Fast and interpretable unsupervised domain adaptation for FIB-SEM cell segmentation

Alexandre Stenger, Luc Vedrenne, Patrick Schultz, Sylvain Faisan, Étienne Baudrier, Benoit Naegel

FAST AND INTERPRETABLE UNSUPERVISED DOMAIN ADAPTATION FOR FIB-SEM CELL SEGMENTATION

Alexandre Stenger [*†] *Luc Vedrenne* [*†] *Patrick Schultz* [‡] *Sylvain Faisan* [†]
Étienne Baudrier [†] *Benoît Naegel* [†]

[†] ICube, UMR 7357 CNRS, University of Strasbourg, France
[‡] IGMBC, UMR 7104 CNRS, U1258 INSERM, University of Strasbourg, France

ABSTRACT

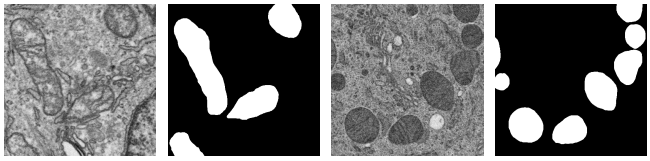
Electron microscopy imaging techniques allow biologists to obtain nanoscale slice volumes of cells in which biological structures can be statistically analyzed under the condition that the tedious task of segmentation can be automated. If a learned segmentation has good results on images obtained under the same acquisition process, variations in this acquisition can drastically affect the final image's features and can lead to a drop of segmentation performances. To tackle this problem, we propose an Unsupervised Domain Adaptation (UDA) framework based on Batch Normalization. Our method tested on mitochondria segmentation shows better results than state-of-the art methods and uses few computation resources, allowing a near real-time use. Furthermore, we propose a metric based on the Wasserstein distance to evaluate the effect of this normalization. We empirically show that this measure is log-linearly correlated with the drop of performances of the adapted segmentation and sheds light on the gain obtained by our UDA framework.

***Index Terms*—** Unsupervised domain adaptation, Cell mitochondria segmentation, FIB-SEM, quantifying domain shift.

1. INTRODUCTION

In the context of Focused Ion Beam with Scanning Electron Miscroscopy (FIB-SEM), an automatic segmentation network is able to predict segmentation mask when trained on cells imaged with a single acquisition process [1, 2]. However huge domain shifts can appear between two images depending on the acquisition protocol used. Thus, between chemical fixation (Fig. 1(a)) and cryo-fixation (Fig. 1(c)) processes, mitochondria change severely in terms of shape, contrast and texture (see segmentation masks Fig. 1(b)(d)), which leads to a drop of performance when a model trained on a database acquired with one protocol is applied to databases acquired with an other protocol. This situation leads either to call on tedious human segmentation, or to adapt the model in some way.

[*]Equal contribution



(a) FS1 (b) FS1 mask (c) FS2 (d) FS2 mask

Fig. 1. Example of 256×256 patches from our cells stacks FS1 and FS2 (a,c) as described in Section 4.1, with their mitochondria segmentation ground truth (b,d).

Domain Adaptation (DA) gained a broad interest with the great expansion of deep learning. In the context of semantic segmentation, most of the work still focus on adapting Cityscapes, GTA5 and SYNTHIA datasets [3]. Applying the same frameworks to biomedical images turns out to be very challenging, due to little datasets, lack of clear structure and high internal variation in this kind of data. As regards domain adaptation for cell segmentation, *Haq et al.* [4] introduced adversarial methods (based on GANs) to overcome domain shift between histological data, while *Li et al.* [5] used the same method. *Roels et al.* investigated Y-Net architecture for Electron Microscopy cell segmentation [6] along with *Franco et al.* brought in Attention Y-Net [7]. Those methods are hard to train, time consuming and energy-guzzling. On the other side, fast and easy domain adaptation based on BatchNorm statistics recomputation caught our attention. As much as this layer is able to reduce internal covariate shift [8], it can also reduce inter-domain shift [9]. The effectiveness of this method depends on the statistics estimate quality which requires a huge amount of data. However, we propose an adaptation framework based on BatchNorm recomputation and show that in some cases, it works even with small datasets, making it suitable for biomedical images. We also introduce a metric to quantify the quality of the statistics estimate, and show that it is a good indicator of the performances after adaptation. Furthermore, this framework is fast, needs few computation resources, does not need any new network training and can be run by biologists on their own.

