Domain Adaptation for Microscopy Imaging

Carlos Becker, C. Mario Christoudias, and Pascal Fua

***Abstract*—Electron and Light Microscopy imaging can now deliver high-quality image stacks of neural structures. However, the amount of human annotation effort required to analyze them remains a major bottleneck. While Machine Learning algorithms can be used to help automate this process, they require training data, which is time-consuming to obtain manually, especially in image stacks. Furthermore, due to changing experimental conditions, successive stacks often exhibit differences that are severe enough to make it difficult to use a classifier trained for a specific one on another. This means that this tedious annotation process has to be repeated for each new stack. In this paper we present a domain adaptation algorithm that addresses this issue by effectively leveraging labeled examples across different acquisitions and significantly reducing the annotation requirements. Our approach can handle complex, non-linear image feature transformations and scales to large microscopy datasets that often involve high-dimensional feature spaces and large 3D data volumes. We evaluate our approach on four challenging Electron and Light Microscopy applications that exhibit very different image modalities and where annotation is very costly. Across all applications we achieve a significant improvement over the state-of-the-art Machine Learning methods and demonstrate our ability to greatly reduce human annotation effort.**

***Index Terms*—Electron and Light Microscopy, Domain Adaptation, Transfer Learning, Boosting, AdaBoost, Machine Learning**

I. Introduction

Imaging modalities such as Electron (EM) and Light Microscopy (LM) can now deliver high-quality, high-resolution image stacks of neural structures, such as the ones depicted by Fig. 1. Typically, a combination of manual and semi-automated segmentation or annotation tools such as [1], [2], [3] are then used to extract structures of interest. However, while the ever growing amount of available imagery should help unlock the secrets of neural functioning, the required amount of human annotation effort remains a major bottleneck. Therefore, there has been a great interest in automating the annotation process and most state-of-the-art algorithms nowadays rely on Machine Learning.

However, such algorithms still require significant amounts of manual annotation to train classifiers that can generalize well to unseen data. In microscopy, this can be a problem because the data preparation processes tend to be complicated and not easily repeatable, which means that a classifier trained on one acquisition will not perform very well on a new one, even when using the same modality. This is because Machine Learning normally relies on the fact that the training and run-time data samples are drawn from the same distribution.

For example, acquiring the Electron Microscopy (EM) images of brain structures shown in the top two rows of Fig. 1 requires tissue staining to increase contrast, followed by resin encasing before the acquisition. As a result, two samples of the same brain region acquired at different times may look significantly different due to differences in their preparation. This is even more true when the samples come from different parts of the brain, so that classifiers trained for one of them perform poorly on the other. While it is theoretically possible to gather new training data after each new image acquisition, it is impractical if high-throughput is desired because manual labeling of 3D image stacks is incredibly time-consuming.

A practical solution is to use Domain Adaptation [4] and acquire sufficient amounts of training data after *one specific* image acquisition and then to use it in conjunction with a small amount of additional training data that can be acquired quickly after each subsequent one to retrain the classifiers. Following the terminology of Domain Adaptation, we refer to the acquisition with sufficient training data as the *source domain* and the one with with limited supervision as the *target domain*. Our goal is then to exploit the labeled data in the source domain to learn an accurate classifier in the target domain despite having only a few labeled samples in the latter. While Domain Adaptation has received significant attention in the Machine Learning and Computer Vision communities, to our knowledge it has only recently been gaining interest in Medical Imaging, and remains largely unexplored for the acquisition problem depicted by Fig. 1. For many bio-medical applications, such as the ones considered in this work, we believe it is greatly needed to reduce annotation effort and make machine learning algorithms of practical use.

Current approaches to Domain Adaptation, and more generally *Transfer* or *Multi-Task* Learning [5], [6], [7], [8], treat classification in each domain as separate but related problems and exploit their relationship to learn from the supervised data available across all of them. Multi-task learning methods typically assume that the decision boundaries in each domain can be decomposed into a private and a shared term in a common feature space $\mathcal{X}$, as illustrated by Fig. 2(a). Unfortunately, acquisition artifacts like the ones shown in Fig. 1(a-d) may induce a significant, possibly non-linear transformation in feature space that may violate this assumption, as shown in Fig. 2(b).

To correct for these unknown transformations, we propose to learn a non-linear mapping of the features in each domain, such that samples can be mapped to a common discriminative latent space $\mathcal{Z}$, where a shared decision boundary exists, as depicted by Fig. 2(b). Such mappings seek to compensate for domain differences and acquisition artifacts, so that the

The authors are with the Computer Vision Lab, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne CH-1015, Switzerland.
E-mail: firstname.lastname@epfl.ch

Manuscript received Apr 18, 2014. Revised Oct 10, 2014.

This work was supported in part by the EU ERC project MicroNano.

classification task can be shared among them.

In this paper we develop a boosting-based approach [9], [10], [8] that can simultaneously learn the non-linear mappings as well as the shared decision boundary. We boost regression trees or stumps and model the domain-specific mappings with a set of common regression trees that are shared across domains, but whose thresholds have been adapted to each of them. Our approach does not require neither specific *a priori* knowledge about the mappings' global analytical form or explicit correspondences between training samples in the different domains. This is unlike more conventional Latent Variable Models that can be applied to learn a shared mapping, such as those based on Canonical Correlation Analysis (CCA) [11], [12]. These methods generally require instance-level correspondences which limits their applicability because they rarely are explicitly available and can be difficult to establish reliably. The situation is further complicated by the fact that the unknown mappings often are non-linear. Although kernel methods can handle this in theory [11], [13], [14], they require kernel functions that can be difficult to specify *a priori*. Furthermore, the computational complexity of kernel methods scales quadratically with the number of training samples, thus limiting their applicability when there are large amounts of data available in the source domain.

In contrast, our approach easily scales to large training datasets and high-dimensional feature spaces, often found in medical imaging [15], [16], [17]. Moreover, unlike other methods, our approach does not require tuning any parameter except those needed by the boosted classifier it relies on. In practice, this is an important advantage, since cross-validation can be unreliable when few labeled data is afforded in the target domain.

We evaluate our approach on the four challenging bio-medical applications depicted by Fig. 1.

- The first two applications are mitochondria and synapse segmentation from large 3D Electron Microscopy (EM) stacks of neural rat tissue where the task is to classify voxels that belong to either structure of interest. We use as source and target domains stacks coming from different parts of the brain, each exhibiting different acquisition artifacts, making it difficult to apply standard machine learning to learn a classifier that generalizes across image stacks and for which domain adaptation is required to reduce costly annotation effort.
- We also consider the detection of Olfactory Projection Fibers from two-photon Light Microscopy stacks and axons in Brightfield imagery. Although these represent two very different imaging modalities, the task is the same in each, where we want to classify voxels as to whether they belong to tubular structures. To showcase the power of our approach, we use as our source domain the 2D aerial images of roads shown in the bottom left of Fig. 1. This is of practical significance for two reasons. First, the appearance of the roads is very different from that of the fibers or dendrites. Second, delineating semi-automatically in 2D is much easier than delineating in 3D and our method makes its possible to leverage this easily obtainable 2D data to perform the much harder 3D task.

We will show that our approach consistently outperforms recent multi-task learning techniques [8], [18], [11] across this wide range of applications. Our approach was first introduced in a conference paper [19]. Here, we include an extended discussion of related work, a more detailed description of our method, and a more extensive evaluation including two new bio-medical applications and a comparison to an additional baseline method [18].

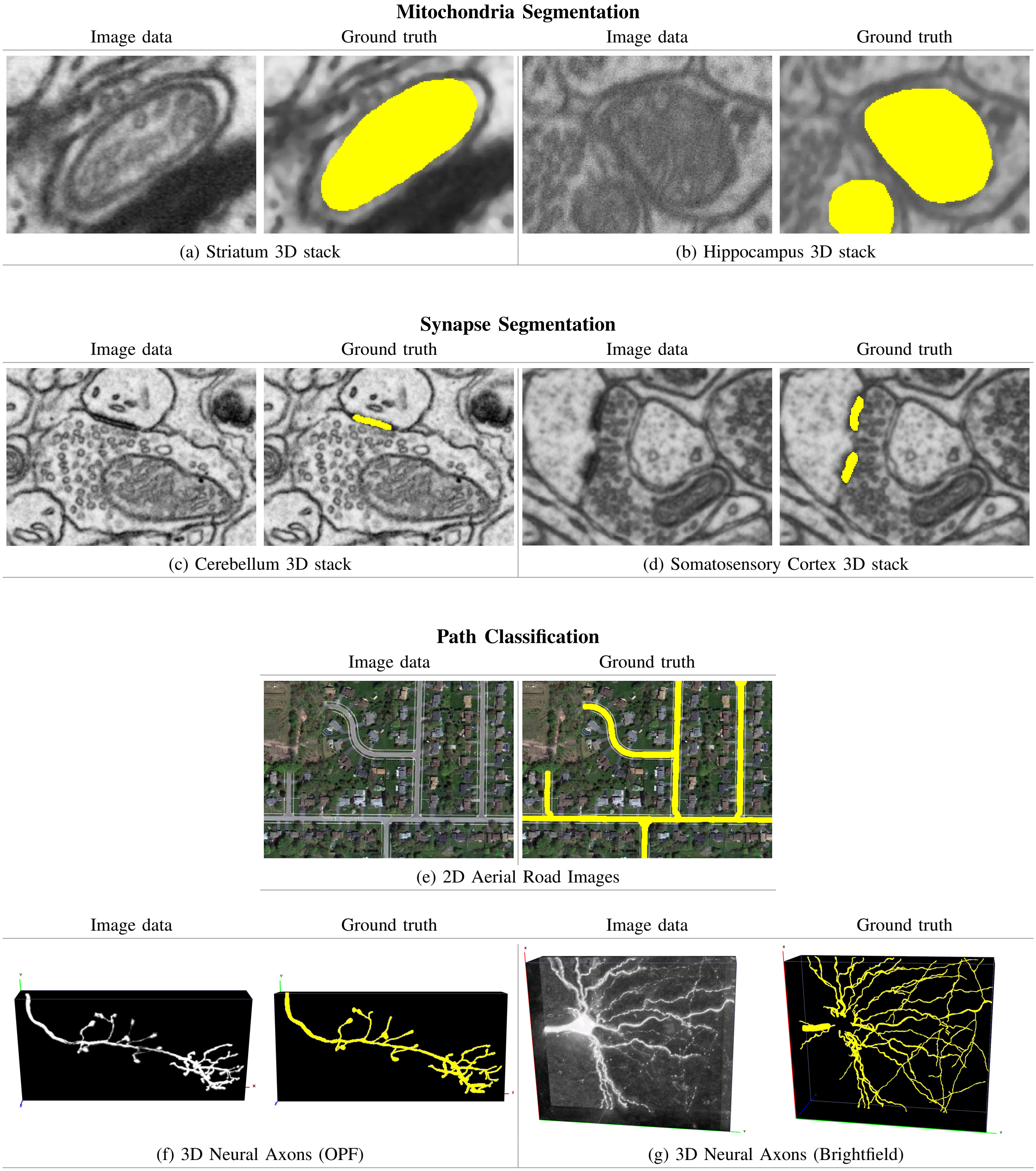
II. Related Work

Domain Adaptation and more generally Multi-Task Learning have received considerable attention in the Machine Learning and Computer Vision communities. However, they have only recently been gaining interest in Medical Imaging [20], [21], [22], and remain largely unexplored for the acquisition problem. In this section we briefly review the state-of-the-art methods in each of these communities and clarify their connections to our work.

