Domain Adaptation for Small Datasets: Applications in Automated In Vivo Oral Cancer Diagnosis

Kayla Caughlin[1], Elvis Duran-Sierra[3], Shuna Cheng[3], Rodrigo Cuenca[2], Beena Ahmed[4], Jim Ji[5], Mathias Martinez[6], Moustafa Al-Khalil[6], Hussain Al-Enazi[7], Yi-Shing Lisa Cheng[8], John Wright[8], Javier A. Jo[2], and Carlos Busso[1]

***Abstract*— Deep learning approaches for medical image analysis are limited by small data set size due to multiple factors such as patient privacy and difficulties in obtaining expert labelling for each image. In medical imaging system development pipelines, phases for system development and classification algorithms often overlap with data collection, creating small disjoint data sets collected at numerous locations with differing protocols. In this setting, merging data from different data collection centers increases the amount of training data. However, a direct combination of datasets will likely fail due to domain shifts between imaging centers.**

In contrast to previous approaches that focus on a single data set, we add a domain adaptation module to a neural network model and train using multiple data sets. Our approach encourages domain invariance between two *multispectral autofluourescence imaging* (maFLIM) data sets of in vivo oral lesions collected with an imaging system currently in development. The two data sets have differences in the sub-populations imaged and in the calibration procedures used during data collection. We mitigate these differences using a gradient reversal layer and domain classifier. Our final model trained with two data sets increases performance on the source data set by 2.16% over the network without domain adaptation. We also achieve a significant increase in average performance over the best baseline model train with two domains ($p = 0.0341$). Furthermore, our model achieves a 0.69% increase in performance over the best model trained on the source data set alone. Our approach lays the foundation for faster development of computer aided diagnostic systems and presents a feasible approach for creating a single classifier that robustly diagnoses images from multiple data centers in the presence of domain shifts.

***Index Terms*— Automated oral cancer diagnosis, domain adaptation, gradient reversal, multispectral autofluorescence imaging, variance regularization**

I. Introduction

Traditionally, a large data set is required for training a successful deep learning model. Unfortunately, medical imaging data sets are typically small and heterogeneous, creating issues with overfitting and exhibiting lack of generalization to different domains and settings across data centers. Many strategies have avoided deep learning solutions altogether, instead, focusing on methods such as *support vector machines* (SVM) and *quadratic discriminant analysis* (QDA) that offer less flexibility, but are less prone to overfitting [1]–[3]. Other methods reduce problems associated with inter-patient variability by including images from each patient in the training set [4]. However, requiring an image from each patient in the training set is impractical for clinical translation, because the model needs to be re-trained with the addition of every new patient. Another alternative approach for modalities with some similarities to natural images (e.g., MRI and digital histology) is to adapt deep learning models pre-trained on large-scale data sets such as ImageNet [5], [6]. However, other medical imaging applications are highly specific and do not have a similar modality that can be pre-trained on an extensive data set (i.e., in vivo fluorescent lifetime images of oral lesions).

In the absence of a pre-trained model and to avoid using images from each test patient in the training set, the combination of small medical image data sets from different centers, including sets from slightly different domains, can increase the size of the data set, and, potentially, mitigate generalization problems. However, domain shifts between data centers caused by differences in imaging systems, data collection protocols, and sub-populations prevent direct combination [7]. In addition, mismatches between data sets from different imaging centers may occur when ground truth annotations contain ambiguity. Specifically in oral cancer identification, histology ground truth labels contain both inter- and intra- observer variability. Due to center-specific variations, generalization problems persist even when a large data set has been collected at a single imaging location [8]. Classification challenges due to domain shift have been documented in multiple imaging modalities, including *computed tomography* (CT), *magnetic resonance imaging* (MRI), ultrasound, and *immunohistochemistry* (IHC) [8]–[12].

Previous studies on oral cancer classification using *multispectral autofluourescence imaging* (maFLIM) focused only on a single, small data set [1]–[3], [13], [14]. As a single large-scale maFLIM oral lesion data set is not currently accessible, we would like to combine images from various collection centers. Specifically, we aim to improve performance on a source dataset through the inclusion of auxiliary training data from a separate imaging center. To mitigate domain shift problems, we propose the use of domain adaptation techniques in a deep learning framework to merge the data sets and create domain invariance feature representations. While domain adaptation techniques have been widely investigated for combination of domains in other applications, little work has been reported on domain adaptation in cancer diagnosis of oral lesions from in vivo maFLIM. Although we focus on source data set performance, this work sets the foundations for a single, robust classifier that can be used for the diagnosis of oral lesions across multiple imaging systems and locations. The main contributions of this work are as follows:

- We quantify source performance decrease when training with multiple domains for automated oral cancer diagnosis from maFLIM.
- We propose a new domain adaptation model to decrease the domain shift between data collection centers for maFLIM.
- We investigate failure modes of standard gradient reversal for

*This work was supported by NIH, grant R01:5R01CA218739-04

[1]Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX, USA
[2]School of Electrical and Computer Engineering, University of Oklahoma, Norman, OK, USA
[3]Department of Biomedical Engineering, Texas A&M University, College Station, TX, USA
[4]School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia
[5]Department of Electrical and Computer Engineering, Texas A&M University at Qatar, Doha, Qatar
[6]Department of Cranio-Maxillofacial Surgery, Hamad Medical Corporation, Doha, Qatar
[7]Department of Otorhinolaryngology Head and Neck Surgery, Hamad Medical Corporation, Doha, Qatar
[8]College of Dentistry, Texas A&M University, Dallas, TX, USA

domain adaptation in a small data setting.
- We propose a variance regularization technique to reduce mode collapse in gradient reversal.
- We explore model selection techniques to further improve source domain performance.

This paper is organized as follows. Section II summarizes related studies in maFLIM cancer diagnosis and describes relevant background for domain adaptation. Section III details our proposed approach. Section IV presents the results of our method in comparison to the baselines and analyzes each component of our model. Section V summarizes the main contributions and results of this study.

II. Related Work

A. In Vivo maFLIM Cancer Classification Methods

Although domain adaptation methods are commonly used in other deep learning applications, approaches for cancer diagnosis and margin delineation from in vivo maFLIM have typically trained using a single domain [1]–[3], [13]–[15]. For example, Jo et al. [2] presented a solution based on a *quadratic discriminant analysis* (QDA) classifier for oral cancer diagnosis using a single data set. The inputs to the QDA classifier were hand-derived features based on signal intensity (spectral) and lifetime (temporal) features [2]. Marsden et al. [1] also used lifetime and intensity features to train a *support vector machine* (SVM) and a *random forest* (RF) classifier for margin detection in oral lesions from one imaging center. Chen et al. [16] used a SVM, but for microscopy images from a single collection site. Vasanthakumari et al. [3] used QDA and *linear discriminant analysis* (LDA) classifiers with hand-crafted features from phasor plots to classify skin lesions. Duran et al. [15] trained LDA, QDA, SVM and logistic regression classifiers on data from one imaging center and tests on data from another imaging center. However, the authors did not explore if an increase in performance might be obtained by using domain adaptation techniques. In addition, the authors report performance on the margin delineation task, while we focus on cancer diagnosis. Studies have only recently started to use deep learning solutions [1], [13]. Marsden et al. [1] explored the use of a *convolutional neural network* (CNN) for oral lesion margin delineation. However, the CNN-based model failed to outperform baselines implemented with traditional machine learning classifiers trained with hand-derived features [1]. In contrast, we previously reported a joint autoencoder and classifier structure (see Fig. 1) for oral cancer diagnosis using data-driven features that showed improved performance over the traditional baseline implemented with SVM in a single-domain setting [13].