2. RELATED WORK

BatchNorm (BN) Convolutional Neural Networks (CNN) were introduced by *Ioffe et al.* [8] in 2015. The idea of using BatchNorm in a domain adaptation framework came 3 years later with AdaBN [9]. This work was a baseline to multiple ideas: DSBN [10] creates a new CNN layer which has sub-BN layers for each domain. UBNA [11] improves this work by reweighting the BatchNorm computation according to its depth in the network. In our work, we simply use the original idea of AdaBN.

Nisar et al. [12] and *Stacke et al.* [13] proposed very similar metrics to quantify the domain shift. However, they both make the assumption that such a metric should be symmetric, which does not fit the observed performances. Moreover, in order to compute the usually very costly Wasserstein distance, they rely on a one dimensional case. They take the distribution of the *mean* activations of a neural network, which supposes the hypothesis – non validated to our knowledge – that these activations show very little variance. *Schneider and al.* [14] studied the effect of batch normalization, and leveraged the closed-form of the Wasserstein distance between Gaussian laws, which fits the context of neural networks activations, and introduced the source normalization to take into account the asymmetry of the task. However they focus on input corruptions for image classification.

3. METHODS

In a usual setup of supervised segmentation learning, we consider a training set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ with images $\mathbf{x}_i \in \mathbb{R}^{h\times w}$ and binary segmentation masks $\mathbf{y}_i \in \{0,1\}^{h\times w}$. We denote $\mathcal{P}$ the underlying joint data distribution and $\widehat{\mathcal{P}} = \frac{1}{n}\sum_i \delta_{\mathbf{x}_i,\mathbf{y}_i}$ its empirical counterpart. We aim at learning a function $f : \mathbb{R}^{h\times w} \mapsto \{0,1\}^{h\times w}$ predicting a mask from an image, through the minimization of the empirical risk $\widehat{\mathcal{R}}(f) = \mathcal{R}_{\widehat{\mathcal{P}}}(f) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\widehat{\mathcal{P}}}[L(\mathbf{y}, f(\mathbf{x}))]$, where L measures the discrepancy between the actual and the predicted label. In UDA, two domains $\mathcal{D}^s$ (source) and $\mathcal{D}^t$ (target) represented by their distributions $\mathcal{P}^s$ and $\mathcal{P}^t$ are considered. The aim is to learn a function f that works on $\mathcal{P}^t$ using only samples and labels from $\mathcal{P}^s$ and samples from $\mathcal{P}^t$.

3.1. Recall: BatchNorm layer

Let X be the input and Y denotes the output of the BatchNorm. In a classical deep learning inference scheme, the BatchNorm layer computes:

$$Y = \gamma \times \frac{X - \hat{\mu}}{\sqrt{\hat{\sigma}^2 + \epsilon}} + \beta \tag{1}$$

where γ and β are learnable parameters while $\hat{\mu}$ and $\hat{\sigma}^2$ are mean and variance estimated on the training set. Note that $\hat{\mu}$ and $\hat{\sigma}^2$ are usually fixed in a classical inference framework.

3.2. BatchNorm for a real time domain adaptation

We propose a two steps procedure:
First step: We train a U-Net [15] in a supervised way on the source domain using source image-label pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, $(\mathbf{x}_i, \mathbf{y}_i) \sim \widehat{\mathcal{P}}^s$.
Second step: We freeze all the U-Net layers but BatchNorms. For each layer, we recompute their statistics $\hat{\mu}$ and $\hat{\sigma}^2$ using all the target samples in the following way. Let B be the number of data batches. Let $\hat{\mu}_s$ and $\hat{\sigma}_s^2$ be the statistics computed on source domain as defined in (1). For each batch and at each BN layer, let $\mathbf{z}_t^i$ be the output of the previous convolutional layer, and $\hat{\mu}_t^i$, $(\hat{\sigma}_t^i)^2$ be the estimated statistics at step i on target domain. The BN re-computation for one layer is described in Alg. 1.

