FEW SHOT DOMAIN ADAPTATION FOR *in situ* MACROMOLECULE STRUCTURAL CLASSIFICATION IN CRYO-ELECTRON TOMOGRAMS

A PREPRINT

Liangyong Yu
Computational Biology Department
Carnegie Mellon University
Pittsburgh, 15213, USA

Ran Li
Department of Automation
Tsinghua University
Beijing, 100084, China

Xiangrui Zeng
Computational Biology Department
Carnegie Mellon University
Pittsburgh, 15213, USA

Hongyi Wang
Department of Electronic Engineering
Tsinghua University
Beijing, 100084, China

Jie Jin
Institute of Automation
Chinese Academy of Science
Beijing, 100190, China

Ge Yang
Institute of Automation
Chinese Academy of Science
Beijing, 100190, China

Rui Jiang
Department of Automation
Tsinghua University
Beijing, 100084, China

Min Xu
Computational Biology Department
Carnegie Mellon University
Pittsburgh, 15213, USA

May 26, 2021

ABSTRACT

Motivation: Cryo-Electron Tomography (cryo-ET) visualizes structure and spatial organization of macromolecules and their interactions with other subcellular components inside single cells in the close-to-native state at sub-molecular resolution. Such information is critical for the accurate understanding of cellular processes. However, subtomogram classification remains one of the major challenges for the systematic recognition and recovery of the macromolecule structures in cryo-ET because of imaging limits and data quantity. Recently, deep learning has significantly improved the throughput and accuracy of large-scale subtomogram classification. However often it is difficult to get enough high-quality annotated subtomogram data for supervised training due to the enormous expense of labeling. To tackle this problem, it is beneficial to utilize another already annotated dataset to assist the training process. However, due to the discrepancy of image intensity distribution between source domain and target domain, the model trained on subtomograms in source domain may perform poorly in predicting subtomogram classes in the target domain.
Results: In this paper, we adapt a few shot domain adaptation method for deep learning based cross-domain subtomogram classification. The essential idea of our method consists of two parts: 1) take full advantage of the distribution of plentiful unlabeled target domain data, and 2) exploit the correlation between the whole source domain dataset and few labeled target domain data. Experiments conducted on simulated and real datasets show that our method achieves significant improvement on cross domain subtomogram classification compared with baseline methods.
Availability: `https://github.com/xulabs/aitom`
Contact: mxu1@cs.cmu.edu

Keywords Few Shot Domain Adaptation · Macromolecule Classification · Cryo-electron Tomograms

1 Introduction

Plentiful complex biochemical processes and subcellular activities sustain the dynamic and complex cellular environment, in which a mass of intricate molecular ensembles participate. A comprehensive analysis of these ensembles *in situ*[1] inside single cells would play an essential role in understanding the molecular mechanisms of cells. Cryo-electron Tomography (cryo-ET), as a revolutionary imaging technique for structural biology, enables the *in situ* 3D visualization of structural organization information of all subcellular components in single cells in a close-to-native state at submolecular resolution. Thus cryo-ET can bring new molecular machinery insights of various cellular processes by systematically visualizing the structure and spatial organizations of all macromolecules and their spatial interactions with all other subcellular components in single cells at unprecedented resolution and coverage.

In particular, because of fractionated total electron dose over entire tilt series [1], we need to average multiple subtomograms[2] that contain identical structures in order to get high SNR subtomogram average representing higher resolution of the underlying structure [2]. However, the macromolecule structures in a cell are highly diverse. Therefore, it is necessary to first accurately classify these subtomograms into subsets of structurally identical macromolecules. This is performed by subtomogram classification. Systematic structural classification of macromolecules is a vital step for the systematic analysis of cellular macromolecular structures and functions [3] in many aspects including macromolecular structural recovery. However, such classification is very difficult, because of the structural complexity in cellular environment as well as the limit of data collection such as missing wedge effects [1]. Therefore, for successful automatic and systematic recognition and recovery of macromolecular structures captured by cryo-ET, it is imperative to have an efficient and accurate method for subtomogram classification.

With the technological breakthrough of cryo-ET and the development of image acquisition automation, collecting tomograms containing millions of macromolecules is no longer the obstacle for researchers, and methods based on deep-learning have been proposed to address the issue of high-throughput subtomogram classification thanks to the high-throughput processing capability of deep learning. Different architectures of Convolutional Neural Network (CNN) have been explored [4]. Despite the significant superiority in speed, accuracy, robustness and scalability compared to traditional methods, these supervised deep learning based subtomogram classification methods often suffer from the high demand of annotated data. Currently labeling is done by a combination of computational template search and manual inspection. However, in practise, template search is time-consuming and quality control through manual inspection is laborious. The complicated structure and distortion caused by noise make subtomogram images hard to distinguish by the naked eyes even by experts, which is a major obstacle for the manual quality insurance of the annotation.

An intuitive idea to tackle the problem of insufficient annotated data is to utilize a separated auxiliary dataset, which has abundant labeled samples, to assist subtomogram classification. Such auxiliary dataset is obtained from a separate imaging source or from simulation. Therefore the auxiliary dataset and our target dataset have the same structural classes but different image intensity distribution. The difference can be attributed to discrepant data acquisition conditions, such as different Contrast Transfer Function (CTF), signal-to-noise ratio (SNR), resolution, backgrounds, etc. The source domain is defined as the domain that the auxiliary dataset belongs to, and the target domain is defined as the domain that the evaluation dataset belongs to. In our case, we assume that we have plenty of labeled subtomograms in the source domain, but only few labeled samples in the target domain are accessible. This is due to the difficulty to annotate the data in target domain. For example, the real cryo-ET data in the target domain acquired from cryo-ET (real dataset) might be extremely time-consuming to annotate. On the other hand, we can generate simulated cryo-ET data in the source domain on the computer as the separated auxiliary dataset to assist us to improve the prediction accuracy of the real dataset in the target domain. Unfortunately, because of the image intensity distribution discrepancy between the source domain and the target domain, a deep learning model trained on the source domain perform poorly on the target domain due to dataset shift [5].

