptation

- **Domain adaptation** (DA)
  - **Domain shift** or covariate shift problem deteriorates the performance of the model
  - Adapting a model trained on labeled source domain data to **unlabeled target domain** data

    e.g. Using a self-driving model trained under sunny conditions for application during **rainy days**
- **Source-free domain adaptation** (SFDA)
  - Source domain data can be **inaccessible or difficult** to obtain
    - Cost, privacy concern, ...
  - SFDA uses only a **model pretrained** on the source domain data

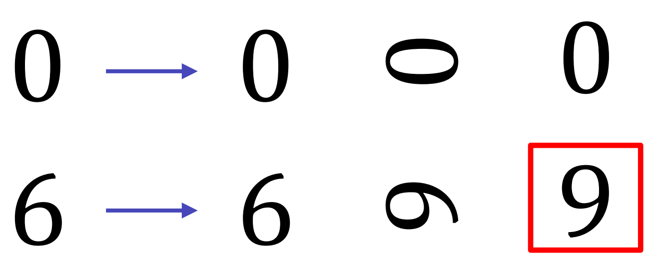


Liang, Jian, Dapeng Hu, and Jiashi Feng. "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation." ICML 2020.

Data Augmentation

- **Data augmentation** (another DA)
  - Increasing the diversity of the training dataset by applying **<u>transformations</u>**
  - Improving the generalization performance of the model

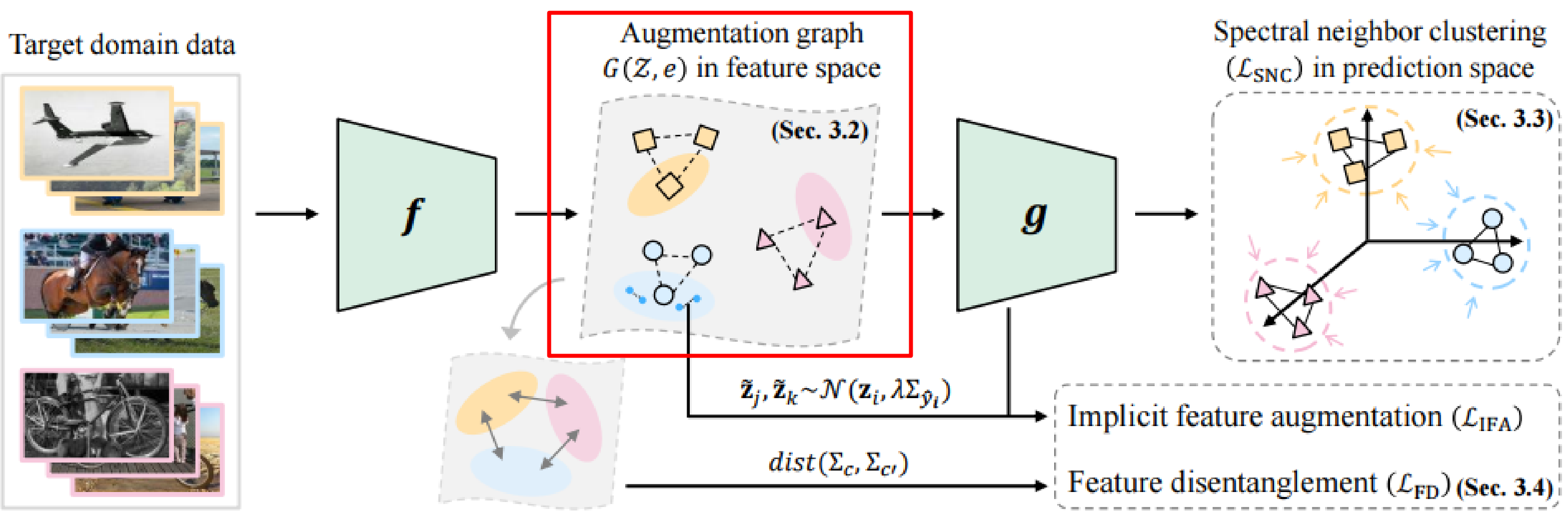
- Reliance on **<u>domain knowledge</u>**
  - Not using **<u>class-preserving</u>** transformations can lead to a decrease in model performance
  - Predefined transformations require **<u>strong domain expertise</u>**