While our previous work focused on automated oral cancer diagnosis using a single data set, our deep learning structure can accommodate domain adaptation techniques that have been successfully used in other domains. As our source data set is small, we wish to improve performance on the source data set by including training data from another data set. Unfortunately, domain shifts between data sets can cause the model capacity to split between the domains, decreasing performance on the source data set. We hypothesize that the addition of domain adaptation methods to our joint autoencoder and classifier structure can create invariant representations such that auxiliary data from a different domain can act as augmented data for the source domain. In our method, we use an auxiliary data set to increase the training data set size, formulating the classification performance on the source data as our primary evaluation criterion. This formulation is radically different from other approaches used for in vivo oral cancer diagnosis from maFLIM.

B. Domain Adaptation Background

The goal of domain adaptation is to mitigate the mismatch between domains. Multiple domain adaptation techniques have been proposed, including using fine-tuning [10], *generative adversarial network* (GAN) [12], gradient reversal [17], transformation of data between domains [11], and augmentation methods [18].

Several studies have used a fine-tuning, transfer learning approach when at least some labels are present for both source and target data sets [5], [6], [10]. In these methods, a base classifier was trained using a source data set. Then, the model was minimally adjusted using the labeled target data to maximize performance on the target data. Fine-tuning differs from our objectives, as we wish to maximize performance on the source data set, using a second domain to augment the training set.

In contrast to fine-tuning, Zhang et al. [9] introduced a series of source-only data augmentation methods for MRI and ultrasound that minimized the performance gap between the source and target data in both small and large data settings. Though not in the medical imaging domain, Anaby et al. [19] similarly used a deep learning method to create synthetic examples for data augmentation in a small data setting. Our work takes inspiration from data augmentation and domain adaptation.

Ganin and Lepinsky [17] proposed the gradient reversal layer, which describes a popular plug-in general purpose technique for unsupervised domain adaptation. The model consisted of three blocks: a feature extractor, a domain classifier, and a task classifier. The domain classifier was trained to generate gradient updates in the direction of maximum domain separation. The gradients resulting from the domain classifier were then reversed during back-propagation to the feature extractor. Following training, the domain classifier was discarded and the feature extractor and task classifier were used for inference [17]. While Ganin et al. [17] focused on improving generalization to a new domain, our focus is on improving the source domain performance.

While the gradient reversal layer has been used in a variety of domain adaptation applications in both unsupervised and semi-supervised settings, some authors have noted issues retaining class-discriminative information [20], [21]. Specifically, Li et al. [20] asserted that gradient reversal does not ensure that class-discriminative information is preserved (referred to as *mode collapse*). Li et al. [20] used *joint adversarial domain adaptation* (JADA) to train two task classifiers using unlabeled target samples and labeled source samples. The model finds samples on the class boundary and optimizes the feature extractor to minimize differences in the predictions of the two classifiers. The authors hypothesize that since the two task classifiers are randomly initialized, if the two task classifiers disagree on the label of the target sample, it is likely near the decision boundary of the source. By training the feature extractor to minimize the discrepancy between the two task classifiers on the target data, the trained target representation will be near the source classes, improving task discrimination on the target data. Overall, the two task classifiers encouraged distinct class boundaries, while the domain classifier encouraged domain invariance [20]. Similarly, Kurmi et al. [21] mitigated mode collapse in domain adaptation by using an approach called *informative domain discriminator for domain adaptation* (IDDA). IDDA used a modified domain classifier and selection of source samples to train the domain classifier. The modified domain discriminator classified each sample as belonging to one of the source domain class labels or to a general label for the target domain [21]. Furthermore, source samples that were misclassified in the task classifier were discarded from training the domain classifier [21]. However, the feature extractor only tried to

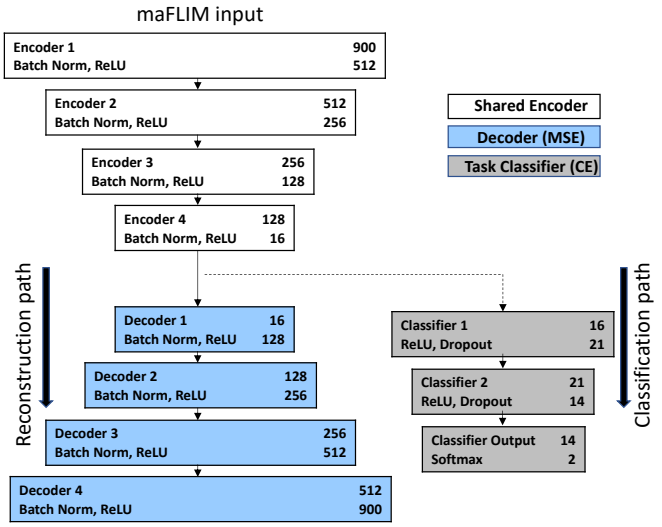


Fig. 1: Joint autoencoder and classifier neural network as presented in Caughlin et al. [13]. Our base model uses the same general structure of this neural network.

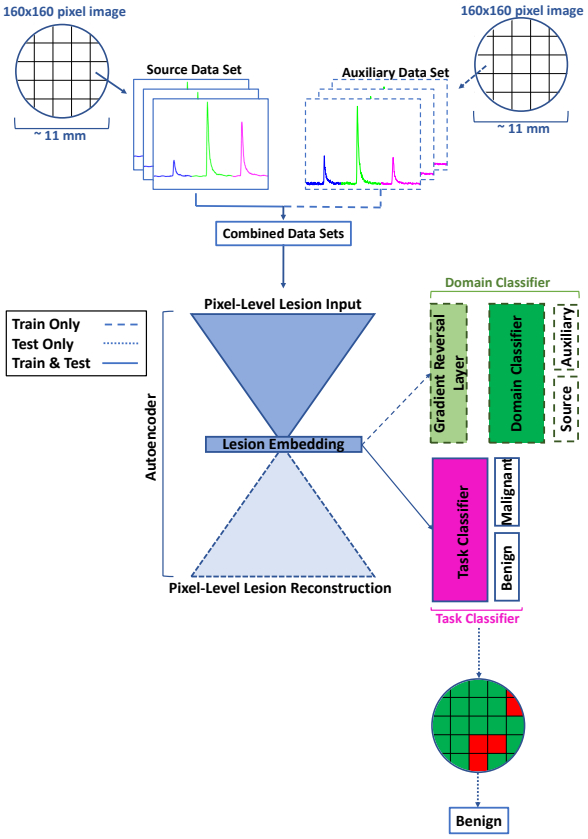


Fig. 2: Proposed architecture with autoencoder, task classifier and domain classifier implemented with gradient reversal layer. Two data sets from separate imaging centers are preprocessed to minimize center-specific differences and combined into a single data set before training the neural network with domain adaptation on a pixel level. At test time, the task classifier pixel-level diagnosis is aggregated for each image with a 50% threshold.

make the domain classifier place the target sample in any of the source classification label bins, disregarding the target label class.

While JADA and IDDA mitigated mode collapse in the unsupervised domain adaptation framework, our framework is different. In our approach, rather than having a single moderate-to-large labeled source data set and an unlabeled target data set, both of our data sets contain labels. In our setting, task supervision helps prevent loss of discriminative information during domain adaptation. The stability problem in our formulation concerns the domain classifier and it is addressed with a variance regularization that is simpler than both JADA and IDDA approaches. In contrast to JADA, our variance regularization method does not rely on randomization in task classifier initialization to prevent mode collapse. Our work details a new method for reducing mode collapse in the domain discriminator that improves domain adaptation in a small data setting.