Algorithm 1 BatchNorm re-computation for UDA

$\hat{\mu}_t^0 \leftarrow \hat{\mu}_s$ ▷ Set initial μ and σ to the one
$(\hat{\sigma}_t^0)^2 \leftarrow (\hat{\sigma}_s)^2$ ▷ computed on source data
for $i \in \{1, \dots, B\}$ **do**
  $\mu_t^i \leftarrow \mathbb{E}[\mathbf{z}_t^i]$
  $(\sigma_t^i)^2 \leftarrow \mathrm{Var}[\mathbf{z}_t^i]$
  $\hat{\mu}_t^i \leftarrow (1-\alpha)\hat{\mu}_t^{i-1} + \alpha\mu_t^i$
  $(\hat{\sigma}_t^i)^2 \leftarrow (1-\alpha)(\hat{\sigma}_t^{i-1})^2 + \alpha(\sigma_t^i)^2$
end for
$\hat{\mu}_t \leftarrow \hat{\mu}_t^B$
$(\hat{\sigma}_t)^2 \leftarrow (\hat{\sigma}_t^B)^2$

3.3. Interpreting the effect of batch normalization

Let $p_s, p_t : \{0,1\}^{h\times w} \times \mathbb{R}^{h\times w} \mapsto \mathbb{R}^+$ be the probability densities of the source and target distributions respectively. We have $p_s(\mathbf{y}, \mathbf{x}) = p_s(\mathbf{y}|\mathbf{x})\, p_s(\mathbf{x})$ and $p_t(\mathbf{y}, \mathbf{x}) = p_t(\mathbf{y}|\mathbf{x})\, p_t(\mathbf{x})$. Following [16], we define the covariate shift as the two joint distributions having identical conditional probabilities but different marginals, that is $p_s(\mathbf{y}|\mathbf{x}) = p_t(\mathbf{y}|\mathbf{x})$ and $p_s(\mathbf{x}) \neq p_t(\mathbf{x})$. Let $\mathbb{E}_r$ and $\mathbb{V}_r$ be the true means and variances of the source (with $r = s$) and target (with $r = t$) distributions, so that $\widetilde{f}_r = (f(\mathbf{x}) - \mathbb{E}_r[f(\mathbf{x})])\,(\mathbb{V}_r[f(\mathbf{x})])^{-1/2}$ with $r \in \{s, t\}$ are the standardized distributions of a neural network's activations. If both the source and target activation distributions are Gaussian distributions, we have $p(\widetilde{f}_s(x)|x)p_s(x) \approx p(\widetilde{f}_t(x)|x)p_t(x)$ and the covariate shift is compensated. Our method aims at approximating such a compensation by estimating the target distribution's two first moments. We claim that the risk discrepancy between the source and target domains is almost entirely due to the quality of this estimation, up to the Gaussian assumption, which is usually seen as suitable for neural network's activations. This quality can be measured by the squared Wasserstein-2 distance between the true moments (μ_t, σ_t^2) and those estimated by the adapted network $(\hat{\mu}_t, \hat{\sigma}_t^2)$, which has the following

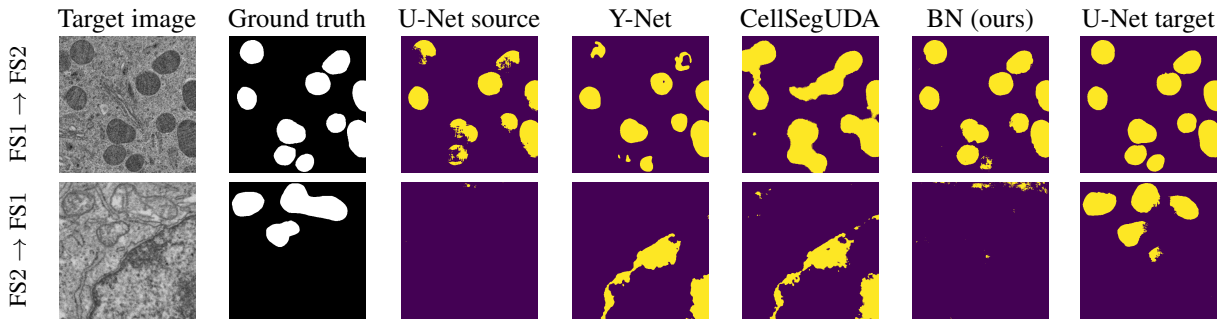


Fig. 2. Comparison of our method with other UDA frameworks. U-Net source refers to a U-Net trained on source domain and tested on the target domain, i.e the case without adaptation. BN refers to our BatchNorm adaptation. U-Net target refers to the case of supervised training on target domain.

closed form for Gaussian distributions [17]:

$$\mathcal{W}_2^2 = \sigma_t^2\hat{\sigma}_t^2 - 2\sigma_t\hat{\sigma}_t + (\mu_t - \hat{\mu}_t)^2 \tag{2}$$

