A DOMAIN-ADAPTIVE TWO-STREAM U-NET FOR ELECTRON MICROSCOPY IMAGE SEGMENTATION

Róger Bermúdez-Chacón, Pablo Márquez-Neila, Mathieu Salzmann, Pascal Fua

Computer Vision Laboratory, EPFL

ABSTRACT

Deep networks such as the U-Net are outstanding at segmenting biomedical images when enough training data is available, but only then. Here we introduce a Domain Adaptation approach that relies on two coupled U-Nets that either regularize or share corresponding weights between the two streams, along with a differentiable loss function that approximates the Jaccard index, to leverage training data from one domain in which it is plentiful, to adapt the network weights in another where it is scarce.

We showcase our approach for the purpose of segmenting mitochondria and synapses from electron microscopy image stacks of mouse brain, when we have enough training data for one brain region but only very little for another. In such cases, we outperform state-of-the-art Domain Adaptation methods.

***Index Terms*—** Image segmentation, Domain Adaptation, Electron Microscopy, Machine Learning

1. INTRODUCTION

Deep Learning techniques became central to image segmentation, and are now used for tissue characterization, digital pathology, and high-content screening. The U-Net [1] is one of the most successful deep architectures, and has been used in many diverse settings. It is an encoding-decoding network that uses intermediate encodings at different resolutions as additional features while decoding. With enough training data available, such design yields exceptional segmentation results in many biomedical problems [2, 3, 4], and ranks above humans [**?**] in academic challenges such as the SNEMI3D. We have also experimentally confirmed that the U-Net is well suited for segmenting mitochondria and synapses in electron microscopy (EM) images of brain tissue.

However, like most machine learning techniques, the U-Net is sensitive to domain shifts, which occur when the statistical properties of the test data differ from that of the training data. In our specific application, this might occur when a U-Net is trained using organelles from one part of the brain and then tested in another one where the appearance differs, as shown in Fig. 1, or due to inconsistent tissue staining processes, among other reasons. This can be addressed by annotating entirely new training data, which is slow, costly, and error-prone. A more effective approach is to perform *Domain Adaptation*, which aims at leveraging existing annotations along with a very small amount of labeled data from the new acquisition, to adapt the model to the new statistics.

Let the *source domain* be the one in which we have enough annotated data and let the *target domain* be another to which we want to adapt using as few new annotations as possible. In the context of Deep Learning, recent Domain Adaptation works aim to learn representations that are invariant to the domain shift, for example by minimizing the Maximum Mean Discrepancy (MMD) between source and target distributions [6, 7], by training adversarial domain classifiers [8, 9] or by aligning the second- or higher-order statistics of the domains [10, 11]. These methods typically either use one common network with identical weights for both domains, or fine-tune the source network using the few annotated target samples. By contrast, in [**?**], it was shown that it may be beneficial to allow source and target network weights to differ while regularizing them to prevent them from drifting too far apart. Although promising, all of the above methods deal only with classification or regression problems, using relatively simple networks.

In this work, we tackle the more complex image segmentation case, using the more sophisticated U-Net architecture. We introduce a simple and effective approach to adapting the U-Net weights from a source to a target domain, which relies on mirroring the U-Net structure into two streams, one for the source domain and one for the target, and allowing some of their weights to differ, while the others are shared. With this, we find a compromise between preserving what can be learned from the source domain using enough training data and adapting the weights to the potentially different statistics of the target domain. In addition, we introduce a novel loss function based on a differentiable version of the Jaccard index, which is more consistent with the standard evaluation criterion in segmentation. Our experiments demonstrate that our approach surpasses state-of-the-art Domain Adaptation techniques on the task of segmenting organelles in EM images.