III. Methods

A. Motivation

Our deep learning framework builds on our previously reported neural network structure [13], shown in Figure 1. Our original framework uses two blocks, an autoencoder and a classifier, that are simultaneously trained on a single data set. The autoencoder block provides regularization by reconstructing the input signal, but cannot ensure a task-discriminative bottleneck representation. Adding the task classifier provides supervision, ensuring the autoencoder generates task-discriminative representations in the bottleneck. While our original framework performed well when trained on a single data set, we observed a decrease in performance when training with multiple data sets. We extend our original framework to work in a multi-center setting by adding a domain adaptation. This approach aims to create a feature representation that cannot distinguish between data collected from different imaging centers.

B. Domain Adaptation using Gradient Reversal Layer

Figure 2 describes our proposed approach, which combines the autoencoder-classifier framework with a domain classifier with gradient reversal. The model works at the pixel level, providing a prediction for each pixel. Our architecture has three components: a feature extractor built with an autoencoder, a task classifier, and a domain classifier. In our architecture, the contracting path of the autoencoder, from the input to the bottleneck layer, plays the role of the feature extractor. Our domain classifier is trained to recognize center-specific differences in order to distinguish between the data sets (i.e., center one versus center two). The task classifier labels each pixel with a diagnosis (i.e., it classifies each pixel as benign or malignant). While margin delineation (classification of a pixel as a lesion or healthy) is another common task in oral maFLIM analysis, we only consider the diagnosis task in this work. The feature extractor is connected to the task and domain classifiers. The network is trained with an adversarial loss to maximize the performance of the task classifier while minimizing the performance of the domain classifier. We find a saddle point where the parameters of the feature extractor minimize the domain classifier performance. The key step in this formulation to minimize the performance of the domain classifier is to reverse the sign of the gradient updates generated by the domain classifier. When the performance of the domain classifier is at random level, the bottleneck layer generates feature representations that are indistinguishable between both domains, compensating for potential differences. Our network looks for the parameters at the saddle point:

$$(\hat{\theta}_{enc}, \hat{\theta}_t) = \text{argmin}\, \mathcal{L}_{total}(\theta_{enc}, \theta_t, \hat{\theta}_d) \quad (1)$$

$$\hat{\theta}_d = \text{argmax}\, \mathcal{L}_{total}(\hat{\theta}_{enc}, \hat{\theta}_t, \theta_d) \quad (2)$$

where $\hat{\theta}_{enc}$, $\hat{\theta}_t$, and $\hat{\theta}_d$ refer to the optimal parameters for the encoder, task classifier, and domain classifier, respectively. Equation 1 states that we find the parameters for the encoder and the task classifiers that minimize the task classifier loss, while Equation 2 states that we want the parameters for the domain classifier that minimizes the domain classifier loss.

The trade-off between the domain classifier and the task classifier is controlled by the hyperparameter λ. The total loss ($\mathcal{L}_{total}$) is given by Equation 3,

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \mathcal{L}_{AE} + \mathcal{L}_{VR} - \lambda \mathcal{L}_{domain} \quad (3)$$

where $\mathcal{L}_{task}$ is the cross entropy loss for the task classifier, $\mathcal{L}_{AE}$ is the reconstruction loss for the autoencoder, $\mathcal{L}_{VR}$ is a variance regularization penalty discussed in Section III-C, and $\mathcal{L}_{domain}$ is the domain classifier loss. The negative sign on the term $\lambda \mathcal{L}_{domain}$ reflects the reversal of the gradient for the domain classifier. As in Ganin and Lepinsky [17], the hyperparameter lambda gradually increases during training, allowing the domain discriminator to distinguish the classes before applying large updates to merge the domains.

Since the relative strength of the domain adaptation increases during training as lambda increases, the model takes some time to achieve domain invariance. Ideally, the domain classifier should approach 50% accuracy on both data sets. This scenario indicates that the domain classifier can no longer tell the difference between the domains (the domain classifier has converged). If we choose the best model before the domain classifier convergences, good performance on only one domain may dominate the task classifier loss, while the other domain may have poor classification performance. Thus, we select the best model only after the development set domain performance falls near 50% for both the source and auxiliary data. Then, we select the best model as determined by the task classifier performance on the development set.

After training, the network predicts a label for each pixel in the test images. The predicted diagnosis for an image is generated by aggregating the pixel-level predictions using a 50% threshold. The long dashed borders and arrows in Figure 2 indicate that the auxiliary data is only used during training. Similarly, the decoder and the domain classifier are discarded after training. The small dashed arrows indicate that the image-level aggregation only applies during testing.

C. Strategies to Increase Regularization

Since we are dealing with small data sets, we implement several approaches to increase the regularization of the architecture, improving the generalization of the model to avoid overfitting. Using an autoencoder is our first regularization approach. The autoencoder includes an unsupervised loss to reconstruct the input features. While we do not have many images, we do have over 2.5 million pixels to train this model.

Our second approach to increase regularization is the use of dropout in the classifier layers. Dropout randomly removes nodes during training, resulting in a model that approximates an average over a collection of sub-models [22]. Dropout discourages the model from relying on specific combinations of nodes (referred to as "co-adaptation") and reduces overfitting [22]. Each layer in the encoder and decoder used batch normalization. Batch normalization was introduced by Ioffe and Szegedy [23] to minimize internal covariate shift and was placed before the activation layer. However, experiments by Santurkar et al. [24] show that internal covariate shift is not always reduced by batch normalization. Instead, the likely performance improvements may be due to a smoother loss surface during the optimization [24]. The results of batch normalization can also regularize the network by encouraging favorable model initialization and increasing generalization [24]. While Ioffe and Szegedy [23] asserted that batch normalization can reduce the need for dropout, Chen et al. [25] found improvements in combining batch normalization with dropout and using these layers after the activation. In our framework we do not use batch normalization and dropout in the same layer. Instead, we use batch normalization in the autoencoder layers and dropout in the task classifier layers. However, we acknowledge that performance increases may be obtained by exploring alternatives such as the combination of dropout and batch normalization in the same layer.

A technical contribution in this study is the use of variance regularization for adversarial domain adaptation. Although we experimented with multiple schedules for the hyperparameter λ, we were unable to stabilize the domain classifier with hyperparameter tuning alone. The domain classifier was unstable, reaching the expected 50% accuracy by collapsing to the predictions of a single domain and switching between which domain was predicted. During preliminary experiments, we found switching was reduced by changing the activation function to a sigmoid at the bottleneck layer (see original activations in Figure 1). However, the sigmoid activation did not entirely stabilize the training of the model, with the domain classifier still frequently collapsing. To reduce domain classifier mode collapse, we implement the variance regularization proposed by Chong et al. [26]. Equation 4 shows the variance regularization that we used.

$$\mathcal{L}_{\mathcal{VR}} = \max\{0, \gamma - \frac{1}{p(n-1)} \sum_{q=1}^{p} \sum_{i=1}^{n} (\phi_{i,q} - \frac{1}{n} \sum_{k=1}^{n} \phi_{k,q})^2\} \quad (4)$$

where $\mathcal{L}_{\mathcal{VR}}$ is the variance regularization penalty, γ is the threshold, p is the size of the bottleneck (16 in our implementation), n is the number of samples in the mini-batch (256 in our implementation), and ϕ is the value of the feature. The only change from the approach presented by Chong et al. [26] is that our threshold γ is a single value that does not change as the training progresses. We apply the regularization to the bottleneck layer, adding a penalty to the loss when the variance within a mini-batch falls below a threshold. With this approach, the input to the domain classifier is more varied and reduces the likelihood that the domain classifier collapses to predict every sample belonging to one domain.

