d-SNE: Domain Adaptation using Stochastic Neighborhood Embedding

Xiang Xu*, Xiong Zhou*, Ragav Venkatesan, Gurumurthy Swaminathan, Orchid Majumder

CBL
Changing the way ~~people~~ computers look at ~~computers~~ people

Computational Biomedicine Lab, University of Houston

Amazon AI, Seattle

aws

* First two authors contribute equally

Problem Statement

- Source domain $D^s = \{x^s, y^s\}_{i=1}^{N^s}$
- Target domain $D^t = \{x^t, y^t\}_{j=1}^{N^t}, N^t \ll N^s$ or $D^t = \{x^t\}_{j=1}^{N^t}$

- Goal: Improve the performance of an existing model M_{D^s} for D^t by adapting the knowledge of the model learned from D^s to D^t

Proposed Method: d-SNE

- Consider the distance between the features from the source and target domain

$$d(x^s, x^t) = \left|\left|\phi_{D^s}(x^s) - \phi_{D^t}(x^t)\right|\right|_2^2$$

- The probability that target samples $x_j^t \in D^t$ has the same label as the source samples $x_i^s \in D^s$

$$p_{ij} = \frac{e^{-d(x_i^s, x_j^t)}}{\sum_{x \in D^s} e^{-d(x, x_j^t)}} \xrightarrow{[1][2]} p_j = \frac{\sum_{x \in D_k^s} e^{-d(x, x_j^t)}}{\sum_{x \in D^s} e^{-d(x, x_j^t)}}, D_k^s = \{\forall x_l^s | y_l^s = k\}$$

[1] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. NIPS, 2003
[2] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighbourhood Components Analysis, NIPS, 2005

Proposed Method: d-SNE (2)

- Given p_j for one sample, the objective function for the domain adaptation problem can be derived as minimizing the ratio of intra-class distances to inter-class distance in the latent space.

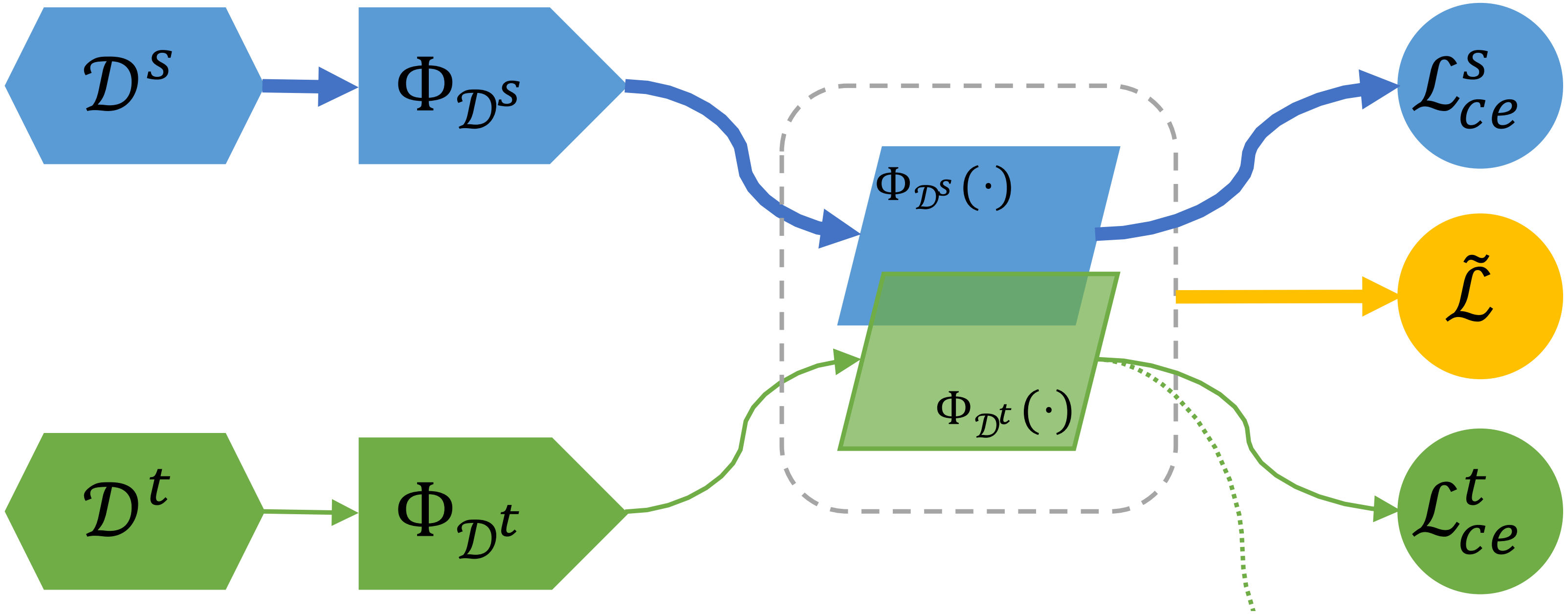
$$L = \log\left(\frac{\sum_{x \in D_{\bar{k}}^s} e^{-d(x, x_j)}}{\sum_{x \in D_k^s} e^{-d(x, x_j)}}\right), k = y_j$$