Method

- **Augmentation graph on feature space**
  - Clustering assumption of source model
    - Target domain data that share the **same semantic information** are mapped to their **neighbors** in the feature space of the pretrained model
  - Augmentation assumption of target domain data
    - Target domain data **sharing class semantic information** may have highly nonlinear functions to **transform each other**
  - Population augmentation graph $G(\mathcal{Z}, e)$, where $\begin{cases} \mathcal{Z} = \{\mathbf{z} = f(\mathbf{x}^t) | \mathbf{x}^t \sim P(\mathcal{X}^t)\} \\ e_{ij} = e(\mathbf{z}_i, \mathbf{z}_j) = Pr(\mathbf{z}_j \in N_i) \end{cases}$

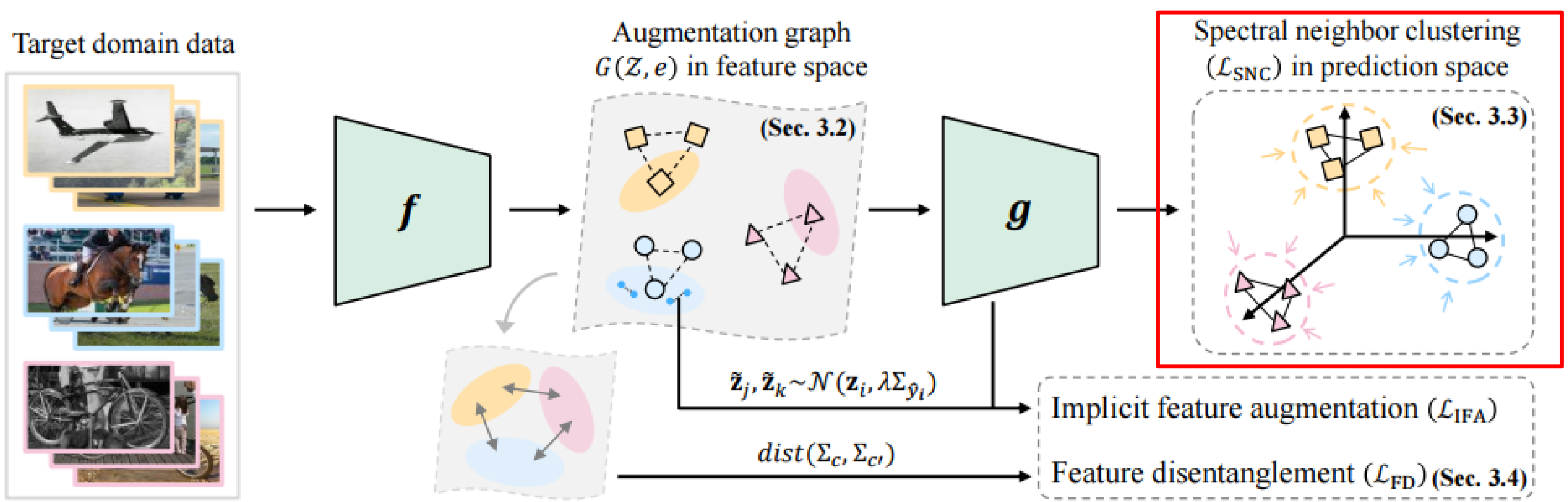
set of neighbors of $\mathbf{z}_i$



Method

- **Finding partition on prediction space**
  - We build an instance of population augmentation graph $\hat{G}$ using target domain data
    - We consider K-nearest neighbors of $\mathbf{z}_i$, denoted by N_i^K, in the feature memory bank $\mathcal{F}$
  - Then, we employ spectral clustering on the graph
  - **<u>Spectral neighborhood clustering (SNC)</u>** loss on $\hat{G}$
    - Identify partitions in the augmentation graph

$$\mathcal{L}_{\text{SNC}}(p_i) = -\frac{2}{K}\sum_{j\in N_i^K} p_i^T p_j + \sum_{k\in B}\left(p_i^T p_k\right)^2$$