2. METHOD

Let us consider a source domain s, for which we have enough annotated data to properly train a U-Net, and a target do-

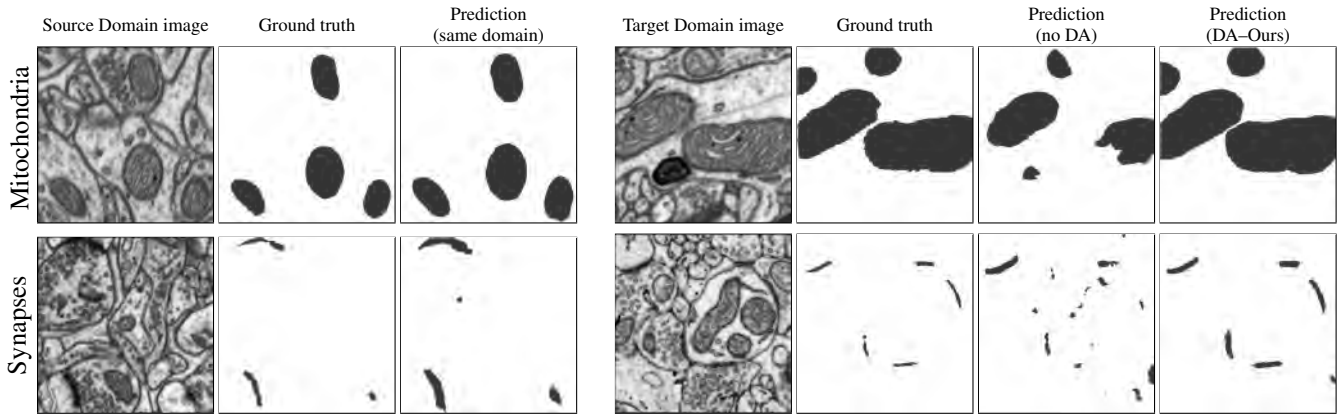

Fig. 1. **Segmenting EM images.** (Left) A U-Net trained to segment mitochondria (top) or synapses (bottom) in a specific brain region does it well. (Right) The same U-Net does poorly elsewhere in the brain *before* Domain Adaptation but much better *after* Domain Adaptation.

main t, in which we have far less. Let $\mathcal{X}^s = \{\mathbf{X}_i^s\}_{i=1}^{N^s}$ be the training images for the source domain with corresponding annotations $\mathcal{Y}^s = \{\mathbf{Y}_i^s\}$. Furthermore, let $\mathcal{X}^t = \mathcal{X}^{tl} \cup \mathcal{X}^{tu}$ be the target images, separated into labeled $\mathcal{X}^{tl} = \{\mathbf{X}_i^{tl}\}$ images, with labels $\mathcal{Y}^t = \{\mathbf{Y}_i^t\}$, and unlabeled images $\mathcal{X}^{tu} = \{\mathbf{X}_i^{tu}\}$. We formulate the segmentation problem as that of assigning pixels a value of either one or zero, depending on whether they belong to a structure of interest or not. To this end, we develop the coupled U-Net architecture described below.

A standard U-Net comprises a series of convolutional layers with an either increasing or decreasing number of channels, interleaved with pooling operations for the encoding layers and up-convolutions for the decoding layers. As shown in Fig. 2, we use two U-Nets, one for each domain. The behavior of these two streams is defined by two sets of weights $\Theta^s = \{\theta_j^s\}_{j=1}^L$ and $\Theta^t = \{\theta_j^t\}_{j=1}^L$, where L is the number of layers, with some layers potentially being shared and others not, as proposed in [**?**].

We train the network by minimizing the loss function

$$
\begin{aligned}
L(\Theta^s, \Theta^t | \mathcal{X}^s, \mathcal{X}^t, \mathcal{Y}^s, \mathcal{Y}^t) \;=\; & \sum_{i=1}^{N^s} c(f^s(\mathbf{X}_i^s|\Theta^s), \mathbf{Y}_i^s) + \\
& \sum_{i=1}^{|\mathcal{X}^{tl}|} c(f^t(\mathbf{X}_i^{tl}|\Theta^t), \mathbf{Y}_i^t) + \\
& \lambda_w \frac{1}{|\Omega|} \sum_{j \in \Omega} r_w(\theta_j^s, \theta_j^t) + \\
& \lambda_o r_o(\mathcal{X}^s, \Theta^s, \mathcal{X}^t, \Theta^t) \;, \qquad (1)
\end{aligned}
$$

where $f^{\cdot}(X^{\cdot}|\Theta^{\cdot})$ denotes the function encoded by one U-Net stream, i.e., the network predictions, $c(\cdot,\cdot)$ is a classification loss that is tailored for segmentation purposes, as discussed in Section 2.3, and the other terms, described in detail in Sections 2.1 and 2.2 below, are regularizers and their corresponding weights. We optimize this loss with respect to Θ^s, Θ^t jointly using the Adam optimizer [12] with 10^{-4} learning rate.