Initial approaches to multi-task learning exploited supervised data from related tasks to define a form of regularization in the target problem [5], [23]. In this setting, related tasks, also sometimes referred to as *auxiliary problems* [6], are used to learn a latent representation and find discriminative features shared across tasks. This representation is then *transferred* to the target task to help regularize the solution and learn from fewer labeled examples. The success of these approaches crucially hinges on the ability to define auxiliary tasks. Although this can be easily done in certain situations, as in [6], in many cases it is unclear how to generate them.

More recent multi-task learning methods jointly optimize over both the shared and task-specific components of each task [7], [24], [8], [25]. In [7] it was shown how the two step iterative optimization of [6] can be cast into a single convex optimization problem. In particular, for each task their approach computes a linear decision boundary defined as a linear combination between a shared hyperplane, shared across tasks, and a task-specific one in either the original or a kernelized feature space. This idea was later further generalized to allow for more generic forms [24], [26], [27], [25], as in [24] that investigated the use of a hierarchically combined decision boundary.

For many problems, such as those common to domain adaptation [4], the decision problem is in fact the same across tasks, however, the features of each task have undergone some unknown transformation. Feature-based approaches seek to uncover this transformation by learning a mapping between the features across tasks [28], [29], [14]. A cross-domain Mahalanobis distance metric was introduced in [28] that leverages across-task correspondences to learn a transformation from the source to target domain. A similar method was later developed in [30] to handle cross-domain feature spaces of a different dimensionality. [31] devises a surrogate kernel approach for modeling covariate shift that matches domain feature distributions in Hilbert space and avoids the need for cross-domain correspondences. Shared latent variable models have also been proposed to learn a shared representation across multiple feature sources or tasks [11], [29], [13], [14], [32].



(a) Striatum 3D stack

(b) Hippocampus 3D stack

(c) Cerebellum 3D stack

(d) Somatosensory Cortex 3D stack

(e) 2D Aerial Road Images

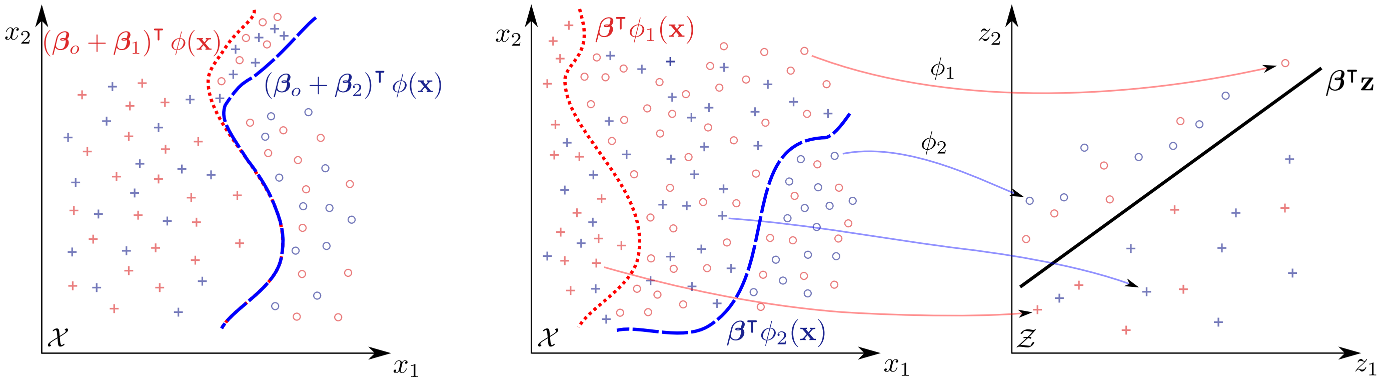
(f) 3D Neural Axons (OPF)

(g) 3D Neural Axons (Brightfield)

Fig. 1. Segmentation and path classification applications we consider: (a,b,c,d) slice cuts from four 3D Electron Microscopy acquisitions from different parts of the brain of a rat. Each 3D stack contains millions of voxels to be classified. (e,f,g) 2D aerial road images and 3D neural axons from Olfactory Projection Fibers (OPF) and Brightfield microscopy. Ground truth positive samples shown in yellow. Best viewed in color.

Feature-based methods generally require well established cross-domain correspondences and/or model non-linearities using the kernel-trick that relies on the selection of a predefined kernel function and is difficult to scale to large datasets. In this paper, we pursue a discriminative learning approach that does not require explicit cross-domain correspondences, and exploits the *boosting-trick* [8], [9] to handle non-linearities and learn a shared representation across tasks, overcoming these limitations.

The use of boosting for multi-task learning was explored


x_2
$(\boldsymbol{\beta}_o + \boldsymbol{\beta}_1)^\intercal \, \phi(\mathbf{x})$
$(\boldsymbol{\beta}_o + \boldsymbol{\beta}_2)^\intercal \, \phi(\mathbf{x})$
$\mathcal{X}$
x_1
x_2
$\boldsymbol{\beta}^\intercal \phi_1(\mathbf{x})$
ϕ_1
ϕ_2
$\mathcal{X}$
$\boldsymbol{\beta}^\intercal \phi_2(\mathbf{x})$
x_1
z_2
$\boldsymbol{\beta}^\intercal \mathbf{z}$
$\mathcal{Z}$
z_1

(a) Standard Multi-task Learning

(b) Domain Adaptation

Fig. 2. Illustration of the difference between (a) standard Multi-task Learning (MTL) and (b) our Domain Adaptation (DA) approach on two tasks. The feature points for each task are shown in either red or blue, and each point is drawn as a cross or circle depending on its class. The dotted and dashed curves represent the decision boundaries of each task. MTL assumes a single, pre-defined transformation $\phi(\mathbf{x}) : \mathcal{X} \to \mathcal{Z}$ and learns shared and task-specific linear boundaries in $\mathcal{Z}$, namely $\boldsymbol{\beta}_o$, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2 \in \mathcal{Z}$. In contrast, our DA approach learns a single linear boundary $\boldsymbol{\beta}$ in a common feature space $\mathcal{Z}$, and task-specific mappings $\phi_1(\mathbf{x})$, $\phi_2(\mathbf{x}) : \mathcal{X} \to \mathcal{Z}$. Best viewed in color.

in [8] as an alternative to kernel-based approaches. For each task they optimize for a shared and task-specific decision boundary, as in [7], except that non-linearities are modeled using a boosted feature space. As with other methods, however, additional parameters are required to control the degree of sharing between tasks and can be difficult to set, especially when one or more tasks have only a few labeled samples. Similarly, [33] devises a boosting-based domain adaptation method assuming that the source domain contains out-dated samples that are down-weighted during learning. Even though [8] and [33] address different adaptation problems, both assume that there exist weak learners that can be shared between domains or tasks as a means of regularizing inter-domain learning, which may not be true in cases such as those shown in Fig. 1.

Another interesting method is that of [18] that learns a boosted regressor for web search ranking, using regression tree weak learners. They adapt boosted regression trees learned in the source domain to the target domain by interpolating the thresholds and leaf-node responses in each tree. In this way, similar to [8], [33], they seek to recover the private component of the target domain that in our problem corresponds to the unwanted acquisition artifacts. Furthermore, they require an interpolation parameter that weights the different domains, which, as with [8], can be difficult to cross-validate when afforded few training samples in the target domain.

In contrast to [8], [33], [18], we learn a mapping to a shared feature space that preserves the task-relevant features and learn the thresholds across domains by jointly minimizing a common loss that does not rely on a pre-defined adaptation parameter.

Within the Medical Imaging community, domain adaptation has been applied to augment training data from synthetically generated samples [22], [34], as well as to modality fusion [35] and multi-task anomaly detection in CT and ultrasound [20]. However, the data acquisition problem depicted by Fig. 1 remains largely unexplored. An exception is [21], which targets image segmentation using labeled samples obtained across multiple image acquisitions. However, [21] is based on a sample re-weighting scheme that relies on having several labeled acquisitions, not always available in large numbers for EM and LM, and is difficult to scale to large training datasets. In contrast, our approach can leverage as little as one source acquisition, and is also easily amenable to large data volumes and high dimensional feature spaces.

III. Our Approach

In this section we first introduce our shared latent space model. We then discuss the specific weak learners we use.

A. Shared Latent Space Model

We consider the problem of learning a binary decision function from supervised data collected across multiple domains. In our setting, each task is an instance of the same underlying decision problem, however, its features are assumed to have undergone some unknown non-linear transformation. Even though *task* and *domain* originally denote different concepts, in the remainder of this paper we use these terms interchangeably as is generally done in the literature [33], [8].

Assume that we are given training samples $X_t = \{\mathbf{x}_i^t, y_i^t\}_{i=1}^{N_t}$ from $t = 1, \ldots, T$ tasks, where $\mathbf{x}_i^t \in \mathbb{R}^D$ represents a feature vector for sample i in task t and $y_i^t \in \{-1, 1\}$ its label. For each task, we seek to learn a non-linear transformation $\phi_t(\mathbf{x}^t)$ that maps $\mathbf{x}^t$ to a common, task-independent feature space $\mathcal{Z}$, accounting for unwanted feature transformations. Instead of relying on pre-defined kernel functions, we model each transformation using a set of M task-specific non-linear functions $\mathcal{H}_t = \{h_1^t, \ldots, h_M^t\}$, $h_j^t : \mathbb{R}^D \to \mathbb{R}$, to define $\phi_t : \mathcal{X}_t \to \mathcal{Z}$ as $\phi_t(\mathbf{x}^t) = [h_1^t(\mathbf{x}^t), \ldots, h_M^t(\mathbf{x}^t)]^\intercal$. In the context of boosting, the $h_j^t(\cdot)$ represent all the possible weak learners and $M = |\mathcal{H}_t|$ is the total number of them, which can be large and possibly infinite.