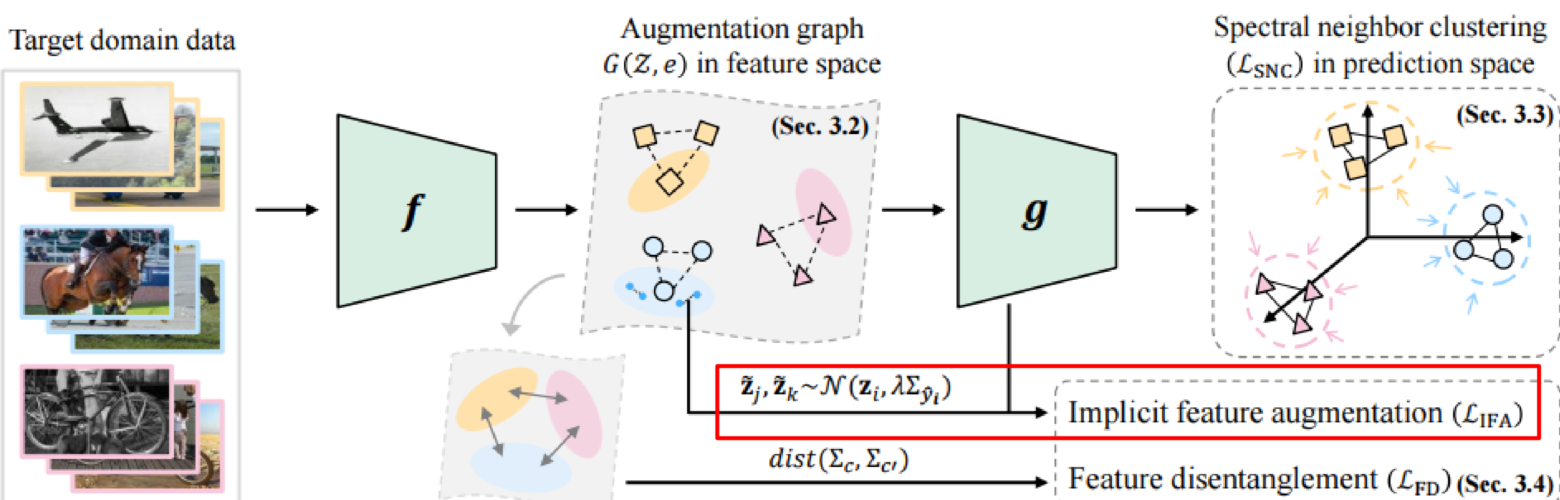


Method

- **Implicit feature augmentation** (IFA)
  - We aim to Simulate the effect of an **unlimited number of augmented features**
    - With **minimal computational and memory overhead**
  - First, we augment target features using estim $\tilde{\mathbf{z}}_j, \tilde{\mathbf{z}}_k, \cdots \sim \mathcal{N}(\mathbf{z}_i, \lambda\Sigma_{\hat{y}_i})$ covariance matrices based on pseudo-labels:
  - Then, we derive the upper bound for the expected (logarithm of) SNC loss

$$\mathcal{L}^{\infty}_{\text{EFA}}(\mathbf{z}_i; f, g) = \mathbb{E}_{\tilde{\mathbf{z}}_j \sim \mathcal{N}(\mathbf{z}_i, \lambda\Sigma_i)}\left[\mathbb{E}_{\tilde{\mathbf{z}}_k \sim \mathcal{N}(\mathbf{z}_i, \lambda\Sigma_i)}\left[-\log \tilde{p}_j^T \tilde{p}_k\right]\right]$$