- Relaxation

$$\tilde{L} = \sup_{x \in D_k^s}\{a | a \in d(x, x_j)\} - \inf_{x \in D_{\bar{k}}^s}\{a | a \in d(x, x_j)\}$$

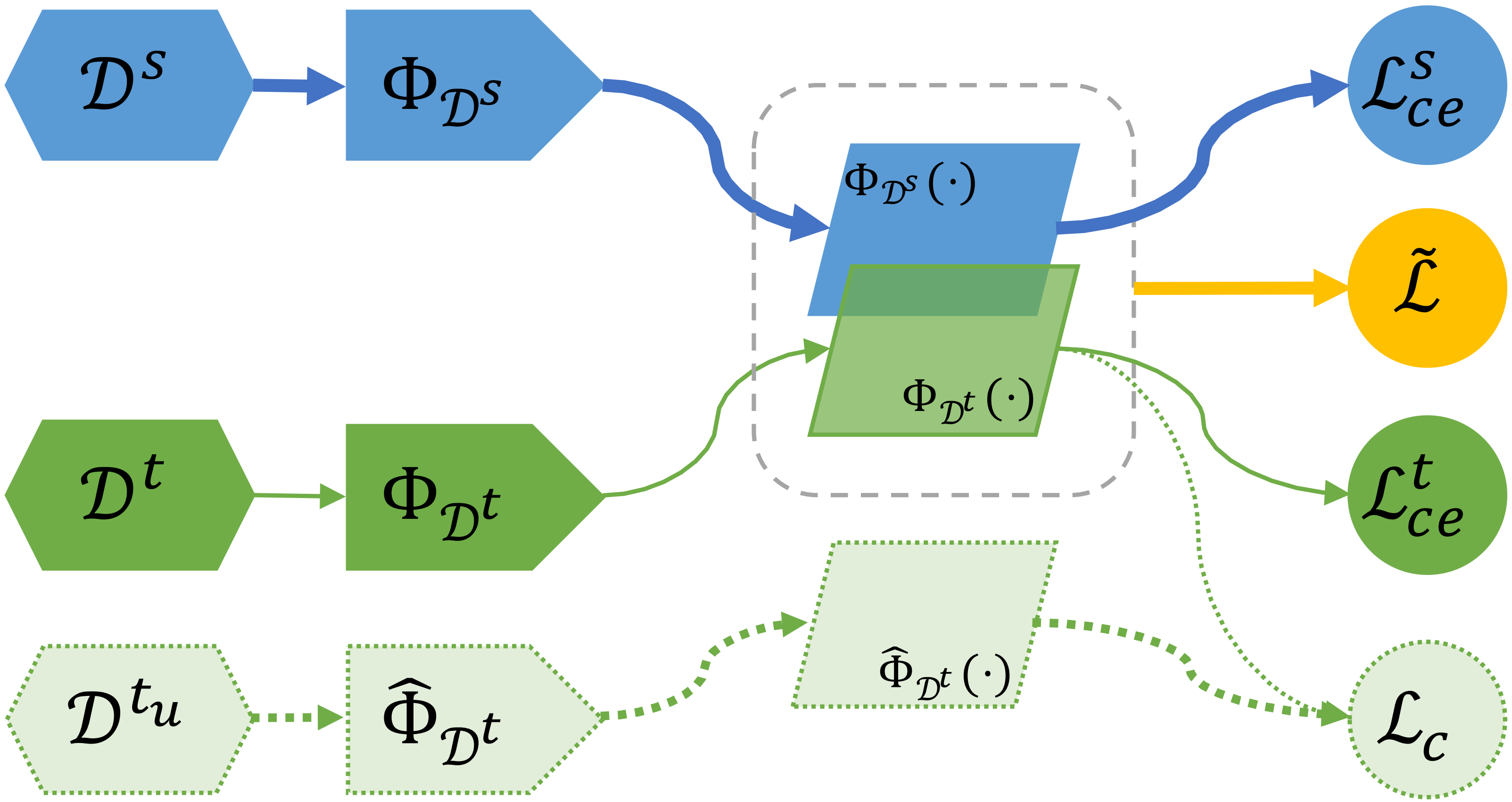
d-SNE: End-to-end learning

- Siamese network



d-SNE: Semi-supervised Extension

- Teacher-Student Network



Evaluation Datasets & Protocols

- Digits Datasets
  - MNIST, USPS, MNIST-M, SVHN
  - 1-10 samples/class labeled in the target domain
- Office31 Datasets
  - Small scale dataset with 31 classes
  - 3 samples in the target domain
- VisDA-C Datasets
  - Large scale synthetic-real dataset