Domain Adaptation [6] is an effective way to solve this problem. This approach resolves the discrepancy of data distribution between source domain and target domain. One type of domain adaptation fine-tunes a trained neural network on source domain, which makes it perform well on both source domain and target domain. Another type of domain adaptation transforms target/source data in order to make it get close to the image intensity distribution of another domain [e.g. [7]]. Therefore, neural network doesn't need to distinguish two domains, because their image intensity distributions are similar by properly transforming the input data. Domain adaptation can also be categorized into unsupervised and supervised approaches: Unsupervised domain adaptation (UDA) requires large amount of data but doesn't need target labels [e.g. [8]], while supervised domain adaptation (SDA) requires target labels to be given [e.g. [9]]. Nowadays these two methods are the mainstream methods to reduce distribution discrepancy in source

[1] At their original locations.

[2] Subtomograms are subvolumes extracted from a tomogram, and each of them usually contains one macromolecule

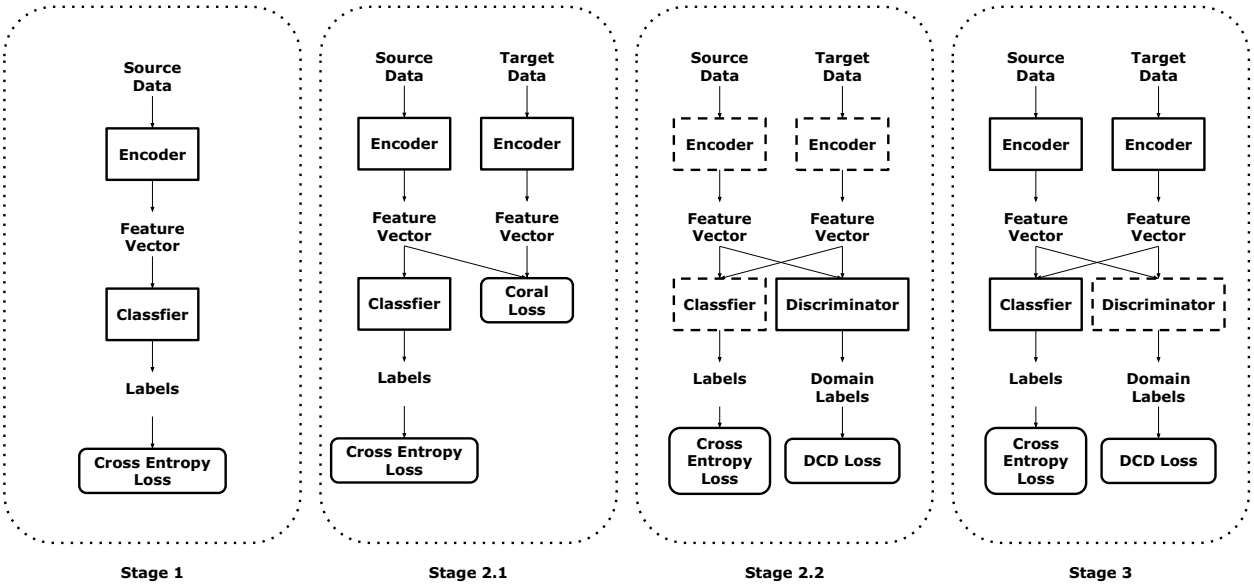


Figure 1: The flowchart of our method. The model whose edge is imaginary line represents that its parameters are fixed. In Stage 1, an encoder f_ϕ and a classifier g are initially trained using data in source domain (Section 2.1). In Stage 2, a discriminator $\mathcal{D}$ is trained to identify the domain of each subtomogram (Section 2.2). In Stage 3, labeled data in both domains are used to fine-tune the encoder f_ϕ with the assistance of discriminator $\mathcal{D}$ (Section 2.3).

domain and target domain. However, in our cryo-ET dataset, the methods based on UDA and SDA have obvious defects: (1) UDA can't utilize the information of labeled data in target domain, therefore intra-class relationship between source domain and target domain is neglected. (2) Often, due to annotation difficulty, there are too few labeled data in target domain that SDA can't reach satisfactory results.

Therefore, we propose a method for Few-Shot Domain Adaptation: Few-Shot Fine-Tuning domain adaptation(FSFT). Few-Shot means that each class contains only very few labels in the target domain [e.g. [10]]. Generally, for each class, we only use three to seven labels in the target domain. The flowchart of our method is presented on Figure 1. It contains three components: encoder f_ϕ, classifier g and discriminator $\mathcal{D}$. Encoder f_ϕ extracts every subtomogram into a feature vector[3]; classifier g transforms each feature vector into a one-hot label, which presents the class of each subtomogram; discriminator $\mathcal{D}$ identifies which domain the feature vectors belong to. The detailed training procedure is explained in the following section.

We have evaluated our method on both simulated and real datasets. Compared with popular baseline methods, our method achieves significantly higher classification accuracy. Additionally, related works and result analysis are presented in supplementary document.

Our main contributions are summarized as follows:

- We are the first to use few-shot domain adaptation for cross-domain subtomogram classification.
- We directly train the discriminator without adversarial training in the training procedure, comparing to FADA [10].
- We introduce a mechanism of partly-shared parameters of encoder f_ϕ between source domain and target domain. The layers whose parameters are shared by two domains are called domain-independent layers, and the other layers are called domain-related layers (Section 2.2.1).
- We combine domain discrimination for the output of independent layers and shared layers (Section 2.2.2).

[3]Feature vectors represent the output of encoder f_ϕ

2 Methods

In this section, we describe our model in details. Our training strategy contains three stages. Stage 1: an encoder f_ϕ and a classifier g are initially trained using data in source domain (Section 2.1). Stage 2: a discriminator $\mathcal{D}$ is trained to identify the domain of each subtomogram (Section 2.2). Stage 3: labeled data in both domains are used to fine-tune the encoder f_ϕ with the assistance of discriminator $\mathcal{D}$ (Section 2.3). Stage 2.1 is Unsupervised Domain Adaptation while Stage 2.2 and Stage 3 are Supervised Domain Adaptation.

Algorithm 1 Overall algorithm

Input:

Encoder in source domain: $f^0 \circ f^{\mathrm{s}}$
Encoder in target domain: $f^0 \circ f^{\mathrm{t}}$
Classifier g, discriminator $\mathcal{D}$.

Output:

Trained $f^0 \circ f^{\mathrm{s}}$, $f^0 \circ f^{\mathrm{t}}$, classifier g and discriminator $\mathcal{D}$.

