Truly Generalizable Radiograph Segmentation With Conditional Domain Adaptation

HUGO N. OLIVEIRA*, EDEMIR FERREIRA, AND JEFERSSON A. DOS SANTOS* 
Computer Science Department, Universidade Federal de Minas Gerais, Belo Horizonte 31270-901, Brazil

Corresponding author: Hugo N. Oliveira (oliveirahugo@dcc.ufmg.br)

This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001, in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant 424700/2018-2 and Grant 311395/2018-0, and in part by the Minas Gerais Research Foundation (FAPEMIG) under Grant APQ-00449-17.

ABSTRACT

Digitization techniques for biomedical images yield disparate visual patterns in radiological exams. These pattern differences, which can be viewed as a domain-shift problem, may hamper the use of data-driven approaches for inference over these images, such as Deep Neural Networks. Another noticeable difficulty in this field is the lack of labeled data, even though in many cases there is an abundance of unlabeled data available. Therefore, an important step in improving the generalization capabilities of these methods and mitigate domain-shift effects is to perform unsupervised or semi-supervised adaptation between different domains of biomedical images. In this work, we propose a novel approach for segmentation of biomedical images based on Generative Adversarial Networks. The proposed method, named Conditional Domain Adaptation Generative Adversarial Network (CoDAGAN), merges unsupervised networks with supervised deep semantic segmentation architectures in order to create a semi-supervised method capable of learning from both unlabeled and labeled data, whenever labeling is available. We conducted experiments to compare our method with traditional and state-of-the-art baselines by using several domains, datasets, and segmentation tasks. The proposed method yielded consistently better results than the baselines in scarce labeled data scenarios, achieving Jaccard values greater than 0.9 and good segmentation quality in most tasks. Unsupervised Domain Adaptation results were observed to be close to the Fully Supervised Domain Adaptation used in the traditional procedure of fine-tuning pretrained networks.

INDEX TERMS

Deep learning, domain adaptation, biomedical images, semantic segmentation, image translation, semi-supervised learning.

I. INTRODUCTION

Radiology has been a useful tool for assessing health conditions since the last decades of the 19th century, when X-Rays were first used for medical purposes. Since then, it has become an essential tool for detecting, diagnosing and treating medical issues. More recently, algorithms have been coupled with radiology imaging techniques and other medical information in order to provide second opinions to physicians via Computer-Aided Detection/Diagnosis (CAD) systems. In this context, segmentation is a very important task [1], [2]. Most common segmentation tools are typically used for delineating nodules, bones or other kind of tissues in an unsupervised way but it is also very common the employment of interactive segmentation.

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano*.

In recent decades, Machine Learning algorithms were incorporated into more modern CAD systems, providing automatic methodologies for finding patterns in big data scenarios, improving the capabilities of human physicians. During the last half decade traditional Machine Learning pipelines have been losing ground to integrated Deep Neural Networks (DNNs) that can be trained from end-to-end [3]. DNNs can integrate both the steps of feature extraction and statistical inference over unstructured data, such as images, temporal signals or text. Deep-based models for images usually are built upon some form of trainable convolutional operation [4]. Convolutional Neural Networks (CNNs) are the most popular architectures for supervised image classification in Computer Vision. Variations of CNNs can be found in both detection [5]–[7] and segmentation [8]–[10] models.

The main challenge to automatically perform suitable semantic segmentation via supervised learning is that, in the
medical imaging community, labeled data is often limited. At the same time, there are large amounts of unlabeled datasets that can be used for unsupervised learning. To make matters worse, the generalization of DNNs is normally limited by the variability of the training data, which is a major hamper, as different digitization techniques and devices used to acquire different datasets tend to produce biomedical images with distinct visual features [11]. In other words, different digitization techniques lead us to model each dataset as a separate domain, aiming to compensate for these distinct visual properties, known as “domain-shift” in the machine learning community. Therefore, the study of methods that can use both labeled and unlabeled data is a hot research topic in both Computer Vision and Biomedical Image Processing. A more detailed description of the state of biomedical image understanding using DNNs is out of the scope of this paper and can be fully appreciated in the survey done by Litjens et al. [3].

Domain Adaptation (DA) [12] methods are often used to improve the generalization of DNNs over images in a supervised manner. The most popular method for deep DA is Transfer Learning via Fine-Tuning pre-trained DNNs from larger datasets, such as ImageNet [13]. However, Fine-Tuning is a Fully Supervised Domain Adaptation (FSDA) strategy only capable of learning from labeled data, ignoring the potentially larger amounts of unlabeled data available. Therefore, during the last years, several approaches have been proposed for Unsupervised Domain Adaptation (UDA) [14], [15] and Semi-Supervised Domain Adaptation (SSDA) [16].