Without Bessel's correction, we have (derived from [14])

$$\mathbb{E}\left[\mathcal{W}_2^2\right] = \sigma_t^2\mathbb{E}[\hat{\sigma}_t^2] - 2\sigma_t\mathbb{E}[\hat{\sigma}_t] + \mathbb{E}[(\mu_t - \hat{\mu}_t)^2] \tag{3}$$

$$= 2\sigma_t^2 - 2\sigma_t\mathbb{E}[\hat{\sigma}_t] \tag{4}$$

The Wasserstein distance depends on the magnitude of the target statistics [14], hence the need for a target-normalized Wasserstein distance:

$$\widetilde{\mathcal{W}}_2^2 = \frac{1}{\sigma_t^2}\mathcal{W}_2^2 = 1 + \frac{\hat{\sigma}_t^2}{\sigma_t} - 2\frac{\hat{\sigma}_t}{\sigma_t} + \frac{(\mu_t - \hat{\mu}_t)^2}{\sigma_t^2} \tag{5}$$

This normalization makes it an asymmetric measure. Because of the Gaussian assumption, it can easily be extended to a multivariate case, in order to be computed on features from a neural network. Unlike the 1D Wasserstein distance, it measures discrepancy between distributions of features, rather than distributions of their means. The measure $\widetilde{\mathcal{W}}_2^2$ quantifies the quality of the adaptation through the covariate shift reduction, rather than the distance between source and target domains. In that sense, it differs from [14]: instead of computing the distance on the adapted model between the source and target data, we compute the distance between an adapted model and a supervised one, on target data. Of course, the true target statistics are intractable, but can be precisely approximated by training a network on the target domain. In fact, the quality of this estimation itself is characterized by Eq. (4). It only depends on the term $\mathbb{E}[\hat{\sigma}_t]$ which decreases with the size of the target set, guaranteeing its robustness when the target set is sufficiently big.

Note that this distance requires training on the target domain. Indeed, it does not aim to quantify the domain shift a priori, but rather is an attempt to study the impact of the ubiquitous batch normalization.

4. EXPERIMENTS AND RESULTS

4.1. Data

We are interested in domain adaptation for mitochondria segmentation. To evaluate our adaptation framework, we perform our experiments on two different FIB-SEM images from HeLa cells. Each image is a stack of 2D slices. The FS1 image (FIB-SEM-1) was acquired with chemical fixation and a sample can be seen Fig. 1(a) along with its ground truth segmentation Fig. 1(b). FS2 was fixed under high pressure cryofixation (sample Fig 1(c) and its associated mask Fig 1(d)). Each image represents a domain. We note FS1 $\rightarrow$ FS2 the adaptation from domain FS1 to domain FS2 and FS2 $\rightarrow$ FS1 the reverse adaptation. For training, we use 250 annotated slices from each stack and extract 4000 patches of size 256×256 pixels. For testing, we use 60 annotated slices from each stack. We use IoU score to evaluate our experiments.

4.2. Unsupervised domain adaptation results

We compare our BatchNorm adaptation with state-of-the-art methods in the context of domain adaptation for cell segmentation: CellSegUDA [4], an adversarial based domain adaptation framework and the Y-Net [6]. Visual and numeric results are shown respectively in Figure 2 and Table 1. First, we observe that adaptation is performing well in one direction while not working at all in the other one. Indeed, the model supervised on target performs better on **FS2** than on **FS1** (Figure 2), reflecting the greater difficulty of the segmentation task on the latter. Usually, DA framework does not show such asymmetric behavior, due to the huge and less shifted datasets they are applied to. It is an important result to point out because DA is not yet well introduced in the biomedical field. For the working side of adaptation FS1 $\rightarrow$ FS2, Y-Net does not segment connected shapes, while CellSegUDA performs a less precise prediction, but more enclosing, which could be caused by the adversarial loss back-propagated through the

Architecture	FS1 → FS2	FS2 → FS1
U-Net (source trained)	0.556	0.006
Y-Net ([6])	0.614	0.014
CellSegUDA ([4])	0.673	**0.041**
BatchNorm (ours)	**0.736**	0.024
U-Net (target trained)	0.881	0.803

Table 1. IoU of each methods. We use the same notations as in Figure 2.