1: Train $f^0 \circ f^{\mathrm{s}}$ and classifier g using source subtomograms (Stage 1)
2: Train $f^0 \circ f^{\mathrm{s}}$, $f^0 \circ f^{\mathrm{t}}$ and g using unlabeled target subtomograms (Stage 2.1) by algorithm 2
3: Train discriminator $\mathcal{D}$ using labeled target and source subtomograms (Stage 2.2)
4: Fine-tune $f^0 \circ f^{\mathrm{t}}$ and classifier g with the assistance of discriminator $\mathcal{D}$ and labeled target and source subtomograms (Stage 3)

2.1 Stage 1: Initialize encoder f_ϕ and classifier g

A series of subtomogram samples in source domain $X^{\mathrm{s}} = (x^{\mathrm{s}}, y^{\mathrm{s}})$ are provided in this section. We apply a 3D encoder f_ϕ, which maps each subtomogram into a feature vector in embedding space. We introduce an embedding function $f_\phi(\cdot)$ to represent the encoder f_ϕ. Because the parameters of encoder f_ϕ are partly shared between source domain and target domain, the embedding function can be composited by two parts: the domain-related function $f^{\mathrm{t}}(\cdot)$ or $f^{\mathrm{s}}(\cdot)$, and the domain-independent function $f^0(\cdot)$. That's to say, we apply $f^0 \circ f^{\mathrm{s}}(\cdot)$ for source domain and $f^0 \circ f^{\mathrm{t}}(\cdot)$ for target domain.

The application of the partly-shared encoder f_ϕ is based on the assumption that different domains have similar high level feature (including details), because the structure of subtomograms in the same class but from different domains are similar; but their low level features are different such as edges due to image intensity difference between domains. The front part is more for low level features and the back part for high level features. In other words, the front parts of encoder f^0 of f_ϕ extract the common structural features of both domains and remove the domain-related features such as image parameters and SNR. The back parts f^{s} and f^{t} further extract their common feature into embedding space. Second, a classifier g maps feature vectors into one-hot labels, which is represented by a prediction function $g(\cdot)$.

We update the encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ and classifier g by the following equation:

$$\theta \leftarrow \theta - \frac{1}{n}\beta\nabla_\theta[-\sum_{i=1}^{n} y_i^s \log(g \circ f^0 \circ f^{\mathrm{s}}(x_i^s))] \tag{1}$$

The loss function is:

$$L^{\mathrm{C}} = -\sum_{i=1}^{n} y_i^s \log(g \circ f^0 \circ f^{\mathrm{s}}(x_i^s)) \tag{2}$$

We set n as batch size. x_i^s represents the i-th subtomogram image, and y_i^s represents the i-th subtomogram label in each subtomogram sample batch.

2.2 Stage 2: Train the discriminator $\mathcal{D}$

After the first training step, the combination of encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ and classifier g have a perfect performance in classification of source domain because plentiful labeled source data $X^{\mathrm{s}} = (x^{\mathrm{s}}, y^{\mathrm{s}})$ is supplied. Unfortunately, due to the different experimental imaging parameters in two domains,

we can hardly reach satisfactory result in target domain. Thus, the essential part of our proposed method is utilizing unlabeled data and few labeled data in target domain in order to improve its performance in target domain. According to our experiments, even though the amount of labeled data in target domain is scarce, they are notably conductive to the improvement of classification accuracy in test stage.

Inspired by [10], we devise a discriminator $\mathcal{D}$ for Domain Adaptation in the following stages. [10] trains the discriminator $\mathcal{D}$ using adversarial training. The method is successful on the popular datasets such as MNIST, USPS and SVHN, because the loss function is easy to design. However, unlike the traditional 2D images, the spatial and structural information of our 3D subtomograms is very complicated and it is severely contaminated by noise. Therefore, it is difficult to train a desirable network using adversarial training because discriminator $\mathcal{D}$ and encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ are very hard to converge at the same time and their performance needs to be synchronized. Thus, as much as adversarial training is able to reach a satisfactory result in the traditional image datasets which have relatively high Signal-to-Noise-Ratio(SNR), when it comes to cryo-ET, the drawback of adversarial training would be exposed. Therefore, instead of training discriminator $\mathcal{D}$ and encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ alternately like adversarial training, in our model, the discriminator $\mathcal{D}$ is only trained once, and the parameters of the encoders f^{s}, f^{t}, and f^0 are not trained during the training of the discriminator $\mathcal{D}$.

In this section, we aim at training a discriminator $\mathcal{D}$ to distinguish the domain of each feature vector, which is described in detailed below.

2.2.1 Stage 2.1: Preprocessing of encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$

In this stage, we adjust the parameters of encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$. By doing this, the encoder $f^0 \circ f^{\mathrm{t}}$ is easier to extract the information of subtomograms of target domain. We use a discriminator $\mathcal{D}$ to assist encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ to confuse two domains; on the other hand, in order to make the discriminator $\mathcal{D}$ distinguish two domains well, our encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ must have the ability to confuse two domains. That's to say, at the beginning the distributions of feature vectors $T^{\mathrm{s}} := \{f^0 \circ f^{\mathrm{s}}(x_i^{\mathrm{s}})\}$ and $T^{\mathrm{t}} := \{f^0 \circ f^{\mathrm{t}}(x_i^{\mathrm{t}})\}$ in the two domains shouldn't have too much notable discrepancy. Otherwise, it would be so easy for the discriminator $\mathcal{D}$ to identify which domain every subtomogram belongs to, and its identification ability can hardly be improved. Therefore, the training of discriminator $\mathcal{D}$ relies on encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$, and the training of encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ relies on discriminator $\mathcal{D}$ too. Unfortunately, neither discriminator $\mathcal{D}$ and encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ are fully trained. The encoders f^{s}, f^{t}, and f^0 in two domains are all pre-trained on the source data $X^{\mathrm{s}} = (x^{\mathrm{s}}, y^{\mathrm{s}})$, so the parameter of the encoder $f^0 \circ f^{\mathrm{t}}$ in target domain is identical to the encoder $f^0 \circ f^{\mathrm{s}}$ in source domain. The model trained on data in source domain can hardly extract the feature in target domain very well, which becomes a major obstacle to train a discriminator $\mathcal{D}$.