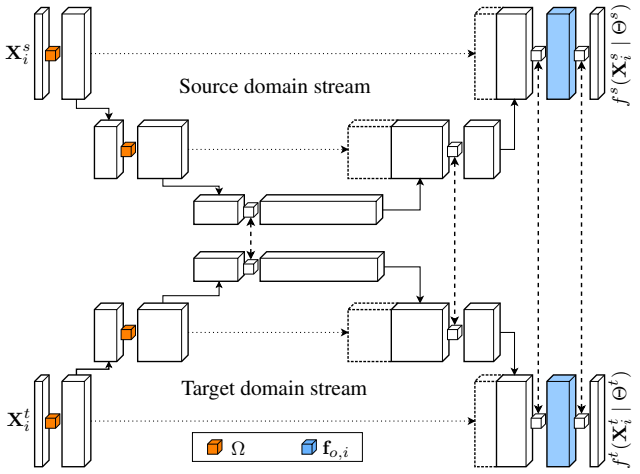

Fig. 2. **Simplified representation of our two-stream U-Net.** (See [1] for the detailed U-Net architecture.) Top stream is trained on the source domain and bottom stream on the target domain. Some of the weights are shared (dashed arrows) while the others can differ but are constrained not to be excessively dissimilar. The final feature maps, shown in blue, are also constrained to exhibit similar statistics.

2.1. Parameter Regularization and Sharing

In practice, for a particular U-Net layer, we either share all or no weights across the two streams. The first regularizer then penalizes weight differences between corresponding layers that do not share their weights. Specifically, let Ω be the subset of weights from layers that the two streams do *not* share. For each weight, we take the penalty term r_w of Eq. 1 as

$$
r_w(\theta_j^s, \theta_j^t) = \|a_j \theta_j^s + b_j - \theta_j^t\|_2^2, \; j \in \Omega \;, \qquad (2)
$$

where a_j and b_j are scalars that are learnt jointly with the network weights. We have observed the best results by sharing the weights of the up-coding layers—the right part of each stream in Fig. 2—and *not* sharing those of the others.

2.2. Feature Regularization

Following common practice, we also apply a regularizer that aims to compare the distribution of the final feature maps $\mathbf{f}^s_{o,i}(\mathbf{X}^s_i, \Theta^s)$ and $\mathbf{f}^t_{o,i}(\mathbf{X}^t_i, \Theta^t)$ produced by the two streams, that is, the representation preceding the classifier layer. We evaluated two such standard regularization terms.

The first one corresponds to the MMD between the source and target features, and can be expressed as

$$r_o^{\text{MMD}}(\mathbf{f}^s_o, \mathbf{f}^t_o) = \left\| \frac{1}{N^s M} \sum_{i=1}^{N^s} \sum_{k=1}^{M} \phi(\mathbf{f}^s_{o,i,k}(\mathbf{X}^s_i, \Theta^s)) - \frac{1}{N^t M} \sum_{i=1}^{N^t} \sum_{k=1}^{M} \phi(\mathbf{f}^t_{o,i,k}(\mathbf{X}^t_i, \Theta^s)) \right\|^2 , \quad (3)$$

where k sums over the spatial positions on the output feature maps (since we have a fully-convolutional architecture), and $\phi(\cdot)$ is a mapping to a reproducing kernel Hilbert space. The MMD can in fact be expressed in terms of a kernel function, and in practice we use the RBF kernel.

The second regularizer we evaluated is based on the correlation alignment method of [10]. Let C^s_o be the correlation matrix of the final source feature map $\mathbf{f}^s_{o,i}(\mathbf{X}^s_i, \Theta^s)$, and C^t_o be the equivalent matrix for the target domain. We then write

$$r_o^{\text{corr}}(\mathbf{f}^s_o, \mathbf{f}^t_o) = \|C^s_o - C^t_o\|^2_F . \quad (4)$$

In our experiments, we have found such regularizer, which implicitly encourages the segmentations in both domains to correlate in a similar fashion, to be more effective than r_o^{MMD}.