$$\leq -2\sum_{c=1}^{C} \log \frac{\exp(g(\mathbf{z}_i)_c)}{\sum_{c'=1}^{C} \exp\left(g(\mathbf{z}_i)_{c'} + \frac{\lambda}{2}(w_{c'} - w_c)^T \Sigma_{\hat{y}_i}(w_{c'} - w_c)\right)} = \mathcal{L}_{\text{IFA}}(\mathbf{z}_i, \Sigma_{\hat{y}_i}, g)$$

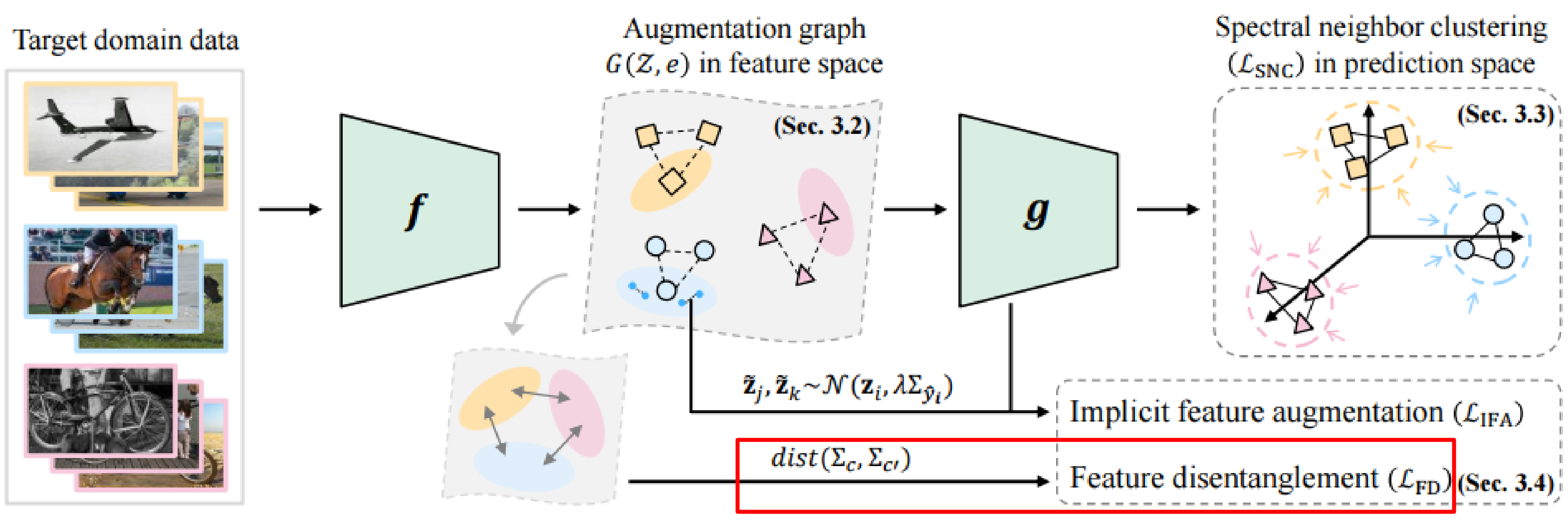


Method

- **Feature disentanglement** (FD)
  - Encourage each direction in the feature space to represent different semantics
  - Maximize the **cosine distance** between covariance matrices corresponding to **similar classes**

$$\mathcal{L}_{\mathrm{FD}} = -\frac{1}{2}\sum_{i,j} a_{ij}\left(1 - \frac{\mathrm{tr}\{\Sigma_i\,\Sigma_j\}}{\|\Sigma_i\|_F\|\Sigma_j\|_F}\right)$$

$$a_{ij} = \bar{p}_i^T\bar{p}_j, \text{ where } \bar{p}_c = \frac{1}{|\{i:\hat{y}_i=c\}|}\sum\nolimits_{i\in\{i:\hat{y}_i=c\}} p_i$$