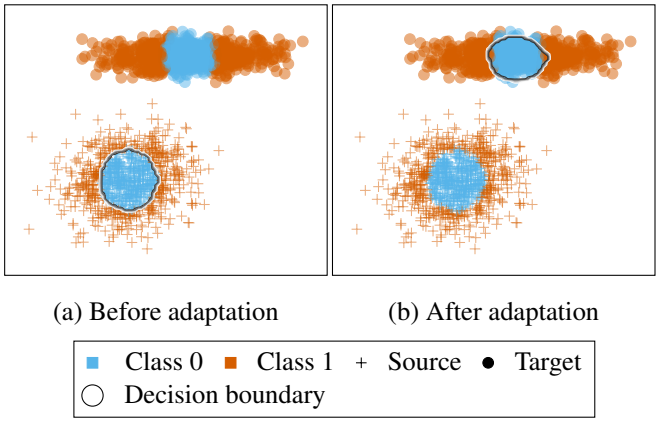
entire network. Our BatchNorm framework gets the best results and seems to be the most appropriate for this kind of domain shift. The next part aims to study this assumption and gives a more precise elucidation of those results. Finally, note that our BN method took 30 seconds to run on a CPU, while CellSegUDA and Y-Net needed respectively 20 and 6 hours to train on a GPU Nvidia RTX 3080Ti.

4.3. Interpretation of results

To study the relation between the target-normalized Wasserstein distance and the risk discrepancy under the Gaussian assumption, we conduct two experiments.

First, we construct a toy binary classification task, so-called *toy UDA*, where a Gaussian distribution $\mathcal{N}_1(\mu_1, \sigma_1^2)$ must be classified, with a circular decision boundary centered on μ_1 and a radius chosen so that the labels are perfectly balanced (to avoid target shift). This distribution is then "shifted" into a second one $\mathcal{N}_2(\mu_2, \sigma_2^2)$. A basic neural network composed of a stack of two Linear and BatchNorm layers trained on $\mathcal{N}_1$ tries to classify $\mathcal{N}_2$. We then visualize the effect of BatchNorm adaptation: the decision boundary of the classifier is computed before and after having adapted the BatchNorm layers statistics on $\mathcal{N}_2$ (Figure 3).

Then, we extend the previous experiment as follows.



(a) Before adaptation (b) After adaptation

Fig. 3. Decision boundary of a simple neural network for a binary classification task before (a) and after (b) adaptation.

$\mathcal{N}_1$ acts as a reference, and many others 2D Gaussian distributions of increasing discrepancy with $\mathcal{N}_1$ are sampled. This simulates true and estimated targets. We compute the target-normalized Wasserstein distance between the reference and every other Gaussian distributions, as well as the relative classification performance, that is, with P denoting the performance metric, $P_{\text{target}}/P_{\text{source}}$. We observe a quasi log-linearity between the target-normalized distance and the classification error discrepancy (blue dots in Figure 4).

We compare this log-linearity with the values observed in our segmentation task. The proposed measure takes larger values in the direction FS2 → FS1 than FS1 → FS2 for all methods, following the segmentation results. This emphasizes the ability of this metric to capture the asymmetry of the adaptation task. In the case FS1 → FS2, the relative performance of all the implemented methods fits this empirical rule. In the other direction, the relative performances remain far above the curve (Figure 4). As we observed above, this corresponds to the case where the adaptation is not working. We suggest that in this direction, the covariate shift alone does not characterize the domain shift.

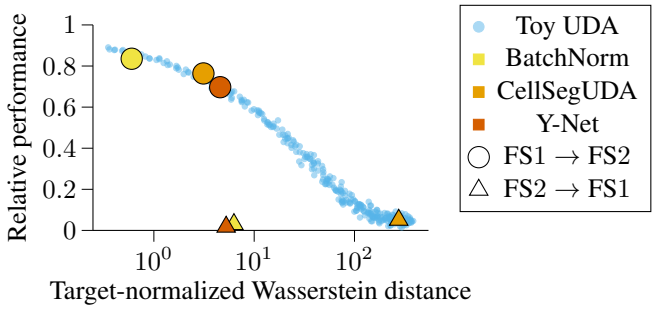


Fig. 4. Relative performance of implemented methods in both adaptation directions, compared to the toy UDA task.