In this paper we consider functions of the form

$$h_j^t(\mathbf{x}^t) = h_j(\mathbf{x}^t - \boldsymbol{\tau}_j^t), \quad j = 1, \ldots, M \, , \tag{1}$$

where $\mathcal{H} = \{h_1, \ldots, h_M\}$ are shared across tasks, while $\boldsymbol{\tau}_j^t \in \mathbb{R}^D$ are task-specific.

An interpretation of Eq. 1 is that all tasks share mid-level representations of the decision boundary, namely the weak learners $h_j(\cdot)$. However, for those representations to be shared among domains, the low level responses must be adapted to compensate for varying imaging conditions. The latter is accomplished through the $\boldsymbol{\tau}_j^t$. Empirically we found this model to work well in cases of domain shift resulting from differences in acquisition artifacts, such as those typically encountered in bio-medical applications.

Assuming that the problem is linearly separable in $\mathcal{Z}$, the predictive function $f_t(\cdot) : \mathbb{R}^D \to \mathbb{R}$ for each task can then be written as

$$f_t(\mathbf{x}) = \boldsymbol{\beta}^\intercal\, \phi_t(\mathbf{x}^t) = \sum_{j=1}^{M} \beta_j h_j(\mathbf{x}^t - \boldsymbol{\tau}_j^t)\,, \tag{2}$$

where $\boldsymbol{\beta} \in \mathbb{R}^M$ is a linear decision boundary in $\mathcal{Z}$ that is common to all tasks, and corresponds to a non-linear boundary in each of the original task-specific input spaces via the ϕ_t. This contrasts with previous approaches to multi-task learning such as [7], [8] that learn a separate decision boundary per task, β_t, in a common input space $\phi(\cdot)$, as shown in Fig. 2. In the results section we show that our approach performs better for applications such as those depicted by Fig. 1.

We learn the functions $f_t(\cdot)$ by minimizing the exponential loss on the training data across each task

$$\boldsymbol{\beta}^*, \Gamma^* = \min_{\boldsymbol{\beta},\Gamma} \sum_{t=1}^{T} c_t\, L(\boldsymbol{\beta}, \Gamma_t; X_t), \tag{3}$$

where $c_t \in \mathbb{R}$ is the weight of task t, and

$$\begin{aligned} L(\boldsymbol{\beta}, \Gamma_t; X_t) \;&=\; \sum_{i=1}^{N_t} \exp\big[-y_i^t f_t(\mathbf{x}_i^t)\big] &(4)\\ &=\; \sum_{i=1}^{N_t} \exp\Big[-y_i^t \sum_{j=1}^{M} \beta_j h_j(\mathbf{x}_i^t - \boldsymbol{\tau}_j^t)\Big], &(5)\end{aligned}$$

with $\Gamma = [\Gamma_1, \ldots, \Gamma_T]$ and $\Gamma_t = [\boldsymbol{\tau}_1^t, \ldots, \boldsymbol{\tau}_M^t]$.

The explicit minimization of Eq. (3) can be very difficult because in practice the dimensionality of $\boldsymbol{\beta}$ can be prohibitively large and the h_j's are typically discontinuous and highly non-linear. Luckily, this is a problem for which boosting is particularly well suited [9]. It has been shown to be an effective method for constructing a highly accurate classifier from a possibly large collection of weak predictors. Similar to the kernel-trick, the resulting *boosting-trick* [9], [10], [8] can be used to define a non-linear mapping to a high dimensional feature space in which we assume the data to be linearly separable. Unlike the kernel-trick, however, the boosting-trick defines an explicit mapping for which $\boldsymbol{\beta}$ is assumed to be sparse [36], [8]. Within this setting, each h_j can be interpreted as a weak non-linear predictor of the task label.

We use gradient boosting [9], [10] to solve for $f_t(\cdot)$. Given any twice-differentiable loss function, gradient boosting minimizes the loss in a stage-wise manner for iterations $k = 1$ to K. More specifically, we use the quadratic approximation introduced by [10]. When applied to minimizing Eq. (3), the goal at each boosting iteration is to find the weak learner $\tilde{h} \in \mathcal{H}$ and the set $\{\tilde{\boldsymbol{\tau}}^1, \ldots, \tilde{\boldsymbol{\tau}}^T\}$ that minimize

$$\sum_{t=1}^{T} \left(\sum_{i=1}^{N^t} w_{ik}^t \left[\tilde{h}(\mathbf{x}^t - \tilde{\boldsymbol{\tau}}^t) - r_{ik}^t \right]^2 \right), \tag{6}$$

where w_{ik}^t and r_{ik}^t can be computed by differentiating the loss of Eq. (5), obtaining $w_{ik}^t = c_t\, e^{-y_i^t f_t(\mathbf{x}_i^t)}$ and $r_{ik}^t = y_i^t$. Once $\tilde{h}$ and $\{\tilde{\boldsymbol{\tau}}^1, \ldots, \tilde{\boldsymbol{\tau}}^T\}$ are found, a line-search procedure is applied to determine the optimal weighting for $\tilde{h}$ and the predictive functions $f_t(\cdot)$ are updated, as described in Alg. 1. Shrinkage may be applied to help regularize the solution, particularly when using powerful weak learners such as regression trees [9].

Our proposed approach is summarized in Alg. 1. The main difficulty in implementing it is at line 4. Finding the optimal values of $\tilde{h}$ and $\{\tilde{\boldsymbol{\tau}}^1, \ldots, \tilde{\boldsymbol{\tau}}^T\}$ that minimize Eq. 6 can be very expensive, depending on the type of weak learners employed. In the next section we show that regression trees and boosted stumps can overcome this problem.

B. Weak Learners

In this section we introduce the weak learners used in our approach and their corresponding training procedure. We consider both regression tree and decision stump weak learners.

Regression trees have proven very effective when used as weak learners in conjunction with gradient boosting [37]. An important advantage is that training regression trees involves almost no parameter tuning and is very efficient when a greedy top-down approach is used [9].

Decision stumps are a special case of single-level regression trees. Despite their simplicity, they have been shown to achieve high performance in challenging tasks such as face and object detection [38], [39]. In cases where feature dimensionality D is very large, decision stumps may be preferred to regression trees to reduce training time.

1) Regression Trees: We use trees whose splits operate on a single dimension of the feature vector, also known as *orthogonal splits*, and follow the top-down greedy tree learning approach described in [9]. The top split is learned first so as to minimize

$$\begin{aligned} \operatorname*{argmin}_{\substack{n\in\{1,\ldots,D\},\\ \eta_1,\eta_2,\{\tau^1,\ldots,\tau^T\}}} \sum_{t=1}^{T} \Bigg(&\sum_{i=1}^{N_t} \mathbf{1}_{\{\mathbf{x}_i^t[n]-\tau^t\}}\, w_{ik}^t \left[\eta_1 - r_{ik}^t\right]^2 \\ &+ \sum_{i=1}^{N_t} \bar{\mathbf{1}}_{\{\mathbf{x}_i^t[n]-\tau^t\}}\, w_{ik}^t \left[\eta_2 - r_{ik}^t\right]^2 \Bigg), \end{aligned} \tag{7}$$

where $\mathbf{x}[n] \in \mathbb{R}$ denotes the value of the n^{th} dimension of $\mathbf{x}$, $\mathbf{1}_{\{\cdot\}}$ is the step function, and $\bar{\mathbf{1}}_{\{\cdot\}} = 1 - \mathbf{1}_{\{\cdot\}}$. As in Eq. 6, the weights, w_{ik}^t, and residuals, r_{ik}^t, are computed by differentiating the loss of Eq. (5). The difference with classic regression trees is that, in addition to learning the values of η_1, η_2 and n, our approach requires the tree to also learn a threshold $\tau^t \in \mathbb{R}$ per task. Given that each split operates on a single attribute $\mathbf{x}[n]$, the resulting $\tilde{\boldsymbol{\tau}}^t$ is sparse, and learned one component at a time as the tree is built.

Algorithm 1 Non-Linear Domain Adaptation with Boosting

```
Input: Training samples and labels for $T$ tasks $X_t = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{N_t}$
       Task weights $c_t \in \mathbb{R}$ for each task $t$. Typically $c_t = 1 \;\forall\; t$
       Number of iterations $K$, shrinkage factor $0 < \gamma \leq 1$
Set $f_t(\cdot) = 0 \;\forall\; t = 1, \dots, T$
for $k = 1$ to $K$ do
    Let $w_{ik}^t = c_t \, e^{-y_i^t f_t(\mathbf{x}_i^t)}$ and $r_{ik}^t = y_i^t$
    Find weak learner and parameters:
        $\left\{\tilde{h}(\cdot), \tilde{\boldsymbol{\tau}}^1, \dots, \tilde{\boldsymbol{\tau}}^T\right\} = \underset{h \in \mathcal{H}, \boldsymbol{\tau}^1, \dots, \boldsymbol{\tau}^T}{\operatorname{argmin}} \sum_{t=1}^{T} \sum_{i=1}^{N_t} w_{ik}^t \left[h(\mathbf{x}_i^t - \boldsymbol{\tau}^t) - r_{ik}^t\right]^2$
    Find $\tilde{\alpha}$ through line search:
        $\tilde{\alpha} = \underset{\alpha}{\operatorname{argmin}} \sum_{t=1}^{T} \sum_{i=1}^{N_t} c_t \exp\left[-y_i^t \Big(f_t(\mathbf{x}_i^t) + \alpha \, \tilde{h}(\mathbf{x}_i^t - \tilde{\boldsymbol{\tau}}^t)\Big)\right]$
    Set $\tilde{\beta} = \gamma \, \tilde{\alpha}$
    Update $f_t(\cdot) = f_t(\cdot) + \tilde{\beta} \, \tilde{h}(\,\cdot - \tilde{\boldsymbol{\tau}}^t) \quad \forall\; t = 1, \dots, T$
end for
return $f_t(\cdot) \quad \forall\; t = 1, \dots, T$
```

Once the top split is learned, new splits are learned on children leaves recursively. This process stops when the maximum depth L, given as a parameter, is reached, or there are not enough samples to learn a new node at a given leaf.

2) Decision Stumps: Decision stumps consist of a single split and return values $\eta_1, \eta_2 = \pm 1$. If also $r_{ik}^t = \pm 1$, which is true when boosting with the exponential loss, then it can be demonstrated that minimizing Eq (7) can be separated into T independent minimization problems for all D attributes for each n. Once this is done, a quick search can be performed to determine the n that minimizes Eq. (7).

This makes decision stumps feasible for large-scale applications with very high dimensional feature spaces.

When using the exponential loss in conjunction with decision stumps, Alg. 1 reduces to a procedure similar to classic AdaBoost [40], with the exception that weak learner search is done in the multi-task manner described above.