Method

- **Final objective**

$$\min_{f,g} \mathcal{L}_{\text{SNC}} + \alpha_1 \mathcal{L}_{\text{IFA}} + \alpha_2 \mathcal{L}_{\text{FD}}$$

- **Pseudo code**

Algorithm 1 Adaptation procedure of SF(DA)^2

Require: f and g (trained on $\mathcal{D}_s$), $\mathcal{D}_t = \{\mathbf{x}_i^t\}_{i=1}^{M_t}$

```
while training loss is not converged do
    if epoch start then
        Update a_ij for FD loss
    end if
    Sample batch B from D_t and update F, S
    Retrieve neighbors N_i^K for each z_i in B
    Update f and g using SGD
        ∇_{f,g} L_SNC + α1 L_IFA + α2 L_FD
end while
```

Experiments

- **Evaluation results**

Table 1: Accuracy (%) on the VisDA dataset (ResNet-101).

Method	SF	plane	bicycle	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
BSP [4]	✗	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 [34]	✗	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
MCC [11]	✗	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
FixBi [19]	✗	96.1	87.8	90.5	90.3	96.8	95.3	92.8	88.7	97.2	94.2	90.9	25.7	87.2
Source only [9]	-	60.9	21.6	50.9	67.6	65.8	6.3	82.2	23.2	57.3	30.6	84.6	8.0	46.6
3C-GAN [15]	✓	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 [17]	✓	94.6	87.5	80.4	59.5	92.9	95.1	83.1	80.2	90.9	89.2	85.8	56.9	83.0
NRC [35]	✓	96.1	90.8	83.9	61.5	95.7	95.7	84.4	80.7	94.0	91.9	89.0	59.5	85.3
CoWA-JMDS [14]	✓	96.2	90.6	84.2	75.5	96.5	97.1	88.2	85.6	94.9	93.0	89.2	53.5	87.0
AaD [37]	✓	96.8	89.3	83.8	82.8	96.5	95.2	90.0	81.0	95.7	92.9	88.9	54.6	<u>87.3</u>
DaC [38]	✓	96.6	86.8	86.4	78.4	96.4	96.2	93.6	83.8	96.8	95.1	89.6	50.0	<u>87.3</u>
SF(DA)^2	✓	96.8	89.3	82.9	81.4	96.8	95.7	90.4	81.3	95.5	93.7	88.5	64.7	**88.1**

Table 2: Accuracy (%) on 7 domain shifts of the DomainNet-126 dataset (ResNet-50).

Method	SF	S→P	C→S	P→C	P→R	R→S	R→C	R→P	Avg.
MCC [11]	✗	47.3	34.9	41.9	72.4	35.3	44.8	65.7	48.9
Source only [9]	-	50.1	46.9	53.0	75.0	46.3	55.5	62.7	55.6
TENT [32]	✓	52.4	48.5	57.9	67.0	54.0	58.5	65.7	57.7
SHOT [17]	✓	66.1	60.1	66.9	80.8	59.9	67.7	68.4	67.1
AdaContrast [3]	✓	65.9	58.0	68.6	80.5	61.5	70.2	69.8	<u>67.8</u>
SF(DA)^2	✓	67.7	59.6	67.8	83.5	60.2	68.8	70.5	**68.3**

Experiments

- **Imbalanced dataset**
  - VisDA-RSUT
    - Labels of source domain and target domain have opposite long-tail distributions