In this paper, we introduce a novel Deep Learning-based DA method that works for the whole spectrum of UDA, SSDA and FSDA, being able to learn from both labeled and unlabeled data. An overview of the proposed approach, named Conditional Domain Adaptation Generative Adversarial Network (CoDAGAN), is presented in Figure 1. CoDAGANs allows multiple datasets to be used conjointly in the training procedure, enforcing automatic domain-shift compensation. It is important to notice that most of the other modern methods in the visual DA literature [17]–[21] only allow for the pairwise training of one source and one target domain. In contrast to these pairwise methods, CoDAGANs learn to perform supervised inference over a common isomorphic representation built upon samples drawn from marginal distributions of multiple domains, as shown in Figure 1. In other words, CoDAGANs are able to take into account samples from distinct datasets with varying distributions drawn from the joint domain distribution to improve generalization.

We claim the following contributions for the current manuscript: 1) a novel approach to perform effective DA in the context of biomedical image segmentation tasks; 2) a new strategy to perform unpaired UDA and SSDA by using generative adversarial networks without the limits of pairwise training; 3) a complete review about the state of the art in image translation for domain adaptation. We also perform an extensive comparison with pairwise baseline methods in the literature [18], [21], [22] – further described in Sections III and V-D – assessing the improvements and limits of CoDAGANs for dense labeling DA in radiographs.
Other sections in this paper are organized as follows. Sections II and III present the previous works that paved the way for the proposal of CoDAGANs. Section IV describes the CoDAGAN modules, architecture, subroutines and loss components. Section V shows the experimental setup discussed in this paper, including datasets, hyperparameters, experimental protocol, evaluation metrics and baselines. Section VI introduces and discusses the results found during the exploratory tests of CoDAGANs for UDA, SSDA and FSDA in a quantitative and qualitative manner. At last, Section VII finalizes this work with our final remarks and conclusions regarding the methods and experiments shown in this work.

II. Background
This section presents background concepts that are crucial for the making of this paper. It comprises recent developments in the literature regarding Semantic Segmentation (Section II-A) and Image Translation (Section II-B).

A. DEEP SEMANTIC SEGMENTATION
Since the resurgence of Neural Network technology as Deep Learning in the early 2010’s, they have been adapted to perform segmentation tasks. Initial approaches used traditional CNNs [4] for image classification in dense labeling tasks by applying them to image patches. Patch-based approaches were observed to be unreasonably slow in these tasks, resulting in the proposal of Fully Convolutional Networks (FCNs) [8], which provided an end-to-end pipeline deep segmentation. These networks have similar architectures to traditional CNNs, which allows for transferring the knowledge from sparse to dense labeling scenarios. One should notice that FCNs use the same loss functions as CNNs for image classification, as dense labeling can be seen as a collection of sparse labels for each pixel in an image. Therefore, Cross Entropy is the most common loss for supervised semantic segmentation and it can be expressed by:

\[ \mathcal{L}_{\text{sup}}(Y, \hat{Y}) = -Y \log(\hat{Y}) - (1 - Y) \log(1 - \hat{Y}), \] (1)

where \( Y \) represents the pixel-wise semantic map and \( \hat{Y} \) the probabilities for each class for a given sample.

Deconvolutional Networks [9], [10] based on transposed convolutions have grown in popularity in recent years, allowing for learnable upsampling of low spatial-resolution activations in the middle of the network. These are Encoder-Decoder networks, as they are often composed of downsampling convolution blocks followed by symmetric upsampling transposed convolutions. U-Nets [9] built upon the idea of skip connections from FCNs, relying heavily on them for the decoding (upscaling) procedure. Most of these Encoder-Decoder networks [9], [10] are based on VGG architectures [23], composed only of \( 3 \times 3 \) convolutions that halve the spatial resolution of the feature map at each convolutional block. U-Nets rely heavily on Skip Connections, employing them on each Encoder/Decoder block pair, as shown in Figure 2.

B. IMAGE-TO-IMAGE TRANSLATION
Image-to-Image Translation Networks are GANs [24] capable of transforming samples from one image domain into images from another. Access to paired images from the two domains simplifies the learning process considerably, as losses can be devised using only pixel-level or patch-level comparisons between the original and translated images [25]. Paired Image-to-Image Translation can be achieved, therefore, by Conditional GANs (CGANs) [26] coupled with simple regression models [27]. In order to achieve image translation, the adversarial components are added to a paired regression loss \( \mathcal{L}_{\text{pair}}(X_A^{(i)}, X_B^{(i)}) \) between a pair of samples of index \( i \) for datasets \( X_A \) and \( X_B \) from domains \( A \) and \( B \):

\[ \mathcal{L}_{\text{pair}}(X_A^{(i)}, X_B^{(i)}) = \mathbb{E} \left\| X_A^{(i)} - G(X_B^{(i)}) \right\|. \] (2)

This regression loss is usually the L1 loss, as it tends to produce less blurry results than the Mean Squared Error (MSE) loss [25].