2.3. Segmentation Loss Function

For binary classification, the most commonly used loss function is the binary cross-entropy [13]. However, the most popular *metric* to assess segmentation quality is the Jaccard index, or Intersection over Union (*IoU*), written as

$$J(\mathbf{y}, \hat{\mathbf{y}}) = \frac{|\mathbf{y}_+ \cap \hat{\mathbf{y}}_+|}{|\mathbf{y}_+ \cup \hat{\mathbf{y}}_+|} = \frac{|\mathbf{y}_+ \cap \hat{\mathbf{y}}_+|}{|\mathbf{y}_+| + |\mathbf{y}_- \cap \hat{\mathbf{y}}_+|} . \quad (5)$$

Here, $\mathbf{y}_+$ and $\hat{\mathbf{y}}_+$ represent the set of positive ground-truth pixels and corresponding segmentation pixels predicted to be positive, respectively, and $\mathbf{y}_-$ are the ground-truth pixels belonging to the negative class, for a specific annotated image.

The Jaccard Index is not differentiable and cannot be directly used when performing backpropagation. We therefore introduce a differentiable "soft" version of it

$$J_{\text{soft}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\sum_i \mathbf{y}_i s(\mathbf{y}_i, \hat{\mathbf{y}}_i)}{\sum_i \mathbf{y}_i + \sum_i (1 - \mathbf{y}_i) s(1 - \mathbf{y}_i, \hat{\mathbf{y}}_i)} , \quad (6)$$

where $s(\cdot, \cdot)$ is a similarity metric that outputs 0 if the values differ, and 1 if they are equal. In particular, we use the Radial Basis Function $s(a, b) = \exp(-\|a - b\|^2/\sigma)$, but other similarity metrics can also be adapted and used. Adapting other popular count-based metrics akin to the Jaccard Index, such as the F1 score and the Matthews correlation coefficient, to their equivalent differentiable formulation is straightforward.

We then write the classification loss of Eq. 1 as

$$c(\mathbf{y}, \hat{\mathbf{y}}) = 1 - J_{\text{soft}}(\mathbf{y}, \hat{\mathbf{y}}) , \quad (7)$$

which approximates the difference between the true Jaccard index and one, the value a perfect segmentation would return.

3. RESULTS

We test our method on different Transmission Electron Microscopy volumes. For **synapse segmentation**, we use a $750 \times 564 \times 750$ stack from the mouse cerebellum as the source domain, and a $1445 \times 987 \times 147$ stack from the mouse somatosensory cortex as the target domain, both at an isotropic 6.8 nm resolution. For **mitochondria segmentation**, we use a stack from the mouse striatum of size $853 \times 506 \times 496$ as the source domain and a $1024 \times 883 \times 165$ stack from mouse hippocampus as the target, both at an isotropic 5 nm resolution.

Our U-Nets are trained on individual 2D slices. For both synapses and mitochondria, we use the entire source domain stack as training data for the source domain stream. For the target domain, we spatially split the volume along one of the dimensions in two separate halves and use one as our test set. We then randomly select 1%, 5%, 10%, or 25% of the slices from the other half for training purposes. In this way, we avoid training the model on slices similar to the ones used at test time. We report results for these different amounts of training data in Fig. 3. For all other experiments, we take our target domain training set to be 10% of the slices.

Regularizing and sharing parameters from different layers yields different segmentation results. Fig. 4 shows the effects of applying different regularization schedules. We obtained the best results by regularizing weights in the down-coding layers of the U-Net, sharing parameters in the up-coding ones, and using correlation alignment to regularize the final feature maps, as shown in Fig. 2. We attribute this to the fact that the down-coding layers deal with the particular appearance of each domain, so those layers require a looser relationship across streams. Note that sharing all weights and using an MMD or a correlation alignment regularizer corresponds to the methods of [14] and [10], respectively and that our two-stream approach therefore outperforms them both.