In order to solve this problem, we use the following tactics. 1) Stage 2.1: Apply unsupervised domain adaptation (UDA) to encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$. 2) Stage 2.2: Train a discriminator $\mathcal{D}$ with the help of encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$. 3) Stage 3: Optimize encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ with the help of discriminator $\mathcal{D}$. The detailed algorithm of Stage 2.1 is discussed as follows.

The encoder $f^0 \circ f^{\mathrm{s}}$ trained by source data $X^{\mathrm{s}} = (x^{\mathrm{s}}, y^{\mathrm{s}})$ is pre-trained in the first stage (Equation 1), which we discussed in detail in the Section 2.1. We apply UDA for encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ in both domains before training a discriminator $\mathcal{D}$. UDA utilizes unlabeled data in target domain to enable our network the ability to initially confuse the data in two domains.

Specifically, for UDA, inspired by [11], deep correlation alignment (CORAL) is applied to encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ to reduce the domain distribution discrepancy between feature vectors T^{s} and T^{t} in source and target domains. We implement this method by appending CORAL loss to original classification loss. CORAL loss measures the distribution discrepancy between source domain and target domain in embedding space. We select a set of feature vectors D^{s} in source domain from T^{s} and a set of feature vectors D^{t} in target domain from T^{t}.

Specifically, CORAL loss is defined as:

$$L^{\text{CORAL}} = \frac{\|C^{\text{S}} - C^{\text{t}}\|_F^2}{4d^2}, \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm; C^{S} is the covariance matrix of D^{S} and C^{t} is the covariance matrix of D^{t}; and d is the dimension of the feature vectors D^{S} and D^{t}. C^{S} and C^{t} are calculated by the following equations:

$$C^{\text{S}} = \frac{1}{n^{\text{S}} - 1}\left[D^{\text{S}T} D^{\text{S}} - \frac{1}{n^{\text{S}}}(\mathbf{1}^T D^{\text{S}})^T(\mathbf{1}^T D^{\text{S}})\right], \tag{4}$$

$$C^{\text{t}} = \frac{1}{n^{\text{t}} - 1}\left[D^{\text{t}T} D^{\text{t}} - \frac{1}{n^{\text{t}}}(\mathbf{1}^T D^{\text{t}})^T(\mathbf{1}^T D^{\text{t}})\right], \tag{5}$$

where $\mathbf{1}$ denotes a column vector whose every element is 1; $n^{\text{S}} := |D^{\text{S}}|$ is number of feature vectors in D^{S}; and $n^{\text{t}} := |D^{\text{t}}|$ is the number of feature vectors in D^{t}.

The combined loss is defined as:

$$L^{\text{total}} = L^{\text{CORAL}} + L^{\text{C}}, \tag{6}$$

where L^{C} is the classification loss defined in 2.

The model architecture of UDA is shown in Figure 2. Generally, L^{CORAL} and L^{C} are opposite: trying to diminish L^{CORAL} must cause category confusion to encoders in source domain: $f^0 \circ f^{\text{S}}$ and encoders in target domain: $f^0 \circ f^{\text{t}}$ and classifier g and vice versa. We set α as 500 such that our model can reach a desirable result on target domain.

We simultaneously input data from two domains, and each batch contains data in both target domain and source domain. We acquire C^{S} and C^{t} by calculating the batch covariance [11] of subtomograms. In other words, D^{S} denotes the feature vectors in a subtomogram batch from source domain, and D^{t} denotes the feature vectors in a subtomogram batch from target domain.

Algorithm 2 Unsupervised Domain Adaptation Training

Input:
Subtomograms X^{S} in source domain.
Subtomograms X^{t} in target domain.
Output:
Trained encoders $f^0 \circ f^{\text{S}}$ and $f^0 \circ f^{\text{t}}$ and classifier g.
1: **for** m epochs **do**
2: **for** k steps **do**
3: Acquire feature vectors batch D^{S} and D^{t} from X^{S} and X^{t}
4: Calculate the covariance matrix C^{S} and C^{t} according to equations 4 and 5.
5: Update the parameters of encoders $f^0 \circ f^{\text{S}}$ and $f^0 \circ f^{\text{t}}$ and classifier g by minimizing 6.
6: **return** encoders $f^0 \circ f^{\text{S}}$ and $f^0 \circ f^{\text{t}}$ and classifier g.

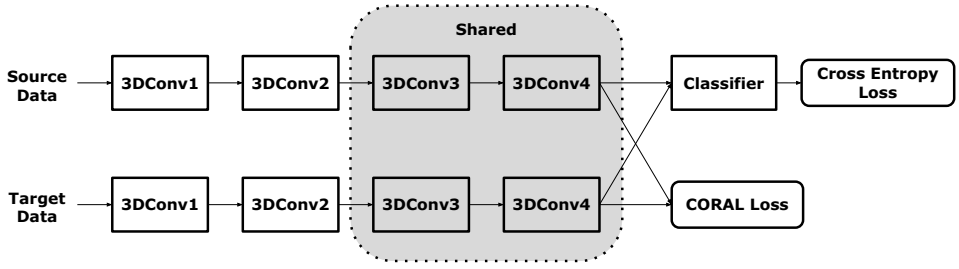


Figure 2: Our model architecture of Unsupervised Domain Adaptation. domain-related layers contain the first and second Convolution Block and domain-independent layers contain the third and last Convolution Block. That's to say, the parameters in the domain-independent layers are shared by data in source domain and target domain.

2.2.2 Stage 2.2: Update the parameters of discriminator $\mathcal{D}$

In this stage, we aim at training a discriminator $\mathcal{D}$ to differentiate two different domains. In order to fully utilize the label of target domain, inspired by [10], we design the discriminator $\mathcal{D}$ to identify whether two subtomograms are from the same domain and whether they belong to the same category. We consider the condition that the labeled data in target domain is scarce (For example, not more that 7 samples are labeled in each class). These labeled target samples are utilized in this step. We train a discriminator $\mathcal{D}$ in order to distinguish feature vectors T^{s} and T^{t} from source domain and target domain with the parameters of encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ and classifier g fixed. We combine all of the feature vectors in source domain T^{s} and labeled feature vectors T_l^{t}, and pair feature vectors in $W = T^{\mathrm{s}} \cup T_l^{\mathrm{t}}$. There are four kinds of pair combinations: 1) two paired feature vectors coming from the same domain and category, 2) from the same domain but different categories, 3) from different domains but the same category and 4) from different domains and categories. Therefore, we divide all pairs into 4 groups: G_1, G_2, G_3 and G_4 to correspond four kinds of pair combination above. The discriminator $\mathcal{D}$ learns to classify each pair into one of the four groups. In each training process, we obtain minibatch by selecting a certain number of feature vector pairs from the 4 groups. The parameters of discriminator $\mathcal{D}$ are updated by the following equation:

$$\theta \leftarrow \theta - \frac{1}{n}\beta\nabla_\theta\big[-\sum_{i=1}^{n} g_i \log(\mathcal{D}(t_i^1, t_i^2))\big] \tag{7}$$

where n denotes the size of minibatch. (t_i^1, t_i^2) represents the i-th feature vector pair from minibatch, and $t_i^1, t_i^2 \in W$. $g_i \in \{G_1, G_2, G_3, G_4\}$ represents the group ID of the i-th pair of minibatch. We use the function $\mathcal{D}(\cdot)$ to denote the discriminator $\mathcal{D}$.

The architecture of discriminator is showed in Figure 3. The discriminator $\mathcal{D}$ contains the 3D discriminator and 1D discriminator corresponding to our partly-shared encoder f_ϕ. The output of domain-independent layers (feature vectors $T^{\mathrm{s}}, T^{\mathrm{t}}$) and output of domain-related layers are both discriminated, because we assume that the input distribution of domain-independent layers has low correlation with domain variation. 3D discriminator distinguishes output domain of domain-related layers. 1D discriminator integrates the output of 3D discriminator and feature vectors $T^{\mathrm{s}}, T^{\mathrm{t}}$ then calculates the group ID of each pairs.

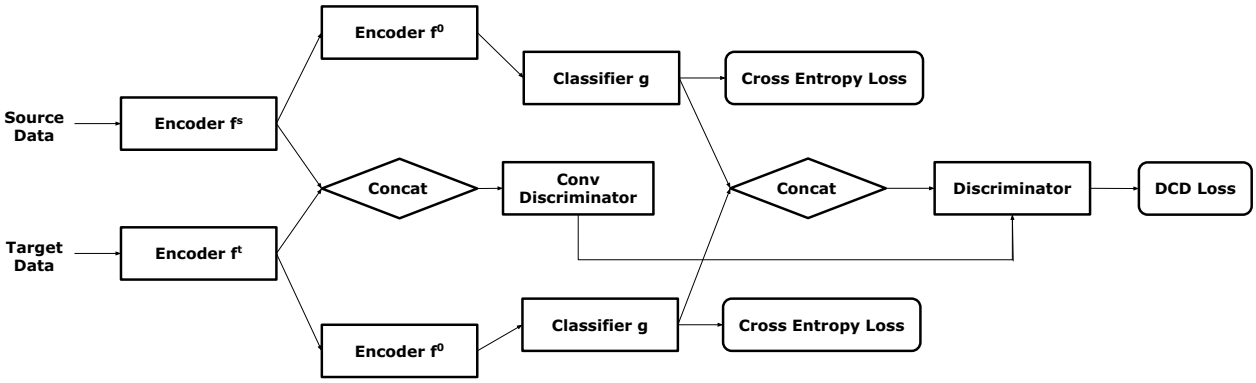


Figure 3: Our model architecture of Supervised Domain Adaptation (Stages 2.2 and 3).

2.3 Stage 3: Fine-tune the encoder f_ϕ

After training the discriminator $\mathcal{D}$, we fine-tune the encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ and classifier g again with the parameters of discriminator $\mathcal{D}$ frozen. We need to make discriminator $\mathcal{D}$ confused between G_1 and G_2, and also between G_3 and G_4 by updating the parameters of encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$, which is measured by the domain-class discriminator (DCD) loss [10]:

$$L^{\mathrm{DCD}} = -E\big[y_{G_1} \log\big(D(G_2)\big) - y_{G_3} \log\big(D(G_4)\big)\big], \tag{8}$$

where y_{G_i} represents the ID of G_i. Therefore the total loss can be denoted as:

$$L^{\mathrm{total}} = \gamma L^{\mathrm{DCD}} + L^{\mathrm{s}} + L^{\mathrm{t}}, \tag{9}$$

where L^{s} and L^{t} are the cross entropy loss functions to the classification of source domain and target domain.

Algorithm 3 Supervised Domain Adaptation Training

Input:
Cryo-ET data in source domain: X^{s}.
Labeled cryo-ET data in target domain: X^{t}.
Output:
Trained encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$, classifier g and discriminator $\mathcal{D}$.
1: Sample groups G_1,G_2,G_3 and G_4
2: **for** m epochs **do**
3: Update $\mathcal{D}$ with encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ and classifier g fixed by minimizing 7.
4: **for** m epochs **do**
5: Update encoders in source domain: $f^0 \circ f^{\mathrm{s}}$ and encoders in target domain: $f^0 \circ f^{\mathrm{t}}$ and classifier g with discriminator $\mathcal{D}$ fixed by minimizing 9.
6: **return** encoders $f^0 \circ f^{\mathrm{s}}$ and $f^0 \circ f^{\mathrm{t}}$, classifier g and discriminator $\mathcal{D}$

3 Results

3.1 Datasets

3.1.1 Simulated Subtomograms

The simulated subtomograms of 35^3 voxels are generated similar to [12]. Two simulated subtomogram dataset batches S_1, S_2 are provided to realize the domain adaptation process. S_1 is acquired through 2.2mm spherical aberration, -10μm defocus and 300kV voltage. S_2 is acquired through 2mm spherical aberration, -5μm defocus and 300kV voltage. Each dataset batch contains four datasets with different SNR levels (0.03, 0.05, 0.1, 0.5, 1000). Specifically, there are 43 macromolecular classes in each dataset. All of macromolecular classes are collected from PDB2VOL program [13], and each class in each dataset contains 100 subtomograms.

3.1.2 Real Subtomogram Datasets

We test our model on two real subtomogram datasets S_1 and S_2. S_1 is extracted from rat neuron tomograms [14], containing Membrane, Ribosome, TRiC, Single Capped Proteasome, Double Capped Proteasome and NULL class(the subtomogram with no macromolecule). Its SNR is 0.01, and the tilt angle ranges from -50° to $+70^\circ$.