Requiring paired samples reduces the applicability of image-to-image translation to a very small and limited subset of image domains where there is the possibility of generating paired datasets. This limitation motivated the creation of Unpaired Image-to-Image Translation methods [28]–[30]. These networks are based on the concept of Cycle-Consistency, which models the translation process between two image domain as an inversible process represented by a cycle. These networks are based on the concept of Cycle-Consistency, which models the translation process between two image domain as an inversible process represented by a cycle, as can be seen in Figure 3. This cyclic structure allows for Cycle-Consistent losses to be used together with the adversarial loss components of traditional GANs.

A Cycle-Consistent loss can be formulated as follows: let \( A \) and \( B \) be two image domains containing unpaired image samples \( X_A \) and \( X_B \). Consider then two functions \( G_{A \rightarrow B} \) and \( G_{B \rightarrow A} \) that perform the translations \( A \rightarrow B \) and \( B \rightarrow A \).
The case for translations $B \rightarrow A \rightarrow B$ is analogous to the case of $A \rightarrow B \rightarrow A$.

Newer architectures as Unsupervised Image-to-Image Translation (UNIT) [29] and Multimodal Unsupervised Image-to-Image Translation (MUNIT) [30] achieve state-of-the-art realism in image translation by optimizing for cycle consistency in the bottleneck activations of the generators. In other words, UNIT and MUNIT are trained by not only minimizing the expectations $\mathbb{E}[X_A^{(i)} - X_A^{(i)}_{A \rightarrow B \rightarrow A}]$ and $\mathbb{E}[X_B^{(i)} - X_B^{(i)}_{B \rightarrow A \rightarrow B}]$, but also in the bottleneck activations of $G_{A \rightarrow B}$ and $G_{B \rightarrow A}$. One important property of this representation is that it forms an isomorphism between $A$ and $B$, which is explored by CoDAGANs, as further explained in Section IV.

Some efforts have been spent in proposing Unpaired Image Translation GANs for multi-domain scenarios, as the case of StarGANs [31], [32], but these networks do not explicitly present isomorphic representations of the data, as UNIT and MUNIT architectures do. Other advantages of UNIT and MUNIT over StarGANs is that they also compute reconstruction losses on the isomorphic representations, beside the traditional Cycle-Consistency between real and reconstructed images. CoDAGANs were built to be agnostic to the image translation network used as basis for the implementation, being able to transform any Image Translation GAN that has an isomorphic representation of the data into a multi-domain architecture with only minor changes to the generator and discriminator networks.

Traditional DA techniques perform knowledge transfer between a single pair of datasets: a source and a target datasets. In many cases it is advantageous to acquire as much data as possible from multiple sources, mainly when there is a lack of labels. Multi-source methods [33]–[37] try to infer a joint probability distribution $p_{X_1, X_2, \ldots, X_N}$ from a multitude of source data $X_1, X_2, \ldots, X_N$, each one with its own marginal probability distribution $p_{X_1}, p_{X_2}, \ldots, p_{X_N}$. These methods must infer joint distributions for the domains based only on the marginal distributions of the source data. CoDAGANs can be classified as a multi-source and multi-target DA method with the caveat that the distinction between source and target data is not clear in these DNNs, as translations and knowledge transfer are performed across all pairs of domains.

As pointed by Csurka [38], Domain Generalization is closely related to multi-source DA, as the objective is often to average the knowledge obtained from related source domains. Most Domain Generalization methods in the literature are based on this premise [39]–[43], including CoDAGANs. Image-to-Image Translation for DA is further discussed in the next section.

III. RELATED WORK

Several surveys on Visual Domain Adaptation [12], [38], [44]–[46] assess that there is an abundance of methods focused on DA for classification tasks in the field, but other tasks such as segmentation and detection have a much more scarce literature. Therefore, there is a lot of improvement to be made mainly in UDA and SSDA, as FSDA can be achieved with simple Fine-Tuning for these tasks, given enough labeled samples.

Since the introduction of Image-to-Image Translation GANs, several works [15], [18]–[21], [47], [48] have used these architectures to perform Domain Adaptation between image domains. In the following paragraphs, when available, we will mainly focus on the experiments of the literature in dense labeling tasks.
As far as the authors are aware, the first use of Image-to-Image Translation for Domain Adaptation purposes was shown by CoGANs [15]. This work showed UDA for digit classification between the MNIST [49] and USPS [50] datasets. While MNIST contains well-behaved, preprocessed and high-contrast handwritten digit samples, USPS better mimics a real-world scenario for digit classification. Thus, being able to adapt a digit classifier from MNIST to USPS without using labels from the target set is a challenging problem. One should notice that CoGANs still did not present UDA results in dense labeling tasks.