Furthermore, using our proposed cost function, formulated as the differentiable approximation of the Jaccard index of Eq. 6, consistently performs better than relying on the more traditional cross-entropy during the whole optimization process, as shown in Fig. 5.

In Table 1, we compare our method against recent ones:

Fully-annotated Target Domain. (Full TD) Using all the available training data from the target domain to train a U-Net from scratch. This is our gold standard, that is, the best results that Domain Adaptation can achieve.

Experiment	*Full TD*	NoDA	Tuning	SA+B	SLSB	MIVC	CORAL	Ours
Synapses								
Cerebellum → Cortex	*0.7511*	0.2224	0.1973	0.1350	0.6705	0.5709	0.6862	**0.7230**
Mitochondria								
Hippocampus → Striatum	*0.8060*	0.5054	0.6979	0.2417	0.6336	0.5933	0.6700	**0.7456**

Table 1. **Jaccard indices for different methods using 10% of the target domain training data,** except in the case of *Full TD* where we used *all* the training data but no Domain Adaptation. Our approach consistently outperforms the other state-of-the-art methods.

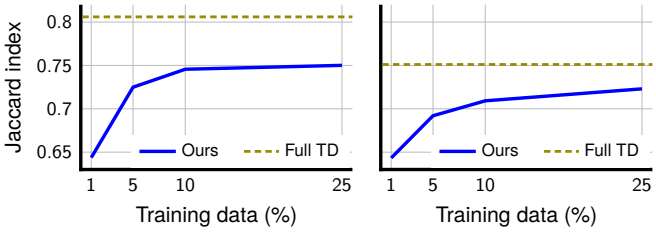

Fig. 3. **Using different amounts of target domain training data.** The Jaccard index when using from 1% to 25% of the training data and Domain Adaptation for the mitochondria (left) and synapse (right). The dotted line denotes the *Full TD* case (our gold standard).

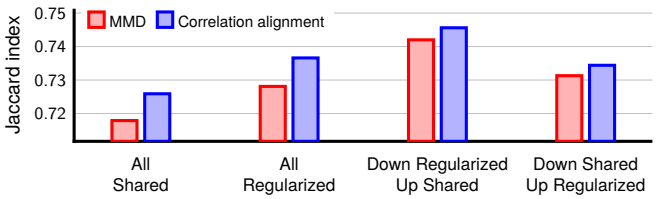

Fig. 4. **Parameter Sharing vs. Regularizing.** Layers can either share their weights or not. We plot the Jaccard indices for different configurations when coupled with the two different feature map regularizations. Results from the mitochondria segmentation problem.

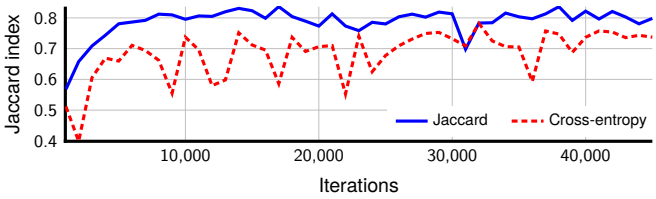

Fig. 5. **Evolution of the Jaccard index during training.** Our Jaccard-based cost function yields a consistently higher score than the standard cross-entropy based one.

No Domain Adaptation. (NoDA) A U-Net trained on the source domain is applied on the target domain as is.

Fine-tuning. (Tuning) Train a standard U-Net on the source domain and refine the weights using a small quantity of annotated data from the target domain.

Subspace Alignment + Boosting. (SA+B) Align the source and target domain PCA subspaces and find a decision boundary by boosting [15].

Shared Latent Space + Boosting. (SLSB) Learn mappings for both source and target domains onto a shared space, where a boosting classifier decides the pixel-wise class assignments [16].

Multiple Instance Visual Correspondence. (MIVC) Establish visual correspondences between image domains to drive the threshold adjustment of a boosted tree parameters from the source to the target domain [17].

Deep Correlation Alignment. (CORAL) Align the correlations of the data from different layers in a deep net [18].

Table 1 shows that our method outperforms all the others. Moreover, it yields Jaccard indices of 0.7230 and 0.7456 for synapses and mitochondria, respectively, which are close to 0.7511 and 0.8060 that we obtain using all the available training data in the target domain instead of only 10%.