D. Imaging System

The basis for fluorescence imaging of oral cancer is that changes in the levels of endogenous fluorophores may reflect changes in tissue structure and metabolism. Collagen, reduced *nicotinamide adenine dinucleotide* (NADH), and *flavin adenine dinucleotide* (FAD) are endogenous fluorophores of interest in cancer diagnosis and margin delineation [1]–[3], [27]. All images ($\sim$11 mm circular field of view, $\sim$100 um lateral resolution) were acquired using the maFLIM system described in Cheng et al. [27]. Figure 3 shows the endoscope. Using this system, the tissue autofluorescence was excited at 355 nm with a pulse width of 1ns. During the acquisition time, which was less than 3s, 2.8 mJ were deposited to the tissue (*maximum permissible exposure* (MPE) = 29.8 mJ [28]). The emission bands imaged collagen (390$\pm$20 nm), NADH (452$\pm$22.5 nm), and FAD ($>$500nm). Following imaging of the biopsy region, an incisional tissue biopsy was performed following standard clinical protocols.

Fig. 3: maFLIM Endoscope.

The histopathological diagnosis for each lesion biopsy was used as the ground truth for training and evaluating the proposed classifier. This process results in an image where each pixel contains a three-channel fluorescence decay. Figure 4 shows an example of this three-channel fluorescence decay. As shown in the figure, the raw images provide no direct visual information on cancer diagnosis or lesion appearance. Feature representations need be estimated from the data.

E. Data: Source and Auxiliary sets

Data was collected from two imaging centers. Each imaging center used two separate prototype versions of the maFLIM endoscope system previously reported in Cheng et al. [27]. At both centers, each patient in the study had a clinically suspicious oral lesion. Each lesion was imaged in vivo with the maFLIM endoscope, followed by surgical resection and histopathological diagnosis. All images contained 160x160 pixels, with three channel decays per pixel corresponding to collagen, reduced NADH, and FAD, respectively. In both data sets, we use the raw signal instead of iterative deconvolution to avoid signal approximations. We rely on our neural network to adjust for differences in the impulse response of the imaging systems.

The source data set contains 67 oral lesions from nine anatomical locations, as shown in Table I. The data was collected in Doha, Qatar. The *institutional review board* (IRB) for the source data collection and processing was approved by the Hamad Medical Corporation in Doha, Qatar. The histopathological diagnosis revealed that 33 lesions were benign, five lesions were dysplasia, and 29 lesions were *squamous cell carcinoma* (SCC).

The full auxiliary data set contained 84 oral lesions, including 23 cancerous/precancerous lesions and 61 benign lesions. The data was collected in Dallas, USA. To avoid introducing highly imbalanced data, we used only 23 of the 61 benign lesions from the auxiliary data set. Therefore, the auxiliary data used in this study contains 46 oral lesions from 6 anatomical locations (see Table II). IRB approval

TABLE I: Anatomical distribution of the lesions in the source data set collected in Doha, Qatar.

Location	Benign	Dysplasia	SCC
Mucosa	10	3	9
Floor of Mouth	2	0	1
Gingiva	0	2	3
Lip	10	0	2
Mandible	0	0	1
Palate	1	0	0
Maxilla	0	0	1
Retromolar	1	0	0
Tongue	9	0	12
Total	33	5	29

TABLE II: Anatomical distribution of the lesions in the auxiliary data set collected in Dallas, USA.

Location	Benign	Dysplasia	SCC
Mucosa	8	1	2
Floor of Mouth	1	0	0
Gingiva	5	1	5
Mandible	1	0	1
Palate	1	0	0
Tongue	7	7	6
Total	23	9	14

for the auxiliary data set experimental protocol was obtained from Texas A&M University. Notably, the auxiliary data set contains a larger portion of dysplasia cases compared to the source data set distribution. Dysplasia cases from both data sets were considered as malignant during classifier training.

In addition to different anatomical locations and dysplasia distributions, the experimental protocol was not consistent between imaging centers. Specifically, the source data was collected after calibrating the imaging system before each image acquisition. In contrast, the auxiliary data set was initially calibrated, but not subsequently adjusted throughout the experiment. The lack of precise calibration factors corrupts the relative intensity values between channels, especially without patient normalization. While an image from the normal contralateral side was collected for each patient, this study only uses the lesion information, resulting in greater challenges due to calibration differences.

F. Implementation

Table III shows the sizes and activations of the layers for the autoencoder and classifier. Images from the separate data sets are separately preprocessed to reduce correctable variations between data sets. We describe the preprocessing steps in Section III-G. After following preprocessing of each pixel, the data sets are combined and the classifier is trained on a pixel level.

Our training auxiliary data set is smaller than that of the source data set (Sec. III-E). Therefore, we use sample weighing on the domain classifier output to prevent the model from collapsing to only predicting the domain with more samples. The values for the sample weights are given by the *sci-kit learn* toolkit based on the training domain distribution as detailed in Pedregosa et al. [29]. Similarly, we reduce class bias by using the same strategy on the task classifier, generating the sample weights from the class distribution of the training data.

The dropout rates are set to $p = 0.5$ and $p = 0.25$ for the task and domain classifiers, respectively. We use Adam [30] as the optimizer with a learning rate of 10^{-5}, training all the models for

TABLE III: Model structure of our architecture. The description includes the encoder, decoder, task classifier and domain classifier.

Block	Layer	Size In	Size Out	Activation
Encoder	1	900	512	ReLU
	2	512	256	ReLU
	3	256	128	ReLU
	4	128	16	Sigmoid
Decoder	1	16	128	ReLU
	2	128	256	ReLU
	3	256	512	ReLU
	4	512	900	ReLU
Task Classifier	1	16	21	ReLU
	2	21	14	ReLU
	3	14	2	Softmax
Domain Classifier	GR	16	16	None
	1	16	8	Sigmoid
	2	8	2	Softmax

25 epochs. All models are trained using Keras with Tensorflow [31]. The hyperparameter for the gradient reversal layer (λ) is initialized to zero and incremented by 0.025 for five epochs. After five epochs, we reduced the increment to 0.015 for the remainder of the training. The threshold γ for the variance regularization is set to 0.05 for all experiments, based on the stability of the domain classifier observed on the development set.

Our training strategy for the source data relies on cross-validation given the limited size of our data sets. We split the data into 10 partitions, with each partition as class-balanced as possible. One partition is reserved as the test set, which is only used to evaluate the performance of the system. From the remaining nine partitions, two are randomly selected as the development set, and the other seven partitions as the train set. We denote the experiments in a specific division of the data into train, development and test sets as a *run*. The data is partitioned by patient, where all the pixels of a single image belong only to the train, development, or test set within a specific run. With the cross-validation, every partition is eventually used as the test set. We build the system using the train set, maximizing the performance on the development set. The final model is then evaluated on the test set. We refer to this cross-validation process as a *trial*. Since the partitions of the data set can affect the performance of the system, we repeat this process 10 times, creating different partitions for each trial. We report the average results across the 10 trials. In addition to the train and development sets for the source data, we append train and development partitions from the auxiliary data set. The same auxiliary data is added to the training and development sets in each run. We use 34 lesions in the auxiliary training set and 12 lesions in the auxiliary development set.