  - 10 samples/class in the target domain

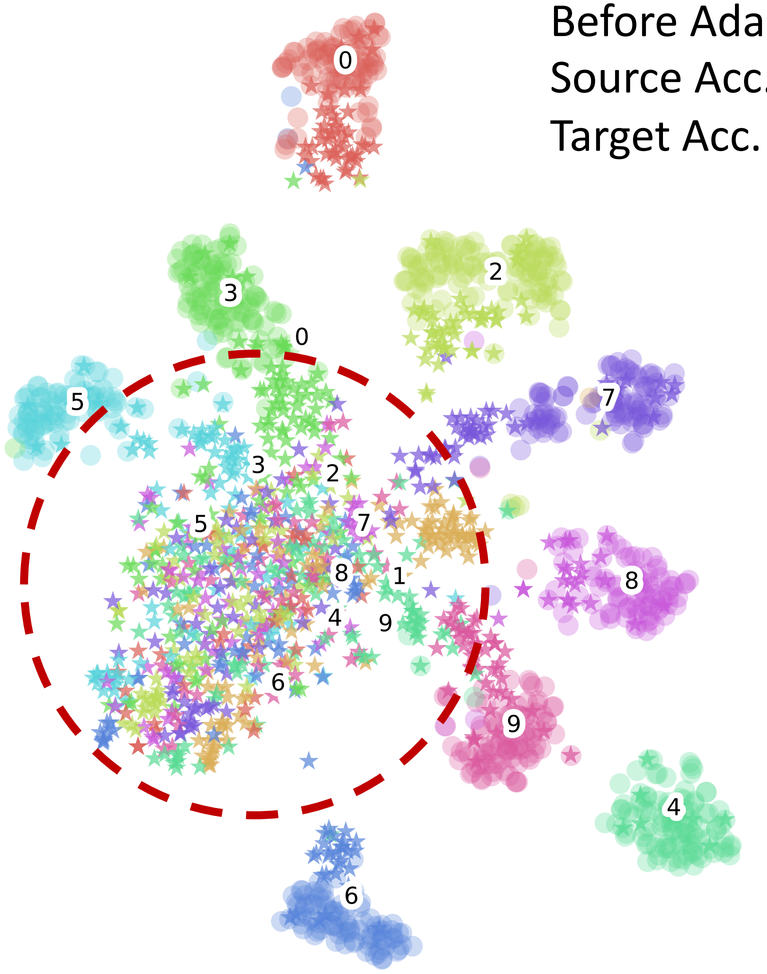
Results: Digits - Quantitative Results

Method	Setting	k	MNIST→ MNIST-M	MNIST→USPS	USPS→MNIST	MNIST→SVHN	SVHN→MNIST
DIRT-T[1]	*U*		98.90	-	-	54.50	**99.40**
SE[2]			-	98.23	99.54	71.40	92.00
SBADA-GAN[3]			**99.40**	95.04	97.60	61.08	76.14
G2A[4]			-	95.30	90.80	-	92.40
FADA[5]	*S*	7	-	94.40	91.50	47.00	87.20
CCSA[6]		10	78.29	97.22	95.71	37.63	94.57
d-SNE		7	84.62	97.53	97.52	53.19	95.68
		10	*87.80*	**99.00**	**98.49**	**61.73**	*96.45*
d-SNE	*SS*	10	94.12	-	-	77.63	97.60