Cycle-Consistent Adversarial Domain Adaptation (CyCADA) [18] was built upon CycleGANs to perform UDA in dense labeling tasks – more specifically semantic segmentation. As most other papers in the area, CyCADA relies on synthetic data from realistic 3D simulations such as third person games to acquire labeled data for outdoor scene classification. It is much less time-consuming to annotate synthetic images from these simulations in an automated or semi-automated manner than to label entire datasets from scratch with pixel-level annotations, such as Pascal VOC [51]. CyCADA achieves UDA by attaching an FCN to the end of a CycleGAN, as shown in Figure 4, limiting it to adapting between a pair of source and target domains \( \{S, T\} \).

One should notice in this architecture that in the case of total lack of target labels \( Y_T \) – that is, in a UDA scenario – semantic consistency gradients are successfully fed to \( G_{T \rightarrow S} \) due to its proximity to \( M \), but very small gradient intensities flow from \( M \) to \( G_{T \rightarrow S} \) in \( S \rightarrow T \rightarrow S \) (Figure 4a). This might represent an imbalance in the training of \( G_{S \rightarrow T} \) and \( G_{T \rightarrow S} \), which is not desirable for DA.

The final loss for CyCADA \( (\mathcal{L}_{\text{CyCADA}}) \) is composed of a Cycle-Consistency loss \( \mathcal{L}_{\text{cyc}} \), adversarial loss component \( \mathcal{L}_{\text{adv}} \), for the pair of generators, adversarial loss component \( \mathcal{L}_{\text{adv}} \) for the pair of discriminators and a supervised Semantic Consistency loss \( \mathcal{L}_{\text{sem}} \). CyCADA reports successful UDA results between the synthetic GTA5 [52] dataset and the real-world CityScapes dataset [53]. CyCADA reports mIoU results of 35.4%, frequency weighted Intersection over Union (fwIoU) of 73.8% and Pixel Accuracy of 83.6% in translations between GTA5\( \rightarrow \)CityScapes. Several works improved on CyCADA by plugging a semantic segmentation DNN on one end of an Unpaired Image-to-Image Translation network [19]–[21], achieving comparable results on Computer Vision datasets.

Oliveira and dos Santos [21] used Unpaired Image-to-Image Translation to perform UDA, SSDA and FSDA between CXR datasets. As the previously mentioned approaches [18]–[20], Oliveira et al.’s approach is only able to perform DA between a single pair of domains due to the supervised DNN being added to one of the ends of a Cycle Consistent GAN. Oliveira and dos Santos [21] report Jaccard results in the Montgomery dataset [54] ranging from 88.2% in the UDA scenario to of 93.18% in the FSDA scenario, surpassing both Fine-Tuning and From Scratch training in scarce label scenarios. Several other similar approaches for biomedical image segmentation using image translation were proposed [22], [55]–[57]. However, all of them can be ultimately reduced to traditional Domain-to-Domain approaches (i.e. CyCADA [18], I2IAdapt [19] and DCAN [20]), as none of them are neither conditional neither multi-domain and perform supervised learning on the translated samples. A concise comparison between the previous literature and CoDAGANs can be seen in Table 1.

As shown, most methods simply combine the supervised learning from classification or segmentation schemes with a supervised or unsupervised image translation architecture to perform UDA, attaching an FCN-like architecture at one end of the image translation. From now on these models will be referenced as Domain-to-Domain (D2D) methods due to their limitations in allowing only pairwise training. CoDAGANs apply a similar framework to D2D in order to perform UDA, SSDA and FSDA, mixing the unsupervised learning of Cycle-Consistent GANs with the supervised pixel-wise learning of an Encoder-Decoder architecture. Two crucial distinctions between D2D methods and CoDAGANs must be addressed, though: 1) only one Encoder, one Decoder and...
The shape of $I$ depends on the architecture choice for $G$. UNIT, for example, assumes a single latent space between the image domains, while MUNIT separates the content of an image from its style. CoDAGANs feeds the whole latent space to the supervised model when it is based on UNIT and only content information when it is built upon MUNIT, as the style vector has no spatial resolution and as we intend to ignore style and preserve content.

A training iteration on a CoDAGAN follows the sequence presented in Figure 5. The generator network $G$ – such as U-nets [9] and Variational Autoencoders [58] – is an Encoder-Decoder architecture. However, instead of mapping the input image into itself or into a semantic map as its Encoder-Decoder counterparts, it is capable of translating samples from one image dataset into synthetic samples from another dataset. The encoding half of this architecture ($G_E$) receives images from the various datasets and creates an isomorphic representation somewhere between the image domains in a high dimensional space. This code will be henceforth described as $I$ and is expected to correlate important features in the domains in an unsupervised manner [15]. Decoders ($G_D$) in CoDAGAN generators are able to read $I$ and produce synthetic images from the same domain or from other domains used in the learning process. This isomorphic representation is an integral part of both UNIT [29] and MUNIT [30] translations, as they also enforce good reconstructions for $I$ in the learning process. It also plays an essential role in CoDAGANs, as all supervised learning is performed on $I$.

As shown in Figure 5, CoDAGANs include five unsupervised subroutines: a) Encode, b) Decode, c) Reencode, d) Redecode and e) Discriminate; and two f) Supervision subroutines, which are the only labeled ones. These subroutines will be detailed further in the following paragraphs.

**a: ENCODE**

First, a pair of datasets $a$ (source) and $b$ (target) are randomly selected among the potentially large number of datasets used in training. A minibatch $X_a$ of images from $a$ is then appended to a code $h_a$ generated by a One Hot Encoding scheme, aiming to inform the encoder $G_E$ of the samples’ source dataset. The 2-uple \(\{X_a, h_a\}\) is passed to the encoder $G_E$, producing an intermediate isomorphic representation $I_a$ for the input $X_a$ according to the marginal distributions computed by $G_E$ for dataset $a$.

**b: DECODE**

The information flow is then split into two distinct branches: 1) $I_a$ is fed to the supervised model $M$; 2) $I_a$ is appended to a code $h_b$ and passed through the decoder $G_D$ conditioned to dataset $b$. The function $G_D(I_a, h_b)$ produces $X_{a\rightarrow b}$, which is a translation of images in the minibatch $X_a$ with the style of dataset $b$.

**c: REENCODE**

The Reencode procedure performs the same operation of generating an isomorphic representation as the Encode...
the isomorphic representation is decoded as in $G$ subroutine, but receiving as input the synthetic image $X$. More specifically, the reencoded isomorphic representation $I$ produced by the adversarial loss component $L_D$ are merged in a single batch and passed to $M$ sample at the end of Encode and Reencode subroutines, for each $a$ which has a corresponding label $Y_a$. The model $M$ in this scenario, $G$ in this subroutines, is trained to perform inference on isomorphic encodings of both originally labeled data ($M$ of other datasets ($G$) in this scenario, $G$ is universal to all datasets used in the training procedure. Therefore any learning performed on $I_a$ and $I_{a-b}$ is universal to all datasets used in the training procedure. Instead of performing only the translation $a \rightarrow b \rightarrow a$ for the randomly chosen datasets $a$ and $b$, all mentioned subroutines are run simultaneously for both $a \rightarrow b \rightarrow a$ and $a \rightarrow b \rightarrow a$, as in UNIT [29] and MUNIT [30]. Translations $b \rightarrow a \rightarrow b$ are analogous to the $a \rightarrow b \rightarrow a$ case described previously.

One should notice that $G_E$ performs spatial downsample, while $G_D$ performs upsample, consequently the model $M$ should take into account the amount of downsampling layers in $G_E$. More specifically, we removed the first two layers of

FIGURE 5. Training procedure for CoDAGANs. This figure exemplifies a translation $a \rightarrow b \rightarrow a$, but the translation $b \rightarrow a \rightarrow b$ - which is performed simultaneously to the procedure for $a \rightarrow b \rightarrow a$ - is analogous. Notice that the reconstruction losses are omitted from this view of our architecture for simplification. The Encode routine transforms the real images in the mini-batch $X_a$ into the isomorphic representation $I_a$ between the datasets (through $G_E$), followed by the Decode subroutine, which builds (using $G_D$) a corresponding fake mini-batch $X_{a-b}$ according to $I_a$. The Reencode procedure reconstructs the isomorphic representation $I$ according to $X_{a-b}$. At last, the Redecode subroutine reconstructs the image $X_{a-b-a}$ according to $I_{a-b}$. The Discriminate subroutine tries to discern between real ($X_a$) and synthetic ($X_{a-b}$) samples from the datasets. If there is a ground truth $Y_a$ for the sample $i$ in the mini-batch, the model $M$ compares the predicted segmentation $Y_a$ with the ground truth $Y_a$ generated by the two encoding subroutines.

\[ Y_a \approx \hat{Y}_a \]

\[ Y_a \approx \hat{Y}_a \]

\[ Y_a \approx \hat{Y}_a \]

\[ Y_a \approx \hat{Y}_a \]
U-Net [9] when using them as the model $M$, resulting in an asymmetrical U-Net to compensate for $G_E$ downsamplings. The amount of input channels of $M$ must also be compatible with the amount of output channels in $G_E$. Another constraint for the architecture of the pair $\{G_E, G_D\}$ is that the upsampling performed by $G_D$ should always compensate the downsampling factor of $G_E$, characterizing $G$ as a whole as a symmetric Encoder-Decoder network.

The discriminator $D$ for CoDAGANs is basically the same as the discriminator from the original Cycle-Consistency network, that is, a basic CNN that classifies between real and fake samples. The only addition to $D$ is conditional training in order for the discriminator to know the domain the sample is supposed to belong to. This allows $D$ to use its marginal distribution for each dataset for determining the likelihood of veracity for the sample. It is important to notice that our model is agnostic to the choice of Unsupervised Image-to-Image Translation architecture, therefore future advances in this area based on Cycle-Consistency should be equally portable to perform DA and further benefit CoDAGAN’s performance.

### A. CONDITIONAL DATASET ENCODING

Conditional dataset training allows CoDAGANs to process data and perform transfer from several distinct source/target datasets. Fully or partially labeled datasets act as source datasets for the method, while unlabeled data is used both to enforce isomorphism in $I$ and to yield adequate image translations between domains. Partially labeled and unlabeled data are, therefore, the target datasets for in this architecture.

While D2D approaches use a coupled architecture composed of 2 encoders ($G_{E_a}$ and $G_{E_b}$) and 2 decoders ($G_{D_a}$ and $G_{D_b}$) for learning a joint distribution over datasets $a$ and $b$, CoDAGANs use only one generator $G$ composed of one encoder and one decoder ($G_E$ and $G_D$). Additionally to the data $X_k$ from some dataset $k$, $G_E$ is conditionally fed a One Hot Encoding $h_k$, as in $I = G_E(X_k, h_k)$. The addition of the data in $X_k$ to the code $h_k$ is achieved by simple concatenation, as shown in Figure 6. The code $h_k$ forces the generator to encode the data according to the marginal distribution optimized for dataset $k$, conditioning the method to the visual style of these data, as exemplified in Figures 5 and 7. The code $h_l$ for a second dataset $l$ is received by the decoder, as in $\hat{X}_{k \rightarrow l} = G_D(I, h_l)$, in order to produce the translation $\hat{X}_{k \rightarrow l}$ to dataset $l$.

### B. TRAINING ROUTINES IN CoDAGANs

In each iteration of a traditional GAN there are two routines for training the networks: 1) freezing the discriminator and updating the generator ($Gen Update$); and 2) freezing the generator and updating the discriminator ($Dis Update$). Performing these routines intermittently allows the networks to converge together in unsupervised settings. CoDAGANs add a new supervised routine to this scheme in order to perform UDA, SSDA and FSDA: $Model Update$. The subroutines described in Section IV that compose the three routines of CoDAGANs are presented in Table 2.

Since the first proposal of GANs [24], stability has been considered a major problem in GAN training. Adversarial training is known to be more susceptible to convergence problems [24], [59] than traditional training procedures for DNNs due to problems as: more complex objectives com-

![FIGURE 6. Illustration of One-Hot-Encoding on image channels in order to encode dataset information.](image-url)
Dis Update is composed of the procedures and b) Training. Training procedure of CoDAGANs into two phases: a) Full Training and finishing with the update of the supervised model. Therefore, in order to achieve more stable results, we split the procedure, during the first 75% of the epochs in a CoDAGAN training on the following paragraphs.

CoDAGANs use One Hot Encoding in order to allow training with more than two domains without scalability hurdles.

During this phase adversarial training enforces the creation of good isomorphic representations by G and translations between the domains. At the same time, the model M uses the existing (and potentially scarce) label information in order to improve the translations performed by G by adding semantic meaning to the translated visual features in the samples.

**h: Supervision Tuning**

The last 25% of the network epochs are trained in the Supervision Tuning setting. This phase removes the unstable adversarial training by freezing G and performing only the Model Update procedure, effectively tuning the supervised model to a stationary isomorphic representation. Freezing G has the effect of removing the instability generated by the adversarial training in the translation process, as it is harder for M to converge properly while the isomorphic input I is constantly changing its visual properties due to changes in the weights of G.

**C. CoDAGAN LOSS**

Both UNIT [29] and MUNIT [30] optimize conjointly GAN-like adversarial loss components and Cycle-Consistency reconstruction losses. Cycle-Consistency losses ($L_{cyc}$) are used in order to provide unsupervised training capabilities to these translation methods, allowing for the use of unpaired image datasets, as paired samples from distinct domains are often hard or impossible to create. Cycle Consistency is often achieved via Variational inference, which tries to find an upper bound to the Maximum Likelihood Estimation (MLE) of high dimensional data [58]. Variational losses allow VAEs to generate new samples learnt from an approximation to the original data distribution as well as reconstruct images from these distributions. Optimizing an upper bound to the MLE allows VAEs to produce samples with high likelihood regarding the original data distribution, but still possessing low visual quality.

Adversarial losses ($L_{adv}$) are often complementarily used with reconstruction losses in order to yield high visual quality and detailed images, as GANs are widely observed to take bigger risks in generating samples than simple regression losses [25]. Simpler approaches to image generation tend to average the possible outcomes of new samples, producing low quality images, therefore GANs produce less blurry and more realistic images than non-adversarial approaches in most settings. Unsupervised Image-to-Image Translation architectures normally use a weighted sum of these previously discussed losses as their total loss function ($L_{tot}$), as in:

$$L_{tot} = \lambda_{cyc} [L_{cyc}(X_a, X_{a\rightarrow b\rightarrow a}) + L_{cyc}(X_b, X_{b\rightarrow a\rightarrow b})] + \lambda_{adv} [L_{adv}(X_b, X_{a\rightarrow b}) + L_{adv}(X_a, X_{b\rightarrow a})].$$ (4)

More details on UNIT and MUNIT loss components can be found in their respective original papers [29], [30]. One should notice that we only presented the architecture-agnostic routines and loss components for CoDAGANs in the previous subsections, as the choice of Unsupervised Image-to-Image Translation basis network might introduce new objective terms and/or architectural changes. MUNIT, for instance, computes reconstruction losses to both the pair of images $\{X_a, X_{a\rightarrow b\rightarrow a}\}$ and the pair of isomorphic representations $\{I_a, I_{a\rightarrow b}\}$, which are separated into style and content components in this architecture.
CoDAGANs add a new supervised component \( L_{sup} \) to the completely unsupervised loss \( L_{tot} \) of Unsupervised Image-to-Image Translation methods. The supervised component for CoDAGANs is the default cost function for supervised classification/segmentation tasks, the Cross-Entropy loss (Equation 1). The full objective \( L_{CoDA} \) for CoDAGANs is, therefore, defined by:

\[
L_{CoDA} = \lambda_{cyc}[L_{cyc}(X_a, X_{a\rightarrow b\rightarrow a}) + L_{cyc}(X_b, X_{b\rightarrow a\rightarrow b})] \\
+ \lambda_{adv}[L_{adv}(X_b, X_{a\rightarrow b}) + L_{adv}(X_a, X_{b\rightarrow a})] \\
+ \lambda_{sup}[L_{sup}(Y_a, M(I_a)) + L_{sup}(Y_b, M(I_b))] \\
+ L_{sup}(Y_a, M(I_{a\rightarrow b})) + L_{sup}(Y_b, M(I_{b\rightarrow a})).
\] (5)

V. EXPERIMENTAL SETUP

All code was implemented using the PyTorch\(^1\) Deep Learning framework. We used the MUNIT/UNIT implementation from Huang et al. [30]\(^2\) as a basis and some segmentation architectures from the pytorch-semantic-segmentation\(^3\) project. All tests were conducted on NVIDIA Titan X Pascal GPUs with 12GB of memory. Our implementation can be found in this project’s website\(^4\).

A. HYPERPARAMETERS

Architectural choices and hyperparameters can be further analysed according to the codes and configuration files in the project’s website, but the main ones are described in the following paragraphs. CoDAGANs were run for 400 epochs in our experiments, as this was empirically found to be a good stopping point for convergence in these networks. Learning rate was set to \(1 \times 10^{-4}\) with L2 normalization by weight decay with value \(1 \times 10^{-5}\) and we used the Adam solver [60]. \( G_{\Theta} \) is composed of two downsampling layers followed by two residual layers for both UNIT [29] and MUNIT [30] based implementations, as these configurations were observed to simultaneously yield satisfactory results and have small GPU memory requirements. The first downsampling layer contains 32 convolutional filters, doubling this number for each subsequent layer. \( D \) was implemented using a Least Squares Generative Adversarial Network (LSGAN) [61] objective with only two layers, although differently from MUNIT, we do not employ multiscale discriminators due to GPU memory constraints. Also distinctly from MUNIT and UNIT, we do not employ the VGG-based [23] perceptual loss – further detailed by Huang et al. [30] – due to the dissimilarities between the domains wherein these networks were pretrained and the biomedical images used in our work.

B. DATASETS

We tested our methodology in a total of 16 datasets, 8 of them being Chest X-Ray (CXR) datasets, 6 of them being Mammographic X-Ray (MXR) datasets and 2 of them being composed of Dental X-Ray (DXR) images. The chosen CXR datasets are the Japanese Society of Radiological Technology (JSRT) [62]\(^5\), OpenIST\(^6\), Shenzhen and Montgomery sets [54]\(^7\), Chest X-Ray 8 [63]\(^8\), PadChest [64]\(^9\), NLM-CXR [65]\(^10\) and the Optical Coherence Tomography and Chest X-Ray Images (OCT CXR) [66]\(^11\) dataset. Label sets for Chest X-Ray 8 were acquired via the XLSor [56] project\(^12\). The MXR datasets used in this work are INbreast [67]\(^13\), the Mammographic Image Analysis Society (MIAS) dataset [68]\(^14\), the Digital Database for Screening Mammography (DDSM) [69]\(^15\), the Breast Cancer Digital Repository (BCDR) [70]\(^16\), and LAPIMO [71]\(^17\). DDSM was split into two groups: 1) samples A, and 2) samples B/C; as these groups were acquired and digitized with different equipments, yielding considerably distinct visual patterns. A random subset of samples from DDSM B/C and DDSM A were manually labeled for the task of pectoral muscle segmentation. These samples were not used during training, but instead for computing objective evaluation metrics during test and can be downloaded for reproducibility in this project’s webpage. At last, the two DXR datasets we used in our experiments are the IvisionLab [72]\(^18\) and the Panoramic X-Ray [73]\(^19\) datasets. A total of 7 distinct segmentation tasks are compared in our experiments: 1) Pectoral muscle, 2) Breast region in MXRs; 3) Lungs, 4) Heart, 5) Clavicles in CXRs; 6) Mandible and 7) Teeth in DXRs. The number of training and testing samples from each domain, dataset and task is available at this project’s webpage.

C. EXPERIMENTAL PROTOCOL

All datasets were randomly split into training and test sets according to an 80%/20% division. Aiming to mimic real-world scenarios wherein the lack of labels is a considerable problem, we did not keep samples for validation purposes. We evaluate results from epochs 360, 370, 380, 390 and 400 for computing the mean and standard deviation values

\(^1\)https://pytorch.org/
\(^2\)https://github.com/nvlabs/MUNIT
\(^3\)https://github.com/zijundeng/pytorch-semantic-segmentation/
\(^4\)http://www.patreo.dcc.ufmg.br/codagans/
\(^5\)http://db.jsrt.or.jp/eng.php
\(^6\)https://github.com/pi-null-mezon/OpenIST
\(^7\)https://lhnbc.nml.nih.gov/publication/pub9931
\(^8\)https://nihcc.app.box.com/v/ChestXray-NIHCC/folder/37178474737
\(^9\)http://bmcv.cipf.es/bmcv-projects/pachest/
\(^10\)https://openi.nlm.nih.gov/
\(^11\)https://data.mendeley.com/datasets/rs cbjbr9sa j/3
\(^12\)https://github.com/rsummers11/CADLab/tree/master/Lung_Segmentation_XLSor
\(^13\)http://medicalresearch.inescporto.pt/breastresearch/index.php/Get_INbreast_Database
\(^14\)https://www.repository.cam.ac.uk/handle/1810/250394
\(^15\)http://marathon.csee.usf.edu/Mammography/Database.html
\(^16\)https://bcdr.eu/patient/list#
\(^17\)http://lapimo.sem.esc.usp.br/bancoweb/english/
\(^18\)https://github.com/IvisionLab/deep-dental-image
\(^19\)https://data.mendeley.com/datasets/hxt48yk462/1
presented in Section VI in order to consider the statistical variability of the methods during the last epochs of the training procedure.

For quantitative assessment we used the Jaccard (Intersection over Union – IoU) metric, which is a common choice in segmentation and detection tasks and is widely used in all tested domains [72], [74], [75]. Jaccard ($J$) for a binary classification task is given by the following equation:

$$J = \frac{TP}{TP + FN + FP},$$

where $TP$, $FN$ and $FP$ refer to True Positives, False Negatives and False Positives, respectively. Jaccard values range between 0 and 1, however we present these metrics as percentages by multiplying them by a factor of 100 in Section VI.

D. BASELINES AND SUPERVISED BACKBONES

Large datasets as ImageNet [13] turned Fine-tuning DNNs into a well known method for Transfer Learning in the Deep Learning literature, as most specific datasets do not possess the large amount of labeled data required for training from scratch in classification tasks. Fine-tuning was later adapted for dense labeling tasks [8] and is nowadays common procedure in semantic segmentation tasks in the Computer Vision domain. However, Fine-tuning still does not work in UDA, as it necessarily requires labeled data. Therefore, we inserted the use of Pretrained DNNs as baselines both