Across all experimental conditions, we use consistent data splits to minimize sources of variation across conditions due to variations in the pairing of train, development, and test sets. Therefore, the comparisons across baselines are consistent since they are using the same partitions.

G. Preprocessing

Both the source and auxiliary data sets were preprocessed with signal inversion, spatial averaging, and SNR masking to reduce noise and reject pixels with low or saturated SNR. Each channel was chopped or zero padded to a consistent length. In addition, each data set had individual preprocessing steps to reduce variations between the domains. Figure 4 shows an example pixel from each domain and the initial preprocessing steps.

The preprocessing for each data set is identical through the second column in Figure 4, which shows the fluorescence decay following spatial averaging and signal inversion. Arrows 1A and 2A in the figure highlight the first dataset-dependent variation, where the different gain used in image acquisition changes the peak value of the channels by 100 units. In the next preprocessing step, the decays are calibrated and normalized to sum to 100 to put the peak values on a similar scale between data sets. In addition, each source data set channel had a temporal resolution of 0.25 ns and length of 300 samples after zero padding. The auxiliary data set had a temporal resolution of 0.16 ns and length of 375 samples after zero padding. The auxiliary data set was interpolated to match the length and sample rate of the source data set. The peak of each channel in the auxiliary data set was also shifted to approximately match the peak of each channel in the source data set (see Arrows 1B and 2B in Figure 4). Finally, Arrows 1C and 2C show an unmitigated difference between images. Both pixels represented in the figure are from benign lesions. However, the ratios between channels peaks are different, with the source domain pixel showing a much higher channel 3 peak than that of the auxiliary domain. While this could partially be due to inter-patient variability, the difference highlights the difficulty in merging images from heterogeneous, small data sets.

IV. Experiments

All experimental results are reported using the source domain on the cancer diagnosis task (classification of benign versus malignant). We use sensitivity and specificity as our performance metrics. We also report the average of these metrics.

A. Comparison with Baselines

We report baseline results from a variety of methods, as shown in Table IV. We use SVM and different feature selection methods with standard features as input, following the work of Jo et al. [2] and Marsden et al. [1]. The standard feature set includes 21 features derived from signal lifetime, intensity, and bi-exponential decay parameters. The lifetime (τ_k) and intensity (I_k) of a single channel are given by Equations 5 and 6 below:

$$\tau_k = \frac{\int t h_k(t) \partial t}{\int h_k(t) \partial t} \tag{5}$$

$$I_k = \int h_k(t) \partial t \tag{6}$$

where t specifies the sample time point and h_k denotes the deconvolved fluorescence decay for channel k. The deconvolved fluorescence decay was obtained by iterative least-squares deconvolution using a bi-exponential decay model, as shown in Equation 7:

$$h_k = \alpha_{fast,k} e^{-t/\tau_{fast,k}} + \alpha_{slow,k} e^{-t/\tau_{slow,k}} \tag{7}$$

where $\alpha_{fast,k}$ and $\alpha_{slow,k}$ are the coefficients of the two exponential terms and $\tau_{slow,k}$ and $\tau_{fast,k}$ determine the decay rates of the exponential terms. We also use two different feature selection methods: *sequential feature selection* (SFS) and L1 regularization. We report results for SVM classifiers for single-domain training, as well as the results for training with source and auxiliary sets (i.e., training and development sets), denoted as *joint* in Table IV.

In addition to the SVM classifiers with standard features, we list the single-domain results using the autoencoder and classifier neural net shown in Figure 1 and reported by Caughlin et al. [13]. This neural network structure uses the pre-processed fluorescence decays without iterative deconvolution and generates data-driven features. As detailed in Section III, the autoencoder for our approach is modified with a sigmoid activation at the bottleneck. We report the single-domain training performance using the modified autoencoder in Table

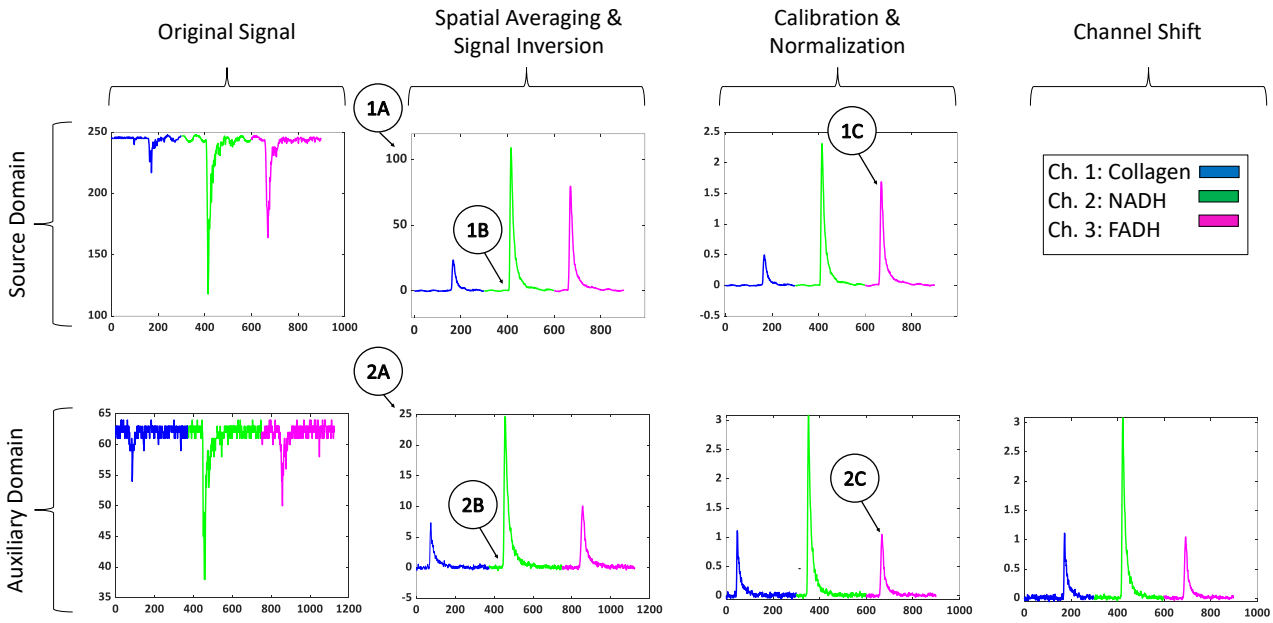


Fig. 4: Visualization of the preprocessing steps for source and auxiliary domains. Each domain was separately preprocessed to reduce variation from gain and peak locations. Peak value trends by channel remained different between domains, even among images from the same class. Top row: source domain. Bottom row: auxiliary domain.

IV to ensure that the improved results with our method are not due to an improved autoencoder structure. While the change in activation helps domain stability with gradient reversal, the sigmoid activation may lose some performance benefits resulting from sparsity in the bottleneck representation when using the *rectified linear unit* (ReLU) activation.

Table IV shows the results. Joint training reduces the average source domain performance in all baseline models. For the modified autoencoder method, joint training reduces the average performance by 0.62% compared with the source only model. The performance of the joint SVM model with L1 regularization dropped an average of 0.5% compared with the source only model. Joint training with SVM and SFS completely failed, with an average performance of 7.71% below the single domain SVM SFS model. The decrease in average performance highlights the problem of domain shift, which causes the source performance to decrease when the model is trained with both source and auxiliary sets, even in a fully-supervised setting. Furthermore, a similar performance drop in models trained using hand-crafted features from the deconvolved signal (SVM models) and the preprocessed signal without deconvolution (autoencoder models) shows that the domain shift problem cannot be solely attributed to differences in the imaging system impulse response.