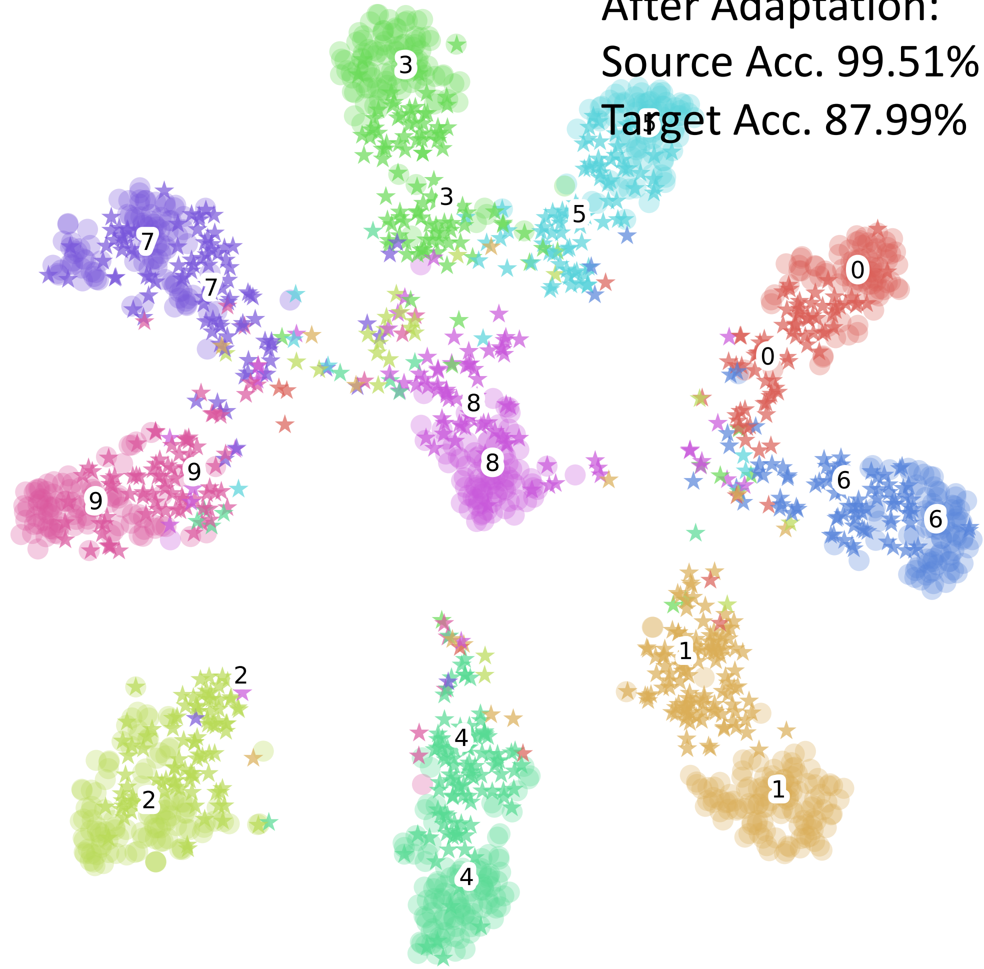
[1] R. Shu, H. H. Bui, H. Narui, and S. Ermon. A DIRT-T Approach to unsupervised Domain Adaption. In *Proc. ICLR*, 2018
[2] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for Visual Domain Adaptation. In *Proc. ICLR*, 2018
[3] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From source to target and back: symmetric bi-directional adaptive GAN. In *Proc. CVPR*, 2018.
[4] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate To Adapt: Aligning Domains using Generative Adversarial Networks. In *Proc. CVPR*, 2018
[5] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto. Few-Shot Adversarial Domain Adaptation. In *Proc. NIPS 2018*
[6] S.Motiian, M.Piccirilli, D.A. Adjeroh, and G.Doretto. Unified Deep Supervised Domain Adaptation and Generalization. In *Proc. IEEE ICCV*, 2017

Results: Digits - Qualitative Results

Before Adaptation:
Source Acc. 99.45%
Target Acc. 51.66%



After Adaptation:
Source Acc. 99.51%
Target Acc. 87.99%



MNIST -> MNIST-M

Conclusions

- Use of stochastic neighborhood embedding and large margin nearest neighbor to learn a domain agnostic latent-space for few-shot supervised learning
- Extension to semi-supervised settings pushing the states-of-the-art further.

166