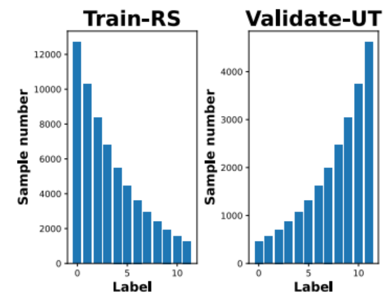
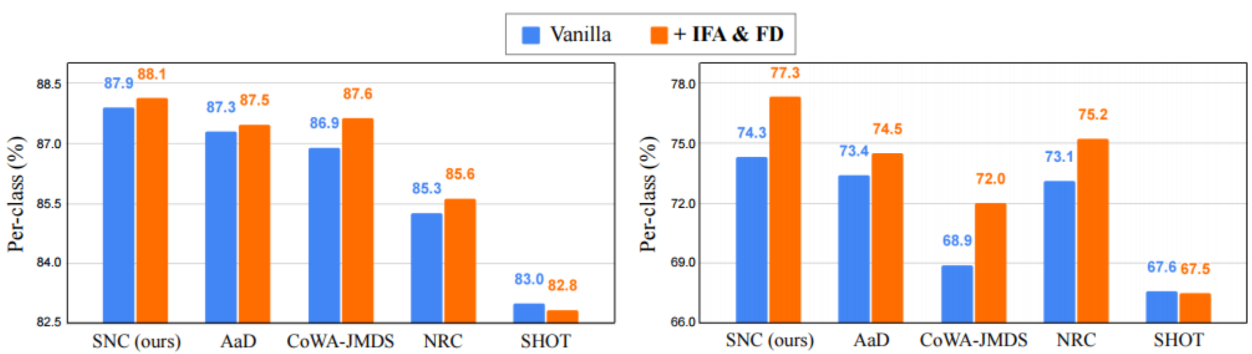


Table 4: Accuracy (%) on the VisDA-RSUT dataset (ResNet-101).

Method	SF	plane	bicycle	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
DANN [7]	✗	71.7	35.7	58.5	21.0	80.9	73.0	45.7	23.7	12.2	4.3	1.5	0.9	35.8
BSP [4]	✗	100.0	57.1	68.9	56.8	83.7	26.7	78.7	16.2	63.7	1.9	0.1	0.1	46.2
MCD [26]	✗	63.0	41.4	84.0	67.3	86.6	93.9	85.6	76.3	84.1	11.3	5.0	3.0	58.5
Source only [9]	-	79.7	15.7	40.6	77.2	66.8	11.1	85.1	12.9	48.3	14.3	64.6	3.3	43.3
SHOT [17]	✓	86.2	48.1	77.0	62.8	92.0	66.2	90.7	61.3	76.9	73.5	67.2	9.1	67.6
CoWA-JMDS [14]	✓	63.8	32.9	69.5	59.9	93.2	95.4	92.3	69.4	85.1	68.4	64.9	32.3	68.9
NRC [35]	✓	86.2	47.6	66.7	68.1	94.7	76.6	93.7	63.6	87.3	89.0	83.6	20.5	73.1
AaD [37]	✓	73.9	33.3	56.6	71.4	90.1	97.0	91.9	70.8	88.1	87.2	81.2	39.4	73.4
SF(DA)^2	✓	79.0	43.3	73.6	74.7	92.8	98.3	93.4	79.1	90.1	87.5	81.1	34.2	**77.3**

- The proposed IFA loss significantly improves the performance of existing methodologies as well as SNC in imbalanced SFDA



(a) VisDA **(b)** VisDA-RSUT

Thank you!

- **TL;DR**
  - a novel SFDA method that leverages intuitions derived from data augmentation

- **Summary**
  - Provide a fresh perspective on SFDA by interpreting it through the lens of **data augmentation**
  - Propose the **spectral neighborhood clustering (SNC)** loss and derive the **implicit feature augmentation (IFA)** using the augmentation graph in the feature space
  - Outperform existing methods under SFDA settings, especially with **imbalanced classes**

- More details can be found in our paper and code!