Interestingly, the simple approach of just fine-tuning the source domain U-Net with 10% of target domain annotated data performs quite well for mitochondria segmentation but poorly for synapses. We attribute this to the fact that synapse segmentation is harder because other membrane-like structures can be easily mistaken for synapses. Therefore the detection heavily relies on context rather than only on texture.

Altogether, our results show that our method leverages the source domain information and requires very few annotated target examples to achieve a performance similar to that of a U-Net trained on fully-annotated data and superior to that of state-of-the-art methods.

4. CONCLUSION

Using a two-stream architecture for Domain Adaptation is advantageous, because it provides a convenient trade-off between exploiting shared elements of the problem and allowing independent streams to learn the specificities of their own domain. This approach is particularly well suited to the U-Net with its different levels of granularity in the learned representations, which encode different aspects of the image.

Furthermore, our approach is also generic and could be applied to many different architectures and modalities beyond electron microscopy. To give it even more flexibility and avoid *negative transfer*, which happens when trying to adapt information that is inherently not transferable between tasks, future work will focus on adapting not only the deep net parameters but also its architecture.

5. ACKNOWLEDGMENTS

This work was funded in part under a Swiss National Science Foundation grant and in part from the European Union?s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 720270 (HBP SGA1)

6. REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Conference on Medical Image Computing and Computer Assisted Intervention*, 2015.

[2] Ö. Çiçek, A. Abdulkadir, S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," in *Conference on Medical Image Computing and Computer Assisted Intervention*, 2016, pp. 424–432.

[3] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," *ArXiv e-prints*, June 2016.

[4] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour, "Auto-context convolutional neural network (auto-net) for brain extraction in magnetic resonance imaging," *IEEE Transactions on Medical Imaging*, 2017.

[5] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. HS Torr, and M. Chandraker, "Desire: Distant Future Prediction in Dynamic Scenes with Interacting Agents," in *Conference on Computer Vision and Pattern Recognition*, 2017.

[6] L. Liu, W. Lin, L. Wu, Y. Yu, and M.Y. Yang, "Unsupervised Deep Domain Adaptation for Pedestrian Detection," in *European Conference on Computer Vision*, 2016, pp. 676–691.

[7] M. Long, J. Wang, and M. I. Jordan, "Deep Transfer Learning with Joint Adaptation Networks," in *International Conference on Machine Learning*, 2017.

[8] Y. Ganin and V. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," in *International Conference on Machine Learning*, 2015, pp. 1180–1189.

[9] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial Discriminative Domain Adaptation," in *Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[10] B. Sun, J. Feng, and K. Saenko, "Correlation Alignment for Unsupervised Domain Adaptation," *arXiv Preprint*, vol. abs/1612.01939, 2016.

[11] P. Koniusz, Y. Tas, and F. Porikli, "Domain Adaptation by Mixture of Alignments of Second- or Higher-Order Scatter Tensors," in *Conference on Computer Vision and Pattern Recognition*, 2017.

[12] D.P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimisation," in *International Conference on Learning Representations*, 2015.

[13] Y. Shen, *Loss Functions for Binary Classification and Class Probability Estimation*, Ph.D. thesis, University of Pennsylvania, 2005.

[14] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep Domain Confusion: Maximizing for Domain Invariance," *arXiv Preprint*, 2014.

[15] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised Visual Domain Adaptation Using Subspace Alignment," in *International Conference on Computer Vision*, 2013, pp. 2960–2967.

[16] C. Becker, M. Christoudias, and P. Fua, "Domain Adaptation for Microscopy Imaging," *IEEE Transactions on Medical Imaging*, vol. 34, no. 5, pp. 1125–1139, 2015.

[17] R. Bermúdez-Chacón, C. Becker, M. Salzmann, and P. Fua, "Scalable Unsupervised Domain Adaptation for Electron Microscopy," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2016, pp. 326–334.

[18] B. Sun and K. Saenko, "Deep CORAL: Correlation Alignment for Deep Domain Adaptation," in *European Conference on Computer Vision*, 2016, pp. 443–450.