S_2 is a single particle dataset from EMPIAR [15], containing Rabbit Muscle Aldolase, Glutamate Dehydrogenase, DNAB Helicase-helicase, T20S Proteasome, Apoferritin, Hemagglutinin and Insulin-bound Insulin Receptor. Its SNR is 0.5, with tilt angle range -60° to $+60^\circ$, size 28^3 voxels, and voxel spacing 0.94nm.

3.2 Classification Results

We conduct experiments respectively with finetune, FADA and our methods on simulated datasets and real datasets, and compare the results of these methods. Finally, we demonstrate the superiority of our method on the simulated and real datasets.

3.2.1 Results of Simulated Datasets

In this experiment, A_s is denoted as source domain and A_t is denoted as target domain. For facilitating computation, we randomly sample 100 subtomograms from each class. Table 1 presents the prediction accuracy in these methods.

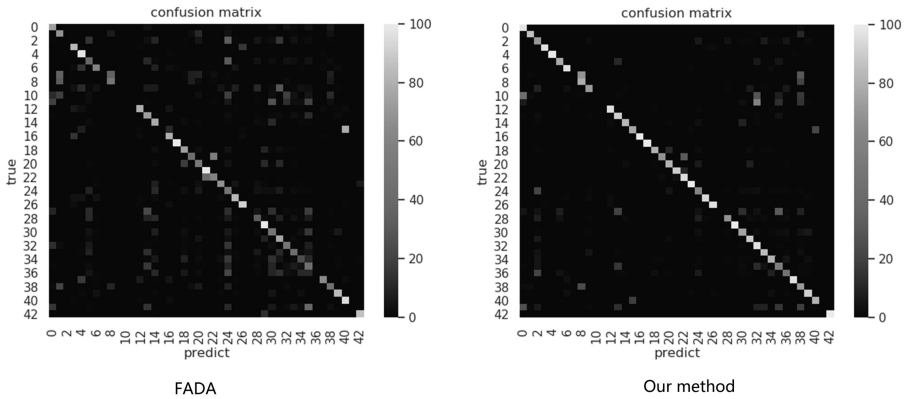
3.2.2 Results of Cross-domain Prediction of Real Subtomograms

The real datasets are acquired in the very complicated environment, causing the heterogeneity of subtomograms and very low SNR comparing to simulated dataset. This characteristic of experimantal datasets poses a challenge to the macromolecule classification.

Table 1: The classification accuracy of the dataset from target domain. The result in each cell represents the accuracy of CORAL [11], Sliced Wasserstein Distance[16], finetune, FADA and our method from top to bottom. The highest accuracy in each cell is highlighted. It shows that the prediction accuracy of our method surpasses the baseline methods in most of the cases.

	Target Domain					
Source Domain	SNR	1000	0.5	0.1	0.05	0.03
	1000	0.470	0.066	0.049	0.034	0.024
		0.442	0.148	0.095	0.078	0.063
		0.513	0.211	0.083	0.062	0.050
		0.664	0.404	0.185	0.161	0.146
		0.761	**0.518**	**0.253**	**0.196**	**0.177**
	0.5	0.189	0.369	0.219	0.125	0.104
		0.321	0.374	0.244	0.144	0.150
		0.387	0.416	0.204	0.137	0.111
		0.660	**0.577**	0.328	**0.255**	0.171
		0.601	0.532	**0.332**	0.254	**0.203**
	0.1	0.107	0.24	0.237	0.188	0.166
		0.125	0.250	0.230	0.166	0.143
		0.280	0.285	0.263	0.184	0.147
		0.415	0.436	0.297	0.231	0.170
		0.513	**0.456**	**0.332**	**0.257**	**0.218**
	0.05	0.034	0.147	0.197	0.145	0.13
		0.057	0.170	0.126	0.152	0.150
		0.184	0.238	0.203	0.191	0.137
		0.280	0.292	0.231	0.205	0.176
		0.439	**0.374**	**0.292**	**0.256**	**0.235**
	0.03	0.045	0.117	0.115	0.122	0.127
		0.061	0.123	0.098	0.088	0.106
		0.089	0.190	0.166	0.166	0.148
		0.276	0.229	0.202	0.194	0.177
		0.218	**0.243**	**0.211**	**0.200**	**0.200**

Five simulated datasets in A_s and A_t with different SNR(1000, 0.5, 0.1, 0.05, 0.03) are utilized. Each of the simulated dataset acts as source domain, and their classes are the same as target domain. and two real subtomogram datasets are acted as the target domain. Table 2 shows the classification results on all of the methods. The result in each cell represents the prediction accuracy in real dataset, and the confusion matrices have been showed in 4. Additionally, 3 and 7 labeled samples are selected in target domain for supervised training in FADA and our method.



FADA Our method

Figure 4: The confusion matrix in our method and baseline method. In view that FADA is far more better than other baselines, we compare the confusion matrix in our method to those in FADA. The left is the confusion matrix of FADA, the right is the confusion matrix of our method.

Table 2: The classification accuracy on the real dataset. The row of the cell denotes which method is utilized and the column of the cell denotes the SNR of source domain. The first row is FADA and the second row is our method. It is obvious that the superiority of our method increases as the SNR becomes lower comparing to FADA.

			Target Domain				
			1000	0.5	0.1	0.05	0.03
Source Domain	3 shot	S_1	**0.801**	0.626	0.453	0.535	0.538
			0.732	**0.720**	**0.608**	**0.606**	**0.586**
		S_2	0.705	0.731	0.793	0.748	0.655
			0.788	**0.733**	**0.891**	**0.849**	**0.774**
	7 shot	S_1	**0.842**	0.664	0.760	0.679	0.690
			0.774	**0.805**	**0.791**	**0.719**	**0.701**
		S_2	**0.959**	0.952	0.947	0.953	0.796
			0.833	**0.969**	**0.971**	**0.954**	**0.958**

4 Conclusion

Recently, Cryo-Electron Tomography emerges as a powerful tool for systematic *in situ* visualization of the structural and spatial information of macromolecules in single cells. However, due to high structural complexity and the imaging limits, the classification of subtomograms is very difficult. Supervised deep learning has become the most powerful method for large scale subtomogram classification. However, the construction of high quality training data is laborious. In such case it is beneficial to utilize another already annotated dataset to train neural network model. However, there often exists a systematic image intensity distribution difference between the annotated dataset and target dataset. In such case the model trained on another annotated dataset may have a poor performance in target domain. In this paper, we propose a Few-shot Domain Adaptation method to for cross-domain subtomogram classification. Our method combines Unsupervised Domain Adaptation and Supervised Domain Adaptation: we first train a discriminator $\mathcal{D}$ to identify the domain of each subtomogram, and we utilize the discriminator $\mathcal{D}$ to assist us the process of SDA. To the best of our knowledge, this is the first work to apply semi-supervised Domain Adaptation on subtomogram classification. We conduct experiments on simulated dataset and real dataset, and the prediction accuracy of our methods surpasses the baseline methods. Therefore, our method can be effectively applied to the subtomogram classification from a new domain with only a few labeled samples supplied. Our work represents an important step toward fully utilizing deep learning for subtomogram classification, which is critical for the large-scale and systematic *in situ* recognition and recovery of macromolecular structures in single cells captured by cryo-ET.