3) Training Complexity: Both regression trees and decision stumps require storage linear in the number of training samples in each task. Similarly, the time complexity of training a single decision stump is linear in the total number of training examples or $\mathcal{O}(\bar{N})$, with

$$\bar{N} = \sum_{t=1}^{T} N_t. \tag{8}$$

This contrasts with kernel machines whose storage and time complexity is $\mathcal{O}(\bar{N}^2)$.

Regression trees are more costly to train as they require a joint search over the thresholds across tasks whose complexity is $\mathcal{O}(\prod_t N_t)$. In this work we mainly focus on applications containing a single source and target task, representative of the most common domain adaptation setting. In such cases $T = 2$ and the complexity of training regression trees remains smaller than that of kernel machines, since $N_1 N_2 < (N_1)^2 + (N_2)^2 + 2N_1N_2$.

For $T > 2$, regression trees become costly and their complexity can grow faster than $\bar{N}^2$. It may still be possible to train them efficiently, but we leave this as a topic for future work.

IV. Evaluation

We evaluated our approach on four challenging and representative domain adaptation problems for which annotation is very time-consuming. We first describe the datasets, our experimental setup and baselines, and finally present and discuss our results.

A. Datasets

The experiments used for evaluation are described below, and Table I summarizes the different datasets employed, their characteristics and amount of labeled data available.

1) Mitochondria and Synapse Segmentation: Mitochondria and synapses are structures that play an important role in cellular functioning. Here, the task is to segment mitochondria and synapses from large 3D Electron Microscopy (EM) stacks, acquired from the brain of a rat. Example slice cuts are presented in Fig. 1(a-d). As in the path classification problem, 3D annotations are time-consuming and exploiting already-annotated stacks is essential to reduce labeling effort and speed up analysis.

We use our boosting-based method with contextual features [17], which is designed for 3D stacks and whose source code is publicly available. This method is based on boosted stumps, which makes it very efficient at both train and test time. Our contextual features capture information about the context surrounding the voxel of interest, which is particularly informative to segment synapses [17].

Experiment		**Modality / Acquisition**	**Image(s)/stack size**	**Available Labeled Data** (pos / neg samples)
Mitochondria Segmentation	Source Domain	EM / Striatum	853×506×496	39 mitochondria (15k, 275k)
	Target Domain			
	Train	EM / Hippocampus	1024×653×165	10 mitochondria (3k, 12k)
	Test		1024×883×165	42 mitochondria (14k, 265k)
Synapse Segmentation	Source Domain	EM / Cerebellum	853×506×496	11 synapses (3k, 645k)
	Target Domain			
	Train	EM / Som. Cortex	1024×653×165	10 synapses (7k, 510k)
	Test		1024×883×165	28 synapses (35k, 6M)
Paths: Brightfield to OPF	Source Domain	Brightfield / Neural axons	6 images ≈ 800×800×90 each	30k paths (15k, 15k)
	Target Domain			
	Train	OPF / Neural axons	4 stacks ≈ 512×512×70 each	20k paths (10k, 10k)
	Test		4 stacks ≈ 512×512×70 each	20k paths (10k, 10k)
Paths: OPF to Brightfield	Source Domain	OPF / Neural axons	8 stacks ≈ 512×512×70 each	40k paths (20k, 20k)
	Target Domain			
	Train	Brightfield / Neural axons	3 stacks ≈ 800×750×80 each	15k paths (7.5k, 7.5k)
	Test		3 stacks ≈ 700×900×100 each	15k paths (7.5k, 7.5k)
Paths: Roads to OPF	Source Domain	Aerial Images / Roads	6 images ≈ 750×850 each	30k paths (15k, 15k)
	Target Domain			
	Train	OPF / Neural axons	4 stacks ≈ 512×512×70 each	20k paths (10k, 10k)
	Test		4 stacks ≈ 512×512×70 each	20k paths (10k, 10k)
Paths: Roads to Brightfield	Source Domain	Aerial Images / Roads	6 images ≈ 750×850 each	30k paths (15k, 15k)
	Target Domain			
	Train	Brightfield / Neural axons	3 stacks ≈ 800×750×80 each	15k paths (7.5k, 7.5k)
	Test		3 stacks ≈ 700×900×100 each	15k paths (7.5k, 7.5k)

TABLE I
DESCRIPTION OF THE SEGMENTATION AND PATH CLASSIFICATION EXPERIMENTS USED FOR EVALUATION.

For mitochondria segmentation, the source domain is a fully-labeled EM stack from the Striatum region of 853x506x496 voxels with 39 labeled mitochondria. The target domain consists of two stacks acquired from the Hippocampus, one a training stack of size 1024x653x165 voxels and the other a test stack of size 1024x883x165 voxels, with 10 and 42 labeled mitochondria in each respectively. The target test volume is fully-labeled, while the training one is partially annotated, similar to a real scenario.

For synapse segmentation, the source domain is a stack acquired from the Cerebellum of size 1027x987x219 voxels with 11 labeled synapses, and the target domain is an EM stack from the Somatosensory Cortex region, which was divided in training and testing stacks, each of size 750x564x750 and 655x429x250, with 10 and 28 labeled synapses respectively.

2) Path Classification: Tracing arbors of curvilinear structures is a well studied problem that finds applications in a broad range of fields from neuroscience to photogrammetry. In earlier work [41] we showed the advantage of using a path classifier and a mixed integer programming formulation to automatically trace such structures. Within this framework, machine learning is used to predict, based on image evidence, if a tubular path between two points in the image belongs to a curvilinear structure or not. We constructed descriptors named Histogram of Gradient Deviations (HGD) designed to capture several characteristics of tubular structures in images. From the HGDs generated from the training images, 300 of them are randomly picked as codewords of a visual dictionary. For each given path of arbitrary length, the feature vector is generated by finding an embedding of its HGDs in the dictionary. In addition to the 300 HGDs embedding, the feature vector also contains the maximum curvature along the path, which provides information about its geometry.

This approach can be used for both 2D images and 3D image stacks, since feature vectors have a fixed size, regardless of the dimensionality of the input image. This allows us, in theory at least, to apply a classifier trained on 2D images to 3D volumes. The latter would be highly beneficial, since labeling 2D images is much easier than annotating 3D stacks. However, differences in appearance and geometry of the structures may adversely affect classifier accuracy when 2D-trained ones are applied to 3D stacks, which motivates domain adaptation.

We choose images from two publicly available datasets [41] to form two separate target domains. The first one consists of 3D image stacks of Olfactory Projection Fibers (OPF) from the DIADEM challenge[42], as depicted by Fig. 1(f). The second one is made of Brightfield microscopy stacks, such as those depicted by Fig. 1(g). The latter generates a significantly harder problem, due to the irregular staining of the dendrites and axons, which produces structured noise [41].

As source domain we explore two possible choices, one that relies on 3D imagery and the other on 2D imagery, even though the target domain is 3D. The former is closer to the target domain but the latter makes sense from an operational point of view because it is far easier to extract large amounts of ground truth data semi-automatically from 2D images than from 3D ones. To highlight the power of our approach, we use 2D aerial road images as our source domain, whose appearance is significantly different from that of the dendrites and axons in the target domain.

B. Experimental Setup

As in [17], we group voxels into supervoxels to reduce training and testing time for mitochondria and synapse segmentation, which yields 15k positive and 275k negative supervoxel samples in the source domain of the Mitochondria dataset and 7k positive and 645k negative samples in the source domain of the synapse dataset. This renders 12k and 510k negative training samples in the target domain of the Mitochondria and synapse datasets respectively.

To simulate a real scenario, we create 10 different transfer learning problems using the samples from either one mitochondria or synapse at a time as positives, which translates into approximately 300 and 800 positive training supervoxels per mitochondria or synapse, respectively. We use the default parameters provided in the publicly-available code of [17]($K = 2000$). We evaluate segmentation performance using the Jaccard Index, as in [17], computed as the number of true positives over the sum of true positives, false negatives and false positives.

For path classification, 2500 positive and negative samples are extracted from each image through random sampling, as in [41]. This results in balanced sets of 30k samples for training in the roads dataset, and 20k for training and 20k for testing for OPF, and 15k in each for Brightfield. When the last two are used as the source domain, training and testing sets are merged together, yielding 40k and 30k samples respectively. To simulate the lack of training data, we randomly pick an equal number of positive and negative samples for training from the target domain.

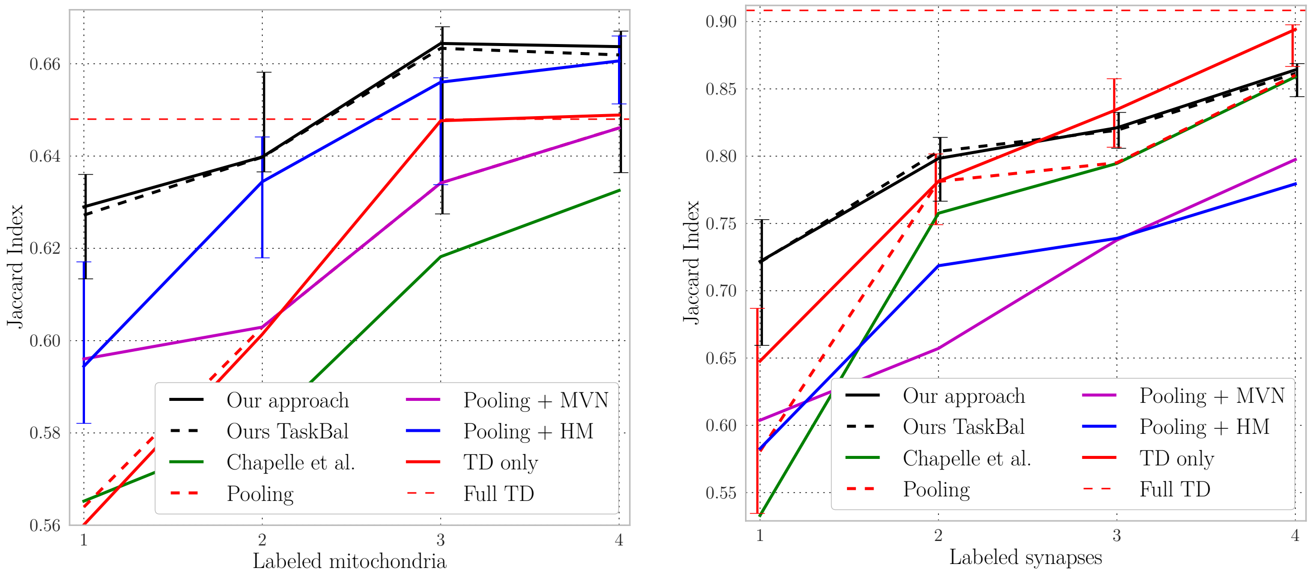
The HGD codewords are extracted from the source domain dataset, and used for both domains to generate consistent feature vectors. We employ gradient boosted trees, which in our experiments outperformed boosted stumps and kernel SVMs. For all the boosting-based baselines we set the maximum tree depth to $L = 3$, equivalent to a maximum of 8 leaves, and shrinkage $\gamma = 0.1$, as in [9]. The number of boosting iterations is set to $K = 500$. For these datasets we report the test error computed as the percentage of mis-classified examples.