5. CONCLUSION

We propose a BatchNorm based domain adaptation applied to small and complex dataset such as biomedical images. However the experiments show asymmetric behavior, highlighting the difficulty of the task on this kind of data. Indeed, adaptation works well in one direction and fails in the other one for all the tested methods. In the successful direction, our method obtains better results than state-of-the-art UDA methods on mitochondria segmentation. It is fast, easy to implement and can be run by biologists themselves, without GPU. To support our method results, we introduce a Wasserstein based measure that quantifies the distance between the statistics from a model adapted to the target and the one supervised on the target. A toy dataset shows a link between this measure and the performance of the tested methods. This result paves the way for a better understanding of domain shift.

6. ACKNOWLEDGEMENTS

This work is financed by the ITI-HealthTech, as part of the ITI 2021-2028 program of the University of Strasbourg, CNRS and Inserm, was supported by IdEx Unistra (ANR-10-IDEX-0002) and SFRI (STRAT'US project, ANR-20-SFRI-0012) under the framework of the French Investments for the Future Program. We thank F. Alpy, V. Mallouh and D. Spehner from the "Institut de Génétique et de Biologie Moléculaire et Cellulaire" for providing images and annotations used in this article. We acknowledge the use of resources of the French Infrastructure for Integrated Structural Biology FRISBI ANR-10-INBS-05 and of Instruct-ERIC. Conflict of Interest: The authors declare that they have no conflict of interest.

7. COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the study does not involve human- or animal-derived samples that require ethical approval. This article does not contain any studies involving human participants performed by any of the authors.

8. REFERENCES

[1] L. Heinrich, D. Bennett, D. Ackerman, W. Park, J. Bogovic, N. Eckstein, A. Petruncio, J. Clements, S. Pang, C. S. Xu, et al., "Whole-cell organelle segmentation in volume electron microscopy," *Nature*, vol. 599, no. 7883, pp. 141–146, 2021.

[2] C. Meyer, V. Mallouh, D. Spehner, E. Baudrier, P. Schultz, and B. Naegel, "Automatic multi class organelle segmentation for cellular fib-sem images," in *ISBI*. IEEE, 2021, pp. 668–672.

[3] M. Toldo, A. Maracani, U. Michieli, and P. Zanuttigh, "Unsupervised domain adaptation in semantic segmentation: a review," *Technologies*, vol. 8, no. 2, pp. 35, 2020.

[4] M. M. Haq and J. Huang, "Adversarial domain adaptation for cell segmentation," in *MIDL*. PMLR, 2020, pp. 277–287.

[5] C. Li, Y. Zhou, T. Shi, Y. Wu, M. Yang, and Z. Li, "Unsupervised domain adaptation for the histopathological cell segmentation through self-ensembling," in *MICCAI Workshop on Computational Pathology*. PMLR, 2021, pp. 151–158.

[6] J. Roels, J. Hennies, Y. Saeys, W. Philips, and A. Kreshuk, "Domain adaptive segmentation in volume electron microscopy imaging," in *ISBI*. IEEE, 2019, pp. 1519–1522.

[7] D. Franco-Barranco, J. Pastor-Tronch, A. González-Marfil, A. Muñoz-Barrutia, and I. Arganda-Carreras, "Deep learning based domain adaptation for mitochondria segmentation on em volumes," *Comput. Methods Programs Biomed.*, vol. 222, pp. 106949, 2022.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*. PMLR, 2015, pp. 448–456.

[9] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognit.*, vol. 80, pp. 109–117, 2018.

[10] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *CVPR*, 2019, pp. 7354–7362.

[11] M. Klingner, J.-A. Termöhlen, J. Ritterbach, and T. Fingscheidt, "Unsupervised batchnorm adaptation (ubna): A domain adaptation method for semantic segmentation without using source domain representations," in *WACV*, 2022, pp. 210–220.

[12] Z. Nisar, J. Vasiljević, P. Gançarski, and T. Lampert, "Towards measuring domain shift in histopathological stain translation in an unsupervised manner," in *ISBI*. IEEE, 2022, pp. 1–5.

[13] K. Stacke, G. Eilertsen, J. Unger, and C. Lundström, "Measuring domain shift for deep learning in histopathology," *IEEE J. Biomed. Health. Inf.*, vol. 25, no. 2, pp. 325–336, 2020.

[14] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge, "Improving robustness against common corruptions by covariate shift adaptation," *NeurIPS*, 2020.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*. Springer, 2015, pp. 234–241.

[16] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 521–530, 2012.

[17] G. Peyré and M. Cuturi, "Computational optimal transport," *Found. Trends Mach. Learn.*, vol. 11, no. 5-6, pp. 355–607, 2019.