Funding

This work was supported in part by U.S. National Institutes of Health (NIH) grant P41GM103712 and R01GM134020, U.S. National Science Foundation (NSF) grant DBI-1949629 and IIS-2007595, and Mark Foundation for Cancer Research grant 19-044-ASP. XZ was supported by a fellowship from Carnegie Mellon University's Center for Machine Learning and Health.

A Few Shot Domain Adaptation for *in situ* Macromolecule Structural Classification in Cryo-electron Tomograms – Supplementary Document

A.1 Related work

Current analysis of cryo-ET includes template matching [17]. First we create the templates for every class; and for every subtomogram, we calculate the matching score between the template and itself. The method is straight-forward and easy to realize, but the computation complexity is unbearable, especially on the data which has countless classes and more dimensions compared to traditional images. What's more, because of the intense disruption of noise, the error rate of this method is very high.

Another method utilizes unsupervised subtomogram classification (e.g. [18]). Now there are a set of subtomograms which correspond to k classes. First we initialize k class centers, each of which represents the average of subtomograms in each class, and therefore all of subtomograms can be classified by computing the distance to k class centers. Second, after labeling all of subtomograms, we redirect k class centers by calculating average of labeled subtomograms in each class. By computing the two above steps iteratively, we can approximately obtain the label of every subtomogram. This

method doesn't require the label of any subtomogram, reducing the workload of labeling our data. Nevertheless, even if some tactics eliminating noises has been elaborated in this paper, the noises are still remaining a severe problems in our subtomogram classification, which extremely affect the performance in this method adversely.

There is another straightforward resolution that we can implement transfer learning into subtomogram classification. We generate simulated dataset in the computer as source domain and set real dataset as target domain. With the development of Neural Network, many Deep Learning models use this tactic to solve this problem. [19] proposed an unsupervised classification method with transfer learning. First we train a CNN model by simulated dataset. In the second stage, we remove the last layer from the original model and then extract the feature vector of the real dataset. In the end, we apply k-means clustering to the feature vector of the real dataset.

Recently [20] applies Unsupervised Domain Adaptation to subtomogram classification, in order to resolve the situation, in which source domain(train dataset) and target domain(test dataset) have different image intensity distribution. Even though it reaches a desirable performance in target domain, there still remains limitations because no label in target domain is utilized; what's more, adversarial training is used in this paper. In Section 2.2, we have discussed in detail that adversarial training is hard to be convergent when using Cryo-ET. Comparing to this paper, we success to utilize the label information in target domain and further improve its performance.

There are two mainstream ways for current Unsupervised Domain Adaptation methods to decrease image intensity distribution. First, the training dataset is used to optimize our model and later the parameter of this model is fine-tuned by test dataset. Even if the label information of test dataset isn't available, some of its global features, such as mean and covariance, can still be calculated. This kind of information is crucial for us to fine-tune the parameters of our model. For instance, [11] and [21] use this way as domain adaptation. The second way is transforming the data in target domain, making its distribution more similar to the data in source domain. Compared to the first way, parameters of the model would not be fine-tuned. For instance, [7] use this way as domain adaptation. In this paper, whitening and re-coloring, which utilizes the covariance of data in source domain and data in target domain, are applied to data in source domain. The source data being transformed are used to train the classification model. Because transformed source data has the similar distribution with target data, the model can reach a desirable result on target domain.

Compared to Unsupervised Domain Adaptation, Few-shot Domain Adaptation utilizes the whole data in source domain and very few labeled data in target domain. The core idea is very similar to Few Shot Learning[22]: we require our model to learn the features in very few images. However, the two fields still have very significant difference. Few shot learning needs to learn the image features whose labels aren't presented in training dataset, while few shot domain adaptation needs to learn the image features whose domain are different from training dataset.

A.2 Time complexity

We test the time complexity of FSFT and other Deep Learning methods, which is presented in Table 3.

Table 3: This table lists cost time of five Deep Learning method. FSFT costs less time than SWD and Fine-tune while costs more time than FADA and CORAL.

Model	Time Cost(s)
CORAL	323.66
SWD	1797.37
Fine-tune	1002.08
FADA	554.80
FSFT	921.36

From the table, CORAL, as the simplest method, costs the least time. Compared to FADA, FSFT add Deep CORAL as one of crucial stage, and its model is more complex. These changes introduce more computation, in order to have a better performance in subtomogram classification.

A.3 Result Analysis

We conduct some experiments to analyze to verify the effectiveness of FSFT. We generate 23 classes for the simulated datasets S_1 and S_2 which are mentioned in Section 3.1.1. In this section, we want to verify the effectiveness of each stage and each contribution we proposed. All the experiments in this section are conducted on these datasets.

Firstly, in the task of subtomogram classification, we split the whole training procedure into 3 stages which are tightly linked. We verify that each stage plays an important role in improving the classification precision. Table 4 presents

the improvement of each stage in FSFT. Unsupervised Domain Adaptation is used in Stage 2 and Supervised Domain Adaptation is used in Stage 3. The combination of them enable encoders $f^0 \circ f^t$ to adapt the target domain.