For all datasets we evaluate our approach with and without class balancing. With class balancing we set $c_t = \frac{1}{N_t}$ to give both tasks equal weight, while without class balancing we set $c_t = 1$ for each task.

C. Baselines

On each dataset, we compare our approach against the following baselines: training with reference or target domain data only (shown as *SD only* and *TD only*), training a single classifier with both target and source domain data (*Pooling*), and with the multi-task approach of [8] (labeled *Chapelle et al.*). On the path classification datasets we evaluate our approach using regression-tree weak learners and therefore also compare to the tree-based adaptation (Trada) method of [18] on these datasets. We evaluate performance with varying amounts of supervision in the target domain, and also show the performance of a classifier trained with all the available labeled data, shown as *Full TD*, which represents fully supervised performance on this domain and is useful in gauging the relative performance improvement of each method. In a sense this represents the gold-standard that the best transfer learning technique could be expected to achieve.

We also compare to linear Canonical Correlation Analysis (CCA) and Kernel CCA (KCCA) [11] for learning a shared latent space on the path classification dataset, and use a Radial Basis kernel function for KCCA, which is a commonly used kernel. Its bandwidth is set to the mean distance across the training observations. Following [28], [30] we establish correspondence between domains using their binary category labels. The data size and dimensionality of the Mitochondria and synapse datasets is prohibitive for these methods, and instead we compare to Mean-Variance Normalization (MVN) and Histogram Matching (HM) that are common normalizations one might apply to compensate for acquisition artifacts. MVN normalizes each input 3D intensity patch to have a unit variance and zero-mean, useful for compensating for linear brightness and contrast changes in the image. HM applies a non-linear transformation and normalizes the intensity values



(a) Mitochondria Segmentation

(b) Synapse Segmentation

Fig. 3. *EM Segmentation:* (a) mitochondria and (b) synapses. Jaccard index measure for our method and the baselines over 10 runs on the target domain, with varying supervision. Simple Mean-Variance Normalization (MVN) and Histogram Matching (HM), although helpful, are unable to fully correct for differences between acquisitions when only afforded few labeled data. In contrast, our method yields a higher performance without the need for such priors and is able to faithfully leverage the source domain data to learn from relatively few examples in the target domain, outperforming the baseline methods. Best viewed in color.

of one data volume such that the histogram of its intensities matches the other.

D. *Results: Mitochondria and Synapse Segmentation*

The Jaccard Index on the test stacks of the EM segmentation datasets for 10 different runs is shown in Fig. 3 for our approach and the baseline methods, with varying amounts of supervision in the target domain. The performance of *SD-only* is not displayed since it performs poorly on both datasets and yields a Jaccard Index below 50%.

The results for mitochondria segmentation are displayed in Fig. 3(a). Our approach significantly outperforms Chapelle et al. and the other baselines. The next most successful method is pooling with histogram matching (HM). However, our method yields even higher performance, its accuracy being close to that of *Full TD* when using only one labeled target mitochondria. When given more labeled data, both our approach and HM yield higher performance than *TD only* and is even able to use the source domain data to improve over *Full TD*.

Similarly, the results for synapse segmentation are shown in Fig. 3(b). Each labeled synapse contains only a few supervoxels. Given such limited supervision, Chapelle et al. does not improve upon *TD-only* performance. Instead, it overfits to the source domain data. Similarly, MVN and HM normalization are unable to account for the transformation between the different data acquisitions. In contrast, our approach is able to effectively leverage the source domain data to obtain a more accurate segmentation even with only one labeled synapse in the target domain. Provided four labeled synapses it becomes difficult to improve over *TD-only* performance. However, as annotation in 3D is costly this already represents a significant labeling effort, and our approach still exhibits the best overall performance.

Qualitative segmentation results obtained with a single labeled mitochondria or synapse are also provided in Fig. 4. Compared to the baselines, the segmentations generated by our approach exhibit higher accuracy and most closely resemble the ground truth. From a practical point of view, our approach does not require parameter tuning and cross-validation is not necessary. This can be a bottleneck in some scenarios where large volumes of data are used for training. For this task, training our method took less than an hour per run, while Chapelle et al. took over 7 hours due to cross-validation.

E. *Results: Path Classification*

We first discuss using 3D imagery as both the source and target domains and then 2D imagery as the source while the target remains 3D.

a) 3D Neural Axons as the Source Domain: Fig. 5 depicts our path classification results using the 3D microscopy images from one microscopy imaging technology as the source domain, and those of the other one as the target domain. As the microscopy images from each dataset depict very different imaging modalities (see Fig. 1), this poses a challenge for transfer learning. The performance of *SD-only* and linear CCA on these datasets is above 29% and 8% respectively, and as such they are not displayed in the figure.

The results of Brightfield to OPF are shown in Fig. 5 (top). With the exception of Trada and our approach, the other baseline methods have difficulty improving over *TD-only* performance, and in fact perform worse than *TD-only*, especially when provided only a few labeled samples in the target domain. In contrast, our approach achieves a consistent

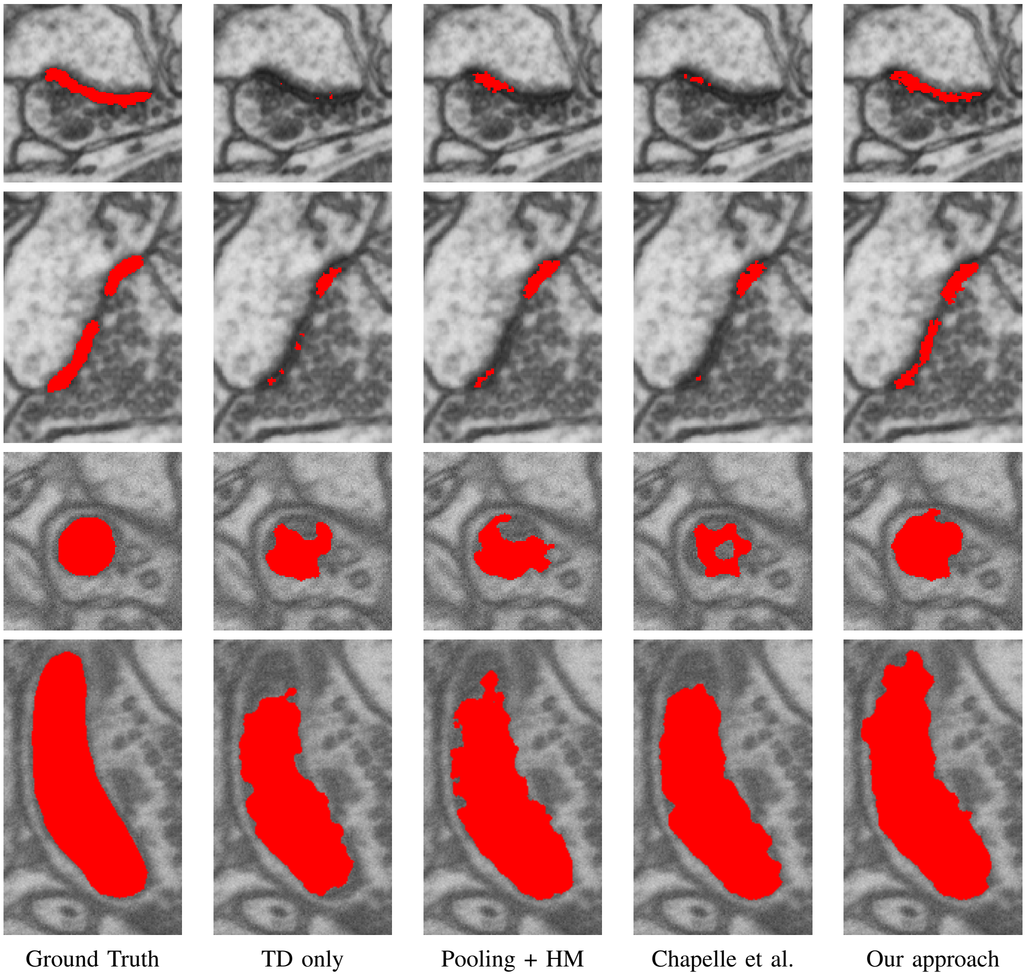


Fig. 4. Qualitative results for the segmentation datasets when using a single labeled mitochondria or synapse in the target domain. The segmentation masks output by our approach and the baselines are shown in red for two example mitochondria and synapses. The ground-truth is also shown. Compared with baselines the segmentations output by our approach exhibit a higher accuracy and most closely resemble the ground-truth. Best viewed in color.