In contrast to all other baseline models, our full neural network with domain adaptation and variance regularization improves source domain performance over all other methods including source-only training settings. Our full model significantly increases the average performance over the best SVM multi-domain baseline by 2.54% ($p = 0.0341$ using a one-tailed, paired t-test). Similarly, our full model shows no significant change in specificity and significantly increases sensitivity over the single-domain training using the same base structure (i.e., modified autoencoder trained on source data only) by 7.33% ($p = 0.0023$ using a two-tailed, paired t-test). The improved results show that domain shift between data centers can be reduced using gradient reversal.

TABLE IV: Benign versus malignant classification results for the proposed approach and the baselines.

Model	Sensitivity	Specificity	Average
AE [13]	87.50	67.58	77.54
AE Joint	70.08	81.17	75.62
AE (modified)	85.08	68.42	76.75
AE (modified) Joint	77.33	74.92	76.13
SVM SFS [13]	81.00	67.25	74.13
SVM SFS Joint	82.00	50.83	66.42
SVM L1 [13]	79.17	73.33	76.25
SVM L1 Joint	88.00	63.50	75.75
Proposed Approach	80.83	75.75	78.29

TABLE V: Contributions of the model's components. The best model uses both domains in training (joint), gradient reversal (GR), and variance regularization (VR).

Joint	GR	VR	Sensitivity	Specificity	Average
		✓	85.33	64.67	75.00
✓			77.33	74.92	76.13
✓	✓		80.00	71.67	75.83
✓	✓	✓	80.83	75.75	78.29

B. Contributions of the Model's Components

To evaluate the contributions of each component in our domain adaptation framework, we present a breakdown of each component in Table V. *Joint* refers to training with both source and auxiliary data in the training and development sets. *GR* refers to joint training with gradient reversal. *VR* refers to the addition of variance regularization to the bottleneck representation (discussed in Section III). Comparing rows two (Joint) and three (Joint+GR) from Table V, we find that gradient reversal failed to correct the performance drop when the variance regularization is not used. This setting produces similar source performance to the model trained without any domain adap-

TABLE VI: Effect of domain stability in model selection. Joint + GR: Joint model trained with source and auxiliary datasets implemented with gradient reversal. Joint + GR + VR: joint model with gradient reversal and variance regularization.

Model	Model Selection	Sens.	Spec.	Ave.
Joint + GR	–	77.17	72.33	74.75
	✓	80.00	71.67	75.83
Joint + GR + VR	–	81.50	74.25	77.87
	✓	80.83	75.75	78.29

tation. After analyzing the domain performance during adaptation, we found evidence of domain collapse where the domain classifier always predicts a single class. When this happens, the domain adaptation module does not achieve its goal of reducing the mismatch between source and auxiliary sets. We provide further analysis on domain collapse and the effect on domain adaptation in Section IV-D. The domain collapse problem is corrected with the variance regularization. Since domain adaptation failed until the addition of variance regularization, we confirm that the increased performance of the full model is due to an appropriate implementation of domain adaptation, rather than an unforeseen effect of variance regularization by training our source-only autoencoder model with variance regularization at the bottleneck (row 1 in Table V). As expected, the addition of variance regularization outside of domain adaptation reduces average performance by 1.75% compared with our base autoencoder. The full model using joint training, gradient reversal, and variance regularization successfully merged the domains, with an increase of 3.50% specificity and 0.83% sensitivity compared to joint training without domain adaptation. Furthermore, the full domain adaptation model outperforms single-domain classifier performance using the same autoencoder by an average of 1.54%. Taken together, Table V shows that all three components of our model (multi-center training, gradient reversal, and domain adaptation) are required to successfully train using multiple data sets.

C. Model Selection Constraints

In addition to the use of domain adaptation, the model performance is also affected by the model selection criteria. In general, the best model from the training process is chosen by the performance on the development set. However, in our domain adaptation framework, the task and domain classifiers are trained at the same time. However, the trade-off hyperparameter for gradient reversal increases over training. If the best model is selected based on the minimum development set performance without considering domain convergence, the model may be selected before the domains have merged. To prevent sub-optimal model performance, we used a model selection constraint in our full model, as described in Section III-B. We verify the efficacy of our model selection constraint for two settings. One setting uses joint training and gradient reversal, and the other setting uses joint training, gradient reversal and variance regularization. Table VI shows the results when we either select the model with the best results on the development set, even if the domain adaptation has not fulfilled its role, or we select the best results in the development set after waiting for the domain adaptation to converge (denoted with the symbol ✓). In both settings, adding the model selection constraint improves average performance by 1.08% and 0.42%, respectively. The greater need for model selection criteria without variance regularization may be due to increased domain instability.

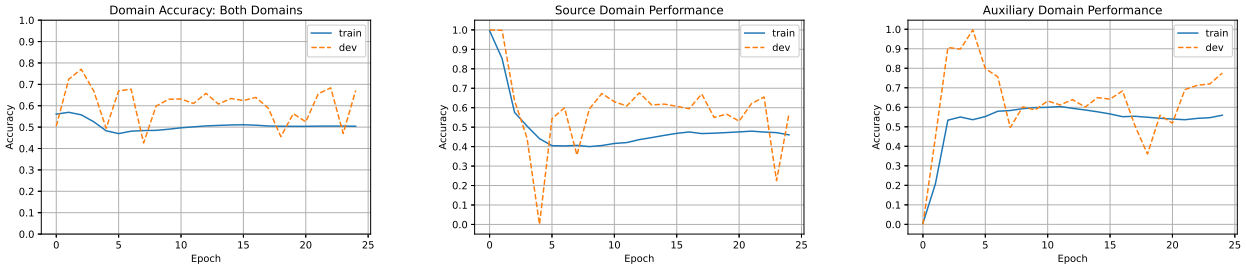
We hypothesized that when the domains have merged, the inclusion of the auxiliary data in the development set will increase source performance. The development set is used to identify the best model to be used in the test set. If the domains have not merged, the development performance may be skewed by higher classification performance on the auxiliary data set, affecting the performance of the source domain. We test this hypothesis for all three joint training scenarios: joint training without domain adaptation (Joint), joint training with gradient reversal (Joint + GR), and joint training, gradient reversal and variance regularization (Joint + GR + VR). Notice that in the three conditions, the auxiliary set is used on the training set. Table VII shows the results, denoting with the symbol ✓ when the auxiliary data is included in the development set. When the variance regularization is not used and the system ineffectively merges the domains (Joint and Joint + GR models), removing the auxiliary data from the model selection (i.e., development set) resulted in better performance on the source domains. Specifically, the joint model without domain adaptation dropped 0.70% on average. The joint model with domain adaptation dropped 1.46%. The decrease in performance with the auxiliary development data indicates that any advantages from a larger, more diverse data set were offset by the domain shift. In contrast, our full model with variance regularization and gradient reversal benefits from the addition of auxiliary data in the development set by 1.55% on average. Therefore, Table VII further confirms our hypothesis that source domain performance can be increased by the inclusion of data from another domain when domain shift has been corrected with domain adaptation.

TABLE VII: Effect of auxiliary data in model selection. Using the auxiliary data in the development set (aux. dev.) only improves performance when the domains are merged using gradient reversal (GR) and variance regularization (VR). Joint: model trained with source and auxiliary datasets. Joint + GR: Joint model implemented with gradient reversal. Joint + GR + VR: joint model with gradient reversal and variance regularization.