Table 4: We calculate the classification accuracy in each stage. This table shows that in Stage 2.1, Deep CORAL method improves the accuracy from 37.2% to 70.9%; In Stage 2.2, we only update the parameter of discriminator, so the prediction accuracy is the same as Stage 2.1. In Stage 3, Fine-tune the encoder f_ϕ improves the accuracy from 70.9% to 95.3%

Stage	Accuracy
Stage 1	37.2%
Stage 2.1	70.9%
Stage 2.2	70.9%
Stage 3	**95.3%**

Secondly, the contributions mentioned in Section 1 are effective in improving the performance of FSFT. In order to verify their effectiveness, we remove each contribution in FSFT and test its performance in target domain. In table 5, FSFT method we proposed realizes the best performance, while others can't reach the optimal accuracy compared to FSFT.

Table 5: Accuracy of ablation study. Row 1 corresponds to FSFT we proposed; Row 2 corresponds to FSFT without Stage 2.1; Row 3 corresponds to FSFT which use GAN to train discriminator and encoder; Row 4 corresponds to FSFT which only use 1D discriminator.

Model	Accuracy
FSFT	**95.3%**
FSFT without CORAL	79.1%
FSFT with GAN	94.2%
FSFT without 3D Discriminator	95.1%

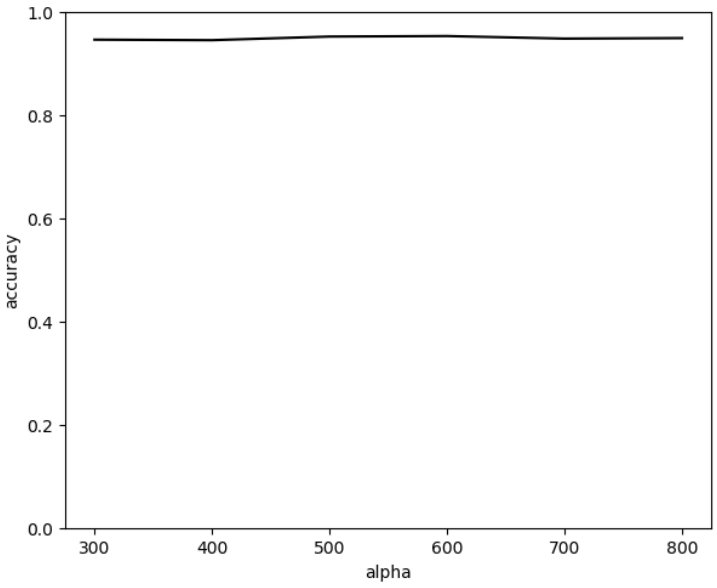


Figure 5: This picture shows how α affect the prediction accuracy.

A.4 Hyper-parameter Adjustment

In this section, we discuss how to choose the hyper-parameter, for example, α in equation 6. In figure 5, the best result is 0.954%. Accuracy, as α changes, the accuracy nearly stays constant. The value of α will have little effect on the performance of our model.

References

[1] Alberto Bartesaghi, P Sprechmann, J Liu, G Randall, G Sapiro, and Sriram Subramaniam. Classification and 3d averaging with missing wedge correction in biological electron tomography. *Journal of structural biology*, 162(3):436–450, 2008.

[2] John AG Briggs. Structural biology in situ—the potential of subtomogram averaging. *Current opinion in structural biology*, 23(2):261–267, 2013.

[3] Rossitza N Irobalieva, Bruno Martins, and Ohad Medalia. Cellular structural biology as revealed by cryo-electron tomography. *J Cell Sci*, 129(3):469–476, 2016.

[4] Chengqian Che, Ruogu Lin, Xiangrui Zeng, Karim Elmaaroufi, John Galeotti, and Min Xu. Improved deep learning-based macromolecules structure classification from electron cryo-tomograms. *Machine Vision and Applications*, 29(8):1227–1236, 2018.

[5] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[6] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics, 2006.

[7] Md Jahangir Alam, Gautam Bhattacharya, and Patrick Kenny. Speaker verification in mismatched conditions with frustratingly easy domain adaptation. In *Odyssey*, pages 176–180, 2018.

[8] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[9] Daniel Garcia-Romero and Alan McCree. Supervised domain adaptation for i-vector based speaker recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4047–4051. IEEE, 2014.

[10] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 6670–6680, 2017.

[11] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer, 2016.

[12] Min Xu, Xiaoqi Chai, Hariank Muthakana, Xiaodan Liang, Ge Yang, Tzviya Zeev-Ben-Mordehai, and Eric P Xing. Deep learning-based subdivision approach for large scale macromolecules structure recovery from electron cryo tomograms. *Bioinformatics*, 33(14):i13–i22, 2017.

[13] Enrique E Abola, Frances C Bernstein, and Thomas F Koetzle. The protein data bank. In *Neutrons in Biology*, pages 441–441. Springer, 1984.

[14] Qiang Guo, Carina Lehmer, Antonio Martínez-Sánchez, Till Rudack, Florian Beck, Hannelore Hartmann, Manuela Pérez-Berlanga, Frédéric Frottin, Mark S Hipp, F Ulrich Hartl, et al. In situ structure of neuronal c9orf72 poly-ga aggregates reveals proteasome recruitment. *Cell*, 172(4):696–705, 2018.

[15] Alex J Noble, Hui Wei, Venkata P Dandey, Zhening Zhang, Yong Zi Tan, Clinton S Potter, and Bridget Carragher. Reducing effects of particle adsorption to the air–water interface in cryo-em. *Nature methods*, 15(10):793, 2018.

[16] Alexander J Gabourie, Mohammad Rostami, Soheil Kolouri, and Kyungnam Kim. System and method for unsupervised domain adaptation via sliced-wasserstein distance, April 23 2020. US Patent App. 16/719,668.

[17] Martin Beck, Johan A Malmström, Vinzenz Lange, Alexander Schmidt, Eric W Deutsch, and Ruedi Aebersold. Visual proteomics of the human pathogen leptospira interrogans. *Nature methods*, 6(11):817, 2009.

[18] Min Xu, Martin Beck, and Frank Alber. High-throughput subtomogram alignment and classification by fourier space constrained fast volumetric matching. *Journal of structural biology*, 178(2):152–164, 2012.

[19] Emmanuel Moebel. *New strategies for the identification and enumeration of macromolecules in 3D images of cryo electron tomography*. Theses, Université de Rennes 1, February 2019.

[20] Ruogu Lin, Xiangrui Zeng, Kris Kitani, and Min Xu. Adversarial domain adaptation for cross data source macromolecule in situ structural classification in cellular electron cryo-tomograms. *Bioinformatics*, 35(14):i260–i268, 2019.

[21] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[22] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.