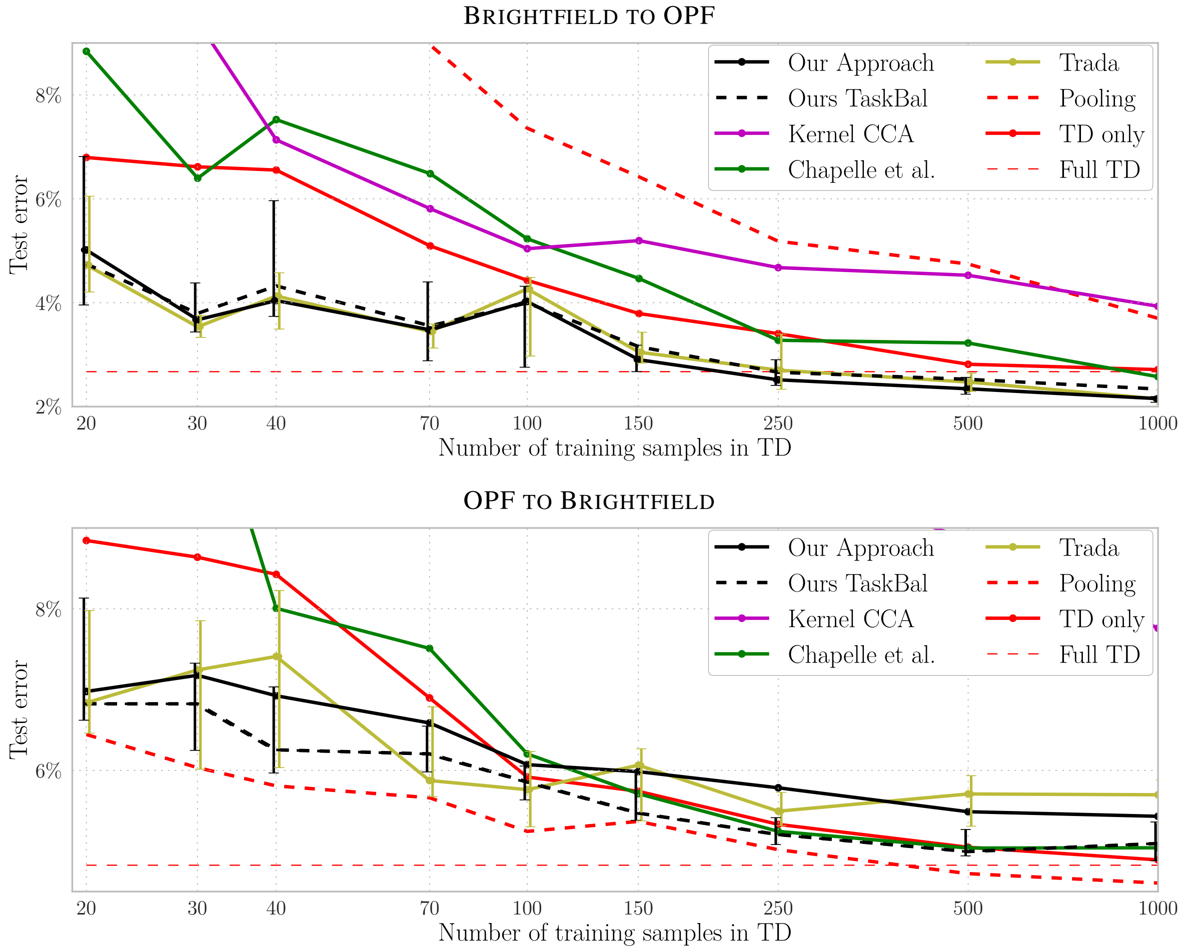
improvement over *TD-only* that is seen to be most significant when the labeled data in the target domain is scarce, which is when domain adaptation is most needed, and it is even able to improve over *Full-TD*. The performance of our approach is matched by Trada on this dataset, which is also able to achieve a significant improvement over *TD-only* and the other baselines.

Fig. 5 (bottom) displays the results for OPF to Brightfield. Our approach with task balancing achieves a significant improvement over *TD-only* when provided few target domain training samples and outperforms the baselines. Task balancing plays a more significant role for the Bightfield dataset that can be attributed to the large appearance difference between them and the rich visual cues that are present in Brightfield but absent from OPF. Unlike Brightfield to OPF, Trada is unable to match the performance of our approach when adapting OPF to Brightfield, which is likely due to its reliance on a cross-domain interpolation parameter that can be difficult to cross-validate, which is not required with our approach.

Surprisingly, naive *Pooling* achieves the best performance for OPF to Brightfield. Note, however, that while it does exceptionally well on this dataset, its preference towards Brightfield is also reflected when transferring from Brightfield to OPF where it results in the worst performance that is significantly worse than *TD-only*. In contrast, our approach is able to consistently improve over *TD-only* performance and the baselines and successfully leverage the source domain data to reduce annotation effort across both datasets.

b) 2D Aerial Roads as the Source Domain: Using the same 3D images as before as our target domain, we now switch to aerial road images such as those in the third row of Fig. 1 to provide our source domain. When compared to the 3D microscopy images, the 2D road images exhibit a much more different appearance to those of the target domain and therefore present a greater challenge.

The results on the OPF dataset are shown in Fig. 6 (top). Our approach outperforms the baselines, especially when there are few training samples in the target domain, and yields a similar performance with and without task balancing. The next best competitor is Trada, followed by Chapelle et al., although this method exhibits a much higher variance than our approach and both baselines perform poorly when only provided a few labeled target examples. This is also the case for KCCA. The results of linear CCA are not shown in the plots because it yielded very low performance compared to the other baselines, achieving a 14% error rate with 1k labeled examples and


Brightfield to OPF

OPF to Brightfield

Fig. 5. *Path Classification, 3D imagery as source domain:* Median, lower and upper quartiles of the test error as the number of training samples is varied. Our approach is able to successfully leverage the source domain data to significantly reduce annotation effort and exhibits the best overall performance across both datasets. Best viewed in color.

its performance significantly degrading with fewer training samples. Similarly, *SD only* performance is 16%.

Our approach comes close to *Full TD* when using as few as 70 training samples, even though *Full TD* was trained with 20k samples from the target domain. This highlights the ability of our method to effectively leverage the large amounts of source-domain data. As shown in Fig. 6, there is a clear tendency for all methods to converge at the value of *Full TD*, although our approach does so significantly faster. Moreover, the parameter tuning required by Chapelle et al. and Trada is done through cross-validation, which can perform poorly when only afforded a few labeled samples in the target domain, and results in longer training times. Chapelle et al. took 25 minutes to train, while our approach only took between 2 and 15 minutes, depending on the amount of labeled data.

The results on the Brightfield dataset are shown in Fig. 6 (bottom). Both linear and kernel CCA perform poorly on this dataset, and are therefore not shown in the plot. Similarly, Chapelle et al. requires a fair amount of supervision in the target domain before achieving an improvement over *SD only* performance. Trada also performs poorly on this dataset. In contrast, our approach obtains a significant improvement with as little as 30 labeled target samples, outperforming the baseline methods. For > 70 labeled target samples, although it still performs better than the other methods, our approach without task balancing performs worse than the *TD only* baseline. We believe this is because of task-specific attributes in the Brightfield dataset that are not modeled with our approach. This effect is diminished with task balancing, which assigns more emphasis to the target training samples during learning. Despite these differences, our approach is still able to more effectively leverage the source domain data to reduce the required amount of supervision in the target domain compared to the baselines.

Qualitative results are displayed for both the OPF and Brightfield datasets in Fig. 7 and 8. The false and missed detections are shown for each of the baselines and our approach. As false detections typically concentrate around overlapping subpaths on these datasets, we display a color coding that for each voxel reflects the number of false or missed detections that include it. On OPF all approaches result in only a few missed detections, however, our approach achieves a significant decrease in false detections. Compared with OPF, the Brightfield dataset contains more complicated path structures. Our approach exhibits the best performance among the baseline methods on this dataset, with the fewest overall number of false and missed detections resulting in a more accurate path reconstruction.

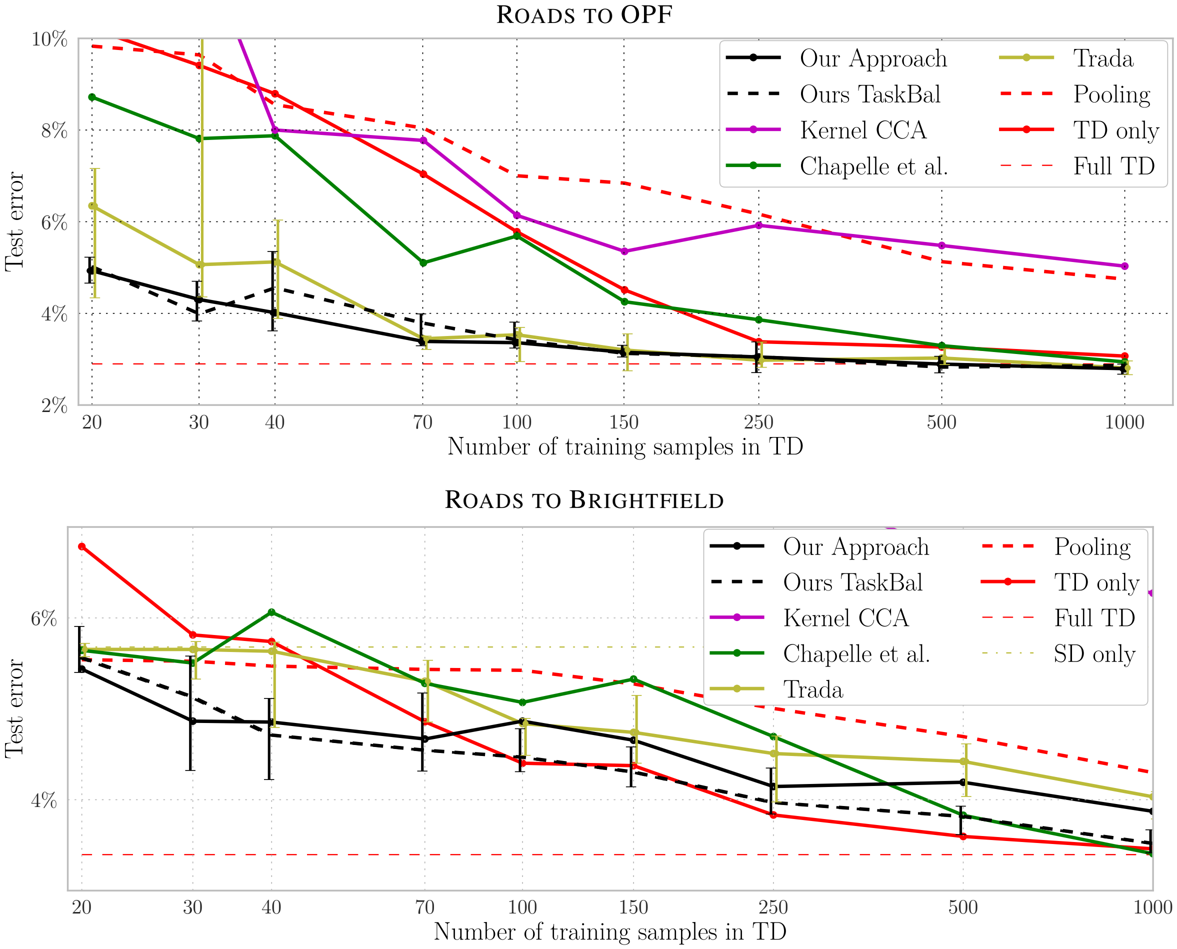


Fig. 6. *Path Classification, 2D imagery as source domain:* Median, lower and upper quartiles of the test error as the number of training samples is varied. For OPF, our approach nears *Full TD* performance with as few as 70 training samples in the target domain and significantly outperforms the baseline methods for both experiments when afforded few training samples. Best viewed in color.