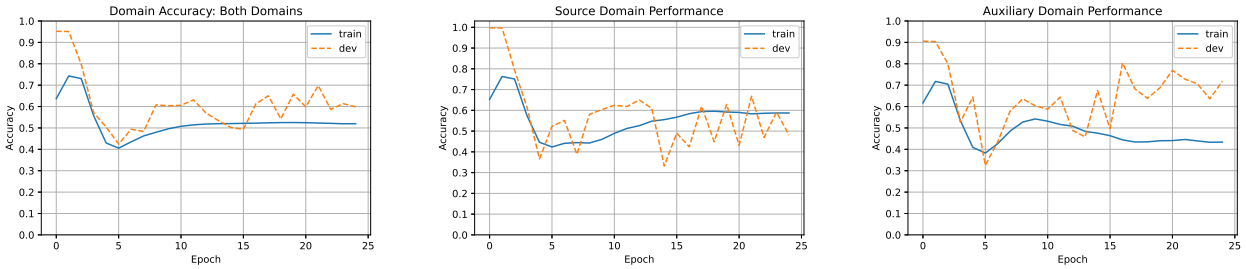
Model	Aux. Dev.	Sens.	Spec.	Avg.
Joint	–	78.33	75.33	76.83
	✓	77.33	74.92	76.13
Joint + GR	–	79.83	74.75	77.29
	✓	80.00	71.67	75.83
Joint + GR + VR	–	80.92	72.25	76.58
	✓	80.83	75.75	78.29

D. Domain Adaptation Verification

We plot the domain accuracy to monitor the effects of gradient reversal. Figure 5 shows the performance of the train and development set. We set the gradient reversal hyperparameter $\lambda = -0.25$ for the first epoch and increment it by 0.025 for five epochs and 0.015 thereafter. Note that this setting only applies to this section to verify domain adaptation. For the rest of the evaluation, we begin with $\lambda = 0$ and gradually increment it. The negative sign allows the domain accuracy to initially increase , showing that the domain classifier is working properly. Once the value of λ increases and domain adaptation progresses, the domain accuracy should reach near random performance, indicating domain invariance. The top row of Fig. 5 shows this process for the model without variance regularization. The development accuracy reaches above 75.0% and then decreases as expected (Fig. 5a, epoch 3). However, the way the model achieves 50% domain accuracy matters. For example, Kim and Kim [32] show that semi-supervised domain adaptation does not ensure that the representations are well-ordered. Rather than creating a compact and homogeneous mix of the domains, the source domain can encompass distinct clusters of the target domain and prevent improvements in target performance [32]. Similarly, we find that the



(a) Domain Accuracy: Both Domains (b) Source Domain Performance (c) Auxiliary Domain Performance



(d) Domain Accuracy: Both Domains (e) Source Domain Performance (f) Auxiliary Domain Performance

Fig. 5: Importance of domain classifier performance for each domain. Top row: Domain classifier performance for the model without variance regularization. Bottom row: Domain classifier performance for the model with variance regularization. Left column: Domain accuracy for the development set using both domains. Center column: Domain accuracy for the development set samples from the source data set. Right column: Domain accuracy for the development set samples from the auxiliary data set.

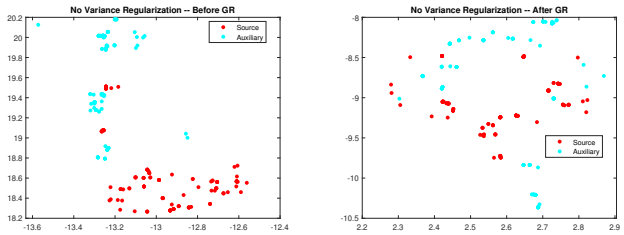
domain classifier can reach a 50% accuracy by always predicting a single domain. In the top row of Figure 5, the initial domain performance is split between the source and auxiliary data sets, with the source accuracy at 100% (Fig. 5b, epoch 1) and the auxiliary performance at 0% (Fig. 5c, epoch 1). Furthermore, the domain performance again collapses at epoch 4, when the source domain performance is 0% (Fig. 5b, epoch 4) and the auxiliary domain performance is 100% (Fig. 5c, epoch 4). When the domain classifier predicts a single domain, the overall domain accuracy calculated using the development sets from both domains reaches ∼50%, but incorrectly indicates domain invariance. In contrast, the model with variance regularization reaches approximately 95% initial accuracy (Fig. 5d, epoch 2), with domain accuracy for both domains at similar values (Figs. 5e and 5f, epoch 25). The overall performance of the domain classifier is near random by the end of training. Additionally, the collapse at epoch 4 is not present, showing that the model with variance regularization had greater stability.

We expect the changes in domain invariance to appear in *t-distributed stochastic neighbor embedding* (T-SNE) plots as improved domain invariance. T-SNE plots use distribution matching to create a lower dimensional representation in which points that are close in the original distribution remain nearby in the reduced representation [33]. We randomly sample 10 data points from each image in the development set using the 16-dimensional bottleneck representation to create the T-SNE plots. We use the MATLAB's implementation of T-SNE from the Statistics and Machine Learning Toolbox. We use the sampling of points to prevent the visualization from showing local structure associated with the similarity of data from each lesion rather than the more global structure associated with differences in the domains (Kobak and Berens [34] describe the trade-off between local and global structure visualization in medical data cases where data points easily cluster by subgroups in the data). The resulting T-SNE plots for the models trained with and without variance regularization are shown in Figure 6. Rows 1 and 2 show the T-SNE embedding of the same points before and after gradient reversal without variance regularization and with variance regularization, respectively. In contrast, Figure 6d shows that the model at the end of domain adaptation has the more distributed T-SNE plots, with the data from both domains mixed together. Taken together, the T-SNE plots show a two-dimensional visualization that indicates improved domain invariance following domain adaptation.

V. Conclusions

This study presented a neural network framework for merging data from multiple maFLIM collection sites using gradient reversal. Our experiments showed that joint training data from two centers without considering the domain shift decreases classifier performance on an individual data set. However, on small data sets, gradient reversal suffers from domain collapse even in the supervised setting. We presented an architecture adjustment and a variance regularization method to stabilize the training process. We also detailed model selection procedures to maximize performance and explored the effect of multi-domain development sets on source domain performance. Our analysis shows that when the domains are properly merged, the additional diversity from the auxiliary data in the development set improves source domain performance. In total, our full model achieves 1.63% increase in sensitivity and 2.45% increase in specificity compared to the best joint trained baseline model. In addition, our full model achieves 0.69% average improvement over the best source domain only model.

In future work, we plan to extend our approach to the semi-supervised setting, using task labels for the source domain only and introducing the images from the auxiliary domain as unlabeled data. Labeling the images is one of the difficult tasks, since it involves surgical resection and histopathological diagnosis. Therefore, we expect that this semi-supervised approach will allow us to leverage more images. In addition, we would like to explore the generation of domain invariant representations with skin and oral lesions. We expect



(a) Joint (b) Joint + GR

(c) Joint + VR (d) Joint + GR + VR

Fig. 6: Visualization of domain invariance using T-SNE plots. Top row: T-SNE representation of the bottleneck embedding for the Joint, and Joint + GR models without variance regularization. Bottom row: T-SNE representation of the bottleneck embedding for the Joint + VR and Joint + GR + VR models. Left column: Bottleneck representation before GR. Right column: Bottleneck representation after GR. Domain invariance improves during training with variance regulation.

domain shift from differences in sub-populations and imaging center, as discussed here, as well as in the underlying tissue and pathology characteristics. Extension to domain adaptation for multiple imaging centers and lesion types will aid the development of a single robust classifier that can be used by the same endoscope for multiple applications.