F. Partial Dependence Analysis

To analyze the behavior of the classifiers learned with our approach, we use Partial Dependence Plots (PDPs) [9] to observe the classifier score as a function of the value of one specific feature, averaging out the effect of the other features. If $\mathbf{x} = (x[1], \ldots, x[M])^\intercal$ and features are indexed with $\mathcal{P} = \{1, 2, \ldots, M\}$, denote the scoring function as $f(\mathbf{x}) = f(x[n], \mathbf{x}_c)$, where $\mathbf{x}_c$ contains all features but the n^{th} one. The partial dependence of $f(\mathbf{x})$ with respect to the n^{th} feature is then computed as

$$\bar{f}_n(\lambda) = \frac{1}{|X|} \sum_{\mathbf{x} \in \mathcal{X}} f(\lambda, \mathbf{x}_c) \,, \qquad (9)$$

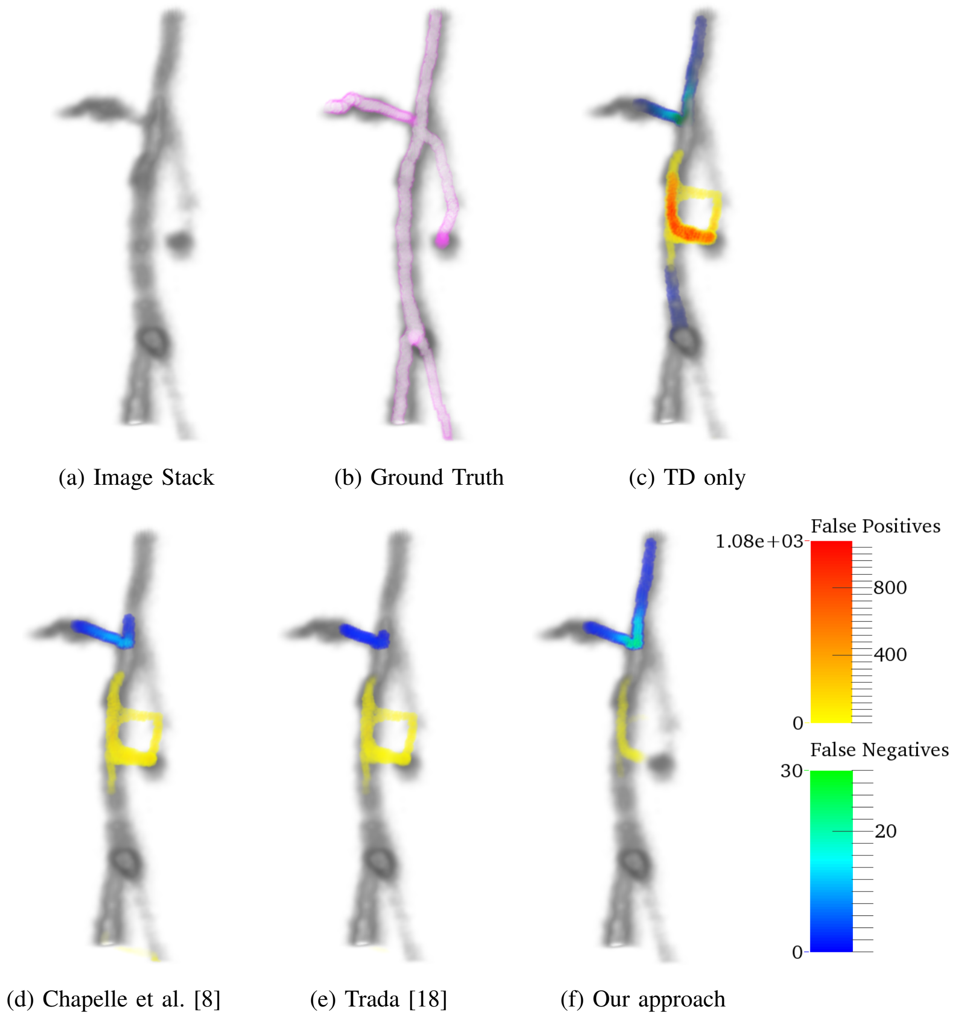
where X is the set of available training data.

We choose λ to be features with high relative importance [9] for the path classification and mitochondria segmentation datasets, and then plot the PDPs for the baselines *SD only*, *TD only*, *Full TD*, and our approach in Fig. 9. When comparing two classifiers, what matters is their behavior as a function of the feature value, i.e., the shape of their response, while the overall scaling is classifier-dependent.

For the OPF dataset, we plot the partial dependence of the feature that encodes the maximum curvature along the path. From Figs. 9(a,b) it is observed that the classifier prefers paths with a low curvature, which is a sensible choice, since the shape of tubular structures is typically smooth. For the mitochondria dataset the partial dependence of one of the structure tensor eigenvalues is displayed, which has a high value when inside a mitochondria, also reflected in Figs. 9(c,d).

In Fig. 9 the PDPs of the learned classifiers are displayed with varying amounts of supervision in the target domain. Figures 9(a,c) depict the errors that can result from overfitting when afforded only few target domain training samples (*TD only*), such as missing important features (Fig. 9(a)), indicated by its constant PDP, or learning an incorrect pattern (Fig. 9(c)). In contrast, our approach is able to leverage the source domain data to discover relevant features and prevent overfitting. Another interesting observation is the shift between the curves for *Full TD* and *SD only*, which reflects acquisition differences that are compensated by our approach.

Finally, Figs. 9(b,d) show the same plots when afforded a considerable amount of training data in the target domain. In this case, the *TD only* classifier exhibits a more similar performance to *Full TD* and is able to learn a more representative pattern. Although our approach also improves, its PDPs are fairly consistent across different amounts of supervision and



(a) Image Stack

(b) Ground Truth

(c) TD only

(d) Chapelle et al. [8]

(e) Trada [18]

(f) Our approach

Fig. 7. Qualitative results for the OPF path classification dataset. The 3D visualizations show the amount of false positive and false negative paths predicted by each approach at every location in the stack along with the ground-truth. The color coding displays the number of false or missed detections passing through each location. While all approaches result in only a few missed detections, compared with the baseline methods our approach produces significantly fewer false detections. Best viewed in color.

it is able to learn a representative pattern even with limited supervision in the target domain.

V. Conclusion

In this paper we presented an approach for performing non-linear domain adaptation with boosting. Our method learns a task-independent decision boundary in a common feature space, obtained via a non-linear mapping of the features in each task. This contrasts with recent approaches that learn task-specific boundaries and is better suited for problems in domain adaptation where each task is of the same decision problem, but whose features have undergone an unknown transformation. In this setting, we illustrated how the boosting-trick can be used to define task-specific feature mappings and effectively model non-linearity, offering distinct advantages over kernel-based approaches both in accuracy and efficiency. Our method relies on mid-level features and its effectiveness depends on the extent to which these features can be shared across the target and source domains.

We evaluated our approach on four challenging bio-medical datasets where it achieved a significant gain over using labeled data from either domain alone and outperformed recent multi-task learning methods.

References

[1] Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B.: Fiji: an open-source platform for biological-image analysis. Nature Methods **9**(7) (2012) 676–682 Code available at http://pacific.mpi-cbg.de.

[2] Morales, J., Alonso-Nanclares, L., Rodríguez, J.R., DeFelipe, J., Rodríguez, Á., Merchán-Pérez, Á.: Espina: A Tool for the Automated Segmentation and Counting of Synapses in Large Stacks of Electron Microscopy Images. Frontiers in neuroanatomy (2011)

[3] Sommer, C., Straehle, C., Koethe, U., Hamprecht, F.: ilastik: Interactive Learning and Segmentation Toolkit. In: International Symposium on Biomedical Imaging. (2011)

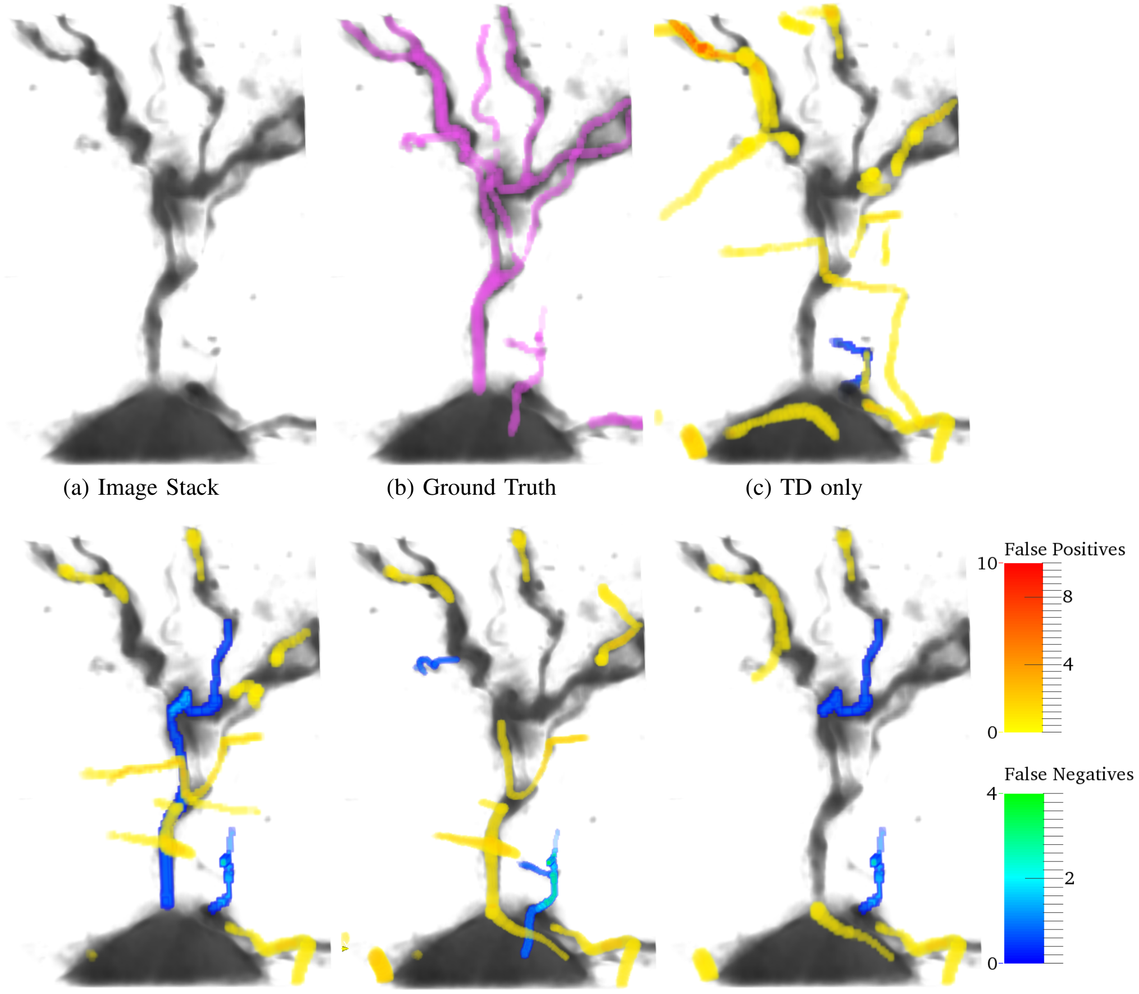
[4] Jiang, J.: A Literature Survey on Domain Adaptation of Statistical Classifiers. Technical report, University of Illinois at Urbana-Champaign (2008)

[5] Caruana, R.: Multitask Learning. Machine Learning **28** (1997)

[6] Ando, R.K., Zhang, T.: A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. Journal of Machine Learning Research **6** (2005) 1817–1853

[7] Evgeniou, T., Micchelli, C., Pontil, M.: Learning Multiple Tasks with Kernel Methods. Journal of Machine Learning Research **6** (2005)

[8] Chapelle, O., Shivaswamy, P., Vadrevu, S., Weinberger, K., Zhang, Y., Tseng, B.: Boosted Multi-Task Learning. Machine Learning (2010)



(a) Image Stack (b) Ground Truth (c) TD only

(d) Chapelle et al. [8] (e) Trada [18] (f) Our approach

Fig. 8. Qualitative results for the Brightfield path classification dataset. The 3D visualizations show the amount of false positive and false negative paths predicted by each approach at every location in the stack along with the ground-truth. The color coding displays the number of false or missed detections passing through each location. Compared with the baselines our approach results in the fewest overall number of false and missed detections yielding a more accurate path classification. Best viewed in color.

[9] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2001)

[10] Zheng, Z., Zha, H., Zhang, T., Chapelle, O., Sun, G.: A General Boosting Method and Its Application to Learning Ranking Functions for Web Search. In: Advances in Neural Information Processing Systems. (2007)

[11] Bach, F.R., Jordan, M.I.: Kernel Independent Component Analysis. Journal of Machine Learning Research **3** (2002) 1–48

[12] Ek, C.H., Torr, P.H., Lawrence, N.D.: Ambiguity Modelling in Latent Spaces. In: Machine Learning in Medical Imaging. (2008)

[13] Salzmann, M., Ek, C.H., Urtasun, R., Darrell, T.: Factorized Orthogonal Latent Spaces. In: International Conference on Artificial Intelligence and Statistics. (2010)

[14] Memisevic, R., Sigal, L., Fleet, D.J.: Shared Kernel Information Embedding for Discriminative Inference. IEEE Transactions on Pattern Analysis and Machine Intelligence (April 2012) 778–790

[15] Tu, Z., Zheng, S., Yuille, A.L., Reiss, A.L., Dutton, R.A., Lee, A.D., Galaburda, A.M., Dinov, I., Thompson, P.M., Toga, A.W.: Automated Extraction of the Cortical Sulci Based on a Supervised Learning Approach. IEEE Transactions on Medical Imaging **26**(4) (2007) 541–552

[16] Lindner, C., Thiagarajah, S., Wilkinson, J., Loughlin, J., Wallis, G., Cootes, T.: Fully Automatic Segmentation of the Proximal Femur Using Random Forest Regression Voting. IEEE Transactions on Medical Imaging (2013)

[17] Becker, C., Ali, K., Knott, G., Fua, P.: Learning Context Cues for Synapse Segmentation. IEEE Transactions on Medical Imaging **32**(10) (October 2013) 1864–1877

[18] Chen, K., Bai, J., Zheng, Z.: Ranking Function Adaptation with Boosting Trees. ACM Transactions on Information Systems (TOIS) (2011)

[19] Becker, C., Christoudias, M., Fua, P.: Non-Linear Domain Adaptation with Boosting. In: Advances in Neural Information Processing Systems. (2013)

[20] Bi, J., Xiong, T., Yu, S., Dundar, M., Rao, R.B.: An Improved Multi-task Learning Approach with Applications in Medical Diagnosis. In: Machine Learning and Knowledge Discovery in Databases. (2008)

[21] van Opbroek, A., Ikram, M.A., Vernooij, M.W., de Bruijne, M.: A Transfer-Learning Approach to Image Segmentation Across Scanners by Maximizing Distribution Similarity. In: MLMI. (2013)

[22] Wang, B., Prastawa, M., Saha, A., Awate, S.P., Irimia, A., Chambers, M.C., Vespa, P.M., Van Horn, J.D., Pascucci, V., Gerig, G.: Modeling 4D Changes in Pathological Anatomy Using Domain Adaptation: Analysis of TBI Imaging Using a Tumor Database. In: Multimodal Brain Image Analysis. (2013) 31–39

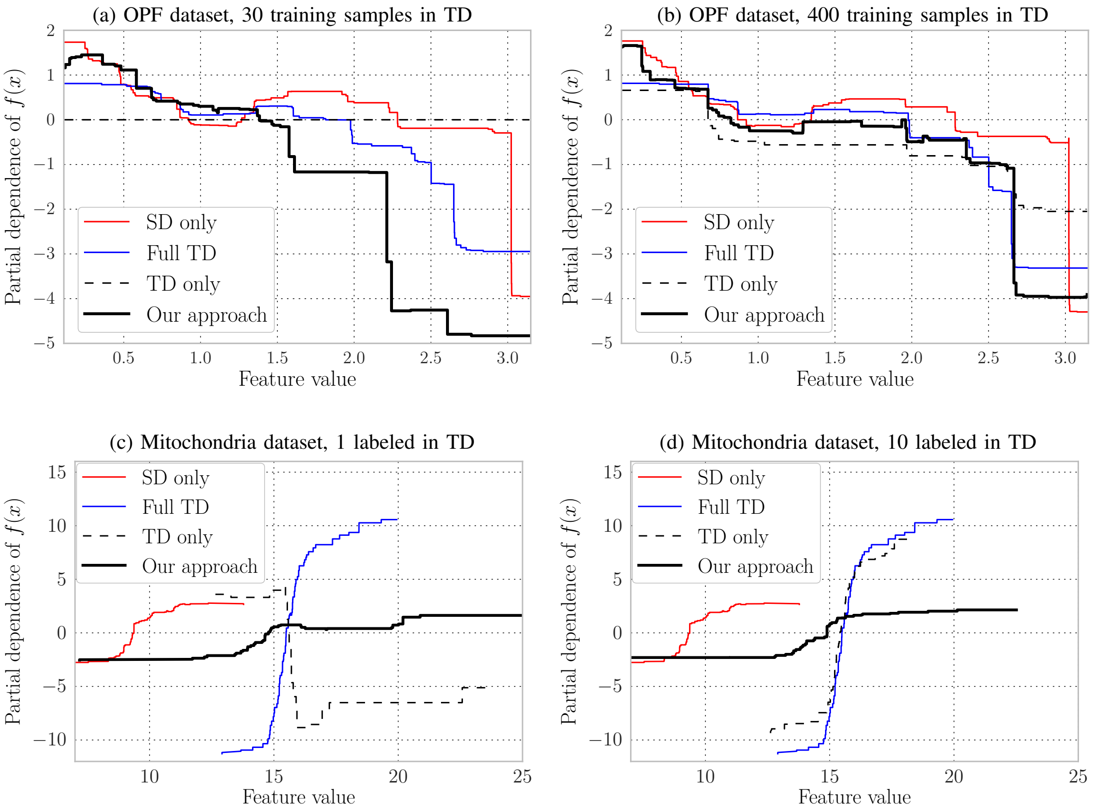
[23] Baxter, J.: A Model of Inductive Bias Learning. Journal of Artificial Intelligence Research (2000)

[24] Daumé, H.: Bayesian Multitask Learning with Latent Hierarchies. In: Uncertainty in Artificial Intelligence. (2009)

[25] Kumar, A., Daumé, H.: Learning Task Grouping and Overlap in Multi-Task Learning. In: International Conference on Machine Learning. (2012)

[26] Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research **8** (2007)

[27] Jacob, L., Bach, F., Vert, J.P.: Clustered Multi-Task Learning: A Convex



(a) OPF dataset, 30 training samples in TD

(b) OPF dataset, 400 training samples in TD

(c) Mitochondria dataset, 1 labeled in TD

(d) Mitochondria dataset, 10 labeled in TD

Fig. 9. Analysis of the behavior of the trained classifiers through partial dependence plots for the OPF (top) and Mitochondria segmentation (bottom) datasets, with different amounts of training data in the Target Domain. Best viewed in color.

Formulation. In: Advances in Neural Information Processing Systems. (2008)

[28] Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting Visual Category Models to New Domains. In: European Conference on Computer Vision. (2010)

[29] Shon, A.P., Grochow, K., Hertzmann, A., Rao, R.P.N.: Learning Shared Latent Structure for Image Synthesis and Robotic Imitation. In: Advances in Neural Information Processing Systems. (2006) 1233–1240

[30] Kulis, B., Saenko, K., Darrell, T.: What You Saw is Not What You Get: Domain Adaptation Using Asymmetric Kernel Transforms. In: Conference on Computer Vision and Pattern Recognition. (2011)

[31] Zhang, K., Zheng, V.W., Wang, Q., Kwok, J.T., Yang, Q., Marsic, I.: Covariate Shift in Hilbert Space: A Solution via Surrogate Kernels. In: International Conference on Machine Learning. (2013)

[32] Gopalan, R., Li, R., Chellappa, R.: Domain Adaptation for Object Recognition: An Unsupervised Approach. In: International Conference on Computer Vision. (2011)

[33] Dai, W., Yang, Q., Xue, G., Yu, Y.: Boosting for Transfer Learning. In: Machine Learning. (2007) 193–200

[34] Heimann, T., Mountney, P., John, M., Ionasec, R.: Learning without Labeling: Domain Adaptation for Ultrasound Transducer Localization. In: Conference on Medical Image Computing and Computer Assisted Intervention. (2013) 49–56

[35] Jie, B., Zhang, D., Cheng, B., Shen, D.: Manifold Regularized Multi-Task Feature Selection for Multi-Modality Classification in Alzheimer's Disease. In: Conference on Medical Image Computing and Computer Assisted Intervention. (2013)

[36] Rosset, S., Zhu, J., Hastie, T.: Boosting as a Regularized Path to a Maximum Margin Classifier. Journal of Machine Learning Research (2004)

[37] Caruana, R., Niculescu-mizil, A.: An Empirical Comparison of Supervised Learning Algorithms. In: International Conference on Machine Learning. (2006)

[38] Viola, P., Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. In: Conference on Computer Vision and Pattern Recognition. (2001)

[39] Ali, K., Fleuret, F., Hasler, D., Fua, P.: A Real-Time Deformable Detector. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(2) (February 2012) 225–239

[40] Freund, Y., Schapire, R.: A Short Introduction to Boosting (1999) Journal of Japanese Society for Artificial Intelligence, 14(5):771-780.

[41] Turetken, E., Benmansour, F., Fua, P.: Automated Reconstruction of Tree Structures Using Path Classifiers and Mixed Integer Programming. In: Conference on Computer Vision and Pattern Recognition. (June 2012)

[42] Ascoli, G., Svoboda, K., Liu, Y.: Digital Reconstruction of Axonal and Dendritic Morphology DIADEM Challenge (2010)