REFERENCES

[1] M. Marsden, B. W. Weyers, J. Bec, T. Sun, R. F. Gandour-Edwards, A. C. Birkeland, M. Abouyared, A. F. Bewley, D. G. Farwell, and L. Marcu, "Intraoperative margin assessment in oral and oropharyngeal cancer using label-free fluorescence lifetime imaging and machine learning," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 3, pp. 857–868, March 2021.

[2] J. Jo, S. Cheng, R. Cuenca-Martinez, E. Duran-Sierra, B. Malik, B. Ahmed, K. Maitland, Y.-S. Cheng, J. Wright, and T. Reese, "Endogenous fluorescence lifetime imaging (FLIM) endoscopy for early detection of oral cancer and dysplasia," in *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2018)*, Honolulu, HI, USA, July 2018, pp. 3009–3012.

[3] P. Vasanthakumari, R. Romano, R. Rosa, A. Salvio, V. Yakovlev, C. Kurachi, and J. Jo, "Classification of skin-cancer lesions based on fluorescence lifetime imaging," in *SPIE Medical Imaging, 2020: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 11317, Houston, TX, USA, February 2020.

[4] W. Li, S. Liao, Q. Feng, W. Chen, and D. Shen, "Learning image context for segmentation of the prostate in ct-guided radiotherapy," *Physics in Medicine & Biology*, vol. 57, no. 5, p. 1283, 2012.

[5] A. Das, I. Tashev, and S. Mohammed, "Ultrasound based gesture recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, New Orleans, LA, USA, March 2017, pp. 406–410.

[6] N. M. Khan, N. Abraham, and M. Hon, "Transfer learning with intelligent training data selection for prediction of alzheimer's disease," *IEEE Access*, vol. 7, pp. 72 726–72 735, June 2019.

[7] H. Guan and M. Liu, "Domain adaptation for medical image analysis: a survey," *arXiv preprint arXiv:2102.09508*, 2021.

[8] E. H. Pooch, P. Ballester, and R. C. Barros, "Can we trust deep learning based diagnosis? the impact of domain shift in chest radiograph classification," in *International Workshop on Thoracic Image Analysis*. Lima, Peru: Springer, October 2020, pp. 74–83.

[9] L. Zhang, X. Wang, D. Yang, T. Sanford, S. Harmon, B. Turkbey, B. J. Wood, H. Roth, A. Myronenko, D. Xu, and Z. Xu, "Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation," *IEEE transactions on medical imaging*, vol. 39, no. 7, pp. 2531–2540, 2020.

[10] M. Saiz-Vivó, A. Colomer, C. Fonfría, L. Martí-Bonmatí, and V. Naranjo, "Supervised domain adaptation for automated semantic segmentation of the atrial cavity," *Entropy*, vol. 23, no. 7, p. 898, 2021.

[11] C. Chen, Q. Dou, H. Chen, J. Qin, and P.-A. Heng, "Synergistic image and feature adaptation: Towards cross-modality domain adaptation for medical image segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, Honolulu, Hawaii, July 2019, pp. 865–872.

[12] L. Diao, H. Guo, Y. Zhou, and Y. He, "Bridging the gap between outputs: Domain adaptation for lung cancer IHC segmentation," in *2021 IEEE International Conference on Image Processing (ICIP)*. Anchorage, AK: IEEE, Sept. 2021, pp. 6–10.

[13] K. Caughlin, E. Duran-Sierra, S. Cheng, R. Cuenca, B. Ahmed, J. Ji, V. Yakovlev, M. Martinez, M. Al-Khalil, H. Al-Enazi, J. A. Jo, and C. Busso, "End-to-end neural network for feature extraction and cancer diagnosis of in vivo fluorescence lifetime images of oral lesions," in *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2021)*, Guadalajara, Mexico, October-November 2021.

[14] R. Romano, R. Teixeira Rosa, A. Salvio, J. Jo, and C. Kurachi, "Multispectral autofluorescence dermoscope for skin lesion assessment," *Photodiagnosis Photodyn. Ther.*, vol. 30, p. 101704, June 2020.

[15] E. Duran-Sierra, S. Cheng, R. Cuenca, B. Ahmed, J. Ji, V. V. Yakovlev, M. Martinez, M. Al-Khalil, H. Al-Enazi, Y.-S. L. Cheng *et al.*, "Machine-learning assisted discrimination of precancerous and cancerous from healthy oral tissue based on multispectral autofluorescence lifetime imaging endoscopy," *Cancers*, vol. 13, no. 19, p. 4751, 2021.

[16] B. Chen, Y. Lu, W. Pan, J. Xiong, Z. Yang, W. Yan, L. Liu, and J. Qu, "Support vector machine classification of nonmelanoma skin lesions based on fluorescence lifetime imaging microscopy," *Analytical chemistry*, vol. 91, no. 16, pp. 10 640–10 647, August 2019.

[17] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*, vol. 37. Lille, France: PMLR, June 2015, pp. 1180–1189.

[18] X. Zhang, C. Broun, R. Mersereau, and M. Clements, "Automatic speechreading with applications to human-computer interfaces," *EURASIP Journal on Advances in Signal Processing*, vol. 1, pp. 1228–1247, January 2002.

[19] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling, "Do not have enough data? deep learning to the rescue!" in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, New York, New York, April 2020, pp. 7383–7390.

[20] S. Li, C. H. Liu, B. Xie, L. Su, Z. Ding, and G. Huang, "Joint adversarial domain adaptation," in *Proceedings of the 27th ACM International Conference on Multimedia*, Nice, France, October 2019, pp. 729–737.

[21] V. K. Kurmi, V. K. Subramanian, and V. P. Namboodiri, "Informative discriminator for domain adaptation," *Image and Vision Computing*, vol. 111, p. 104180, July 2021.

[22] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, June 2014.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (PMLR 2015)*, vol. 37, Lille, France, July 2015, pp. 448–456.

[24] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Proceedings of the 32nd international conference on neural information processing systems*, 2018, pp. 2488–2498.

[25] G. Chen, P. Chen, Y. Shi, C.-Y. Hsieh, B. Liao, and S. Zhang, "Rethinking the usage of batch normalization and dropout in the training of deep neural networks," *arXiv preprint arXiv:1905.05928*, 2019.

[26] P. Chong, L. Ruff, M. Kloft, and A. Binder, "Simple and effective prevention of mode collapse in deep one-class classification," in *2020 International Joint Conference on Neural Networks (IJCNN)*. Glasgow, UK: IEEE, July 2020, pp. 1–9.

[27] S. Cheng, R. Cuenca, B. Liu, B. Malik, J. Jabbour, K. Maitland, J. Wright, Y.-S. Cheng, and J. Jo, "Handheld multispectral fluorescence lifetime imaging system for in vivo applications," *Biomedical Optics Express*, vol. 5, no. 3, pp. 921–931, March 2014.

[28] L. America, "Safe use of lasers, ansi z136. 1–2007," *American National Standards Institute*, 2007.

[29] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, San Diego, CA, USA, May 2015, pp. 1–13.

[31] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Symposium on Operating Systems Design and Implementation (OSDI 2016)*, Savannah, GA, USA, November 2016, pp. 265–283.

[32] T. Kim and C. Kim, "Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation," in *European Conference on Computer Vision*. Glasgow, UK: Springer, August 2020, pp. 591–607.

[33] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, November 2008.

[34] D. Kobak and P. Berens, "The art of using t-sne for single-cell transcriptomics," *Nature communications*, vol. 10, no. 1, pp. 1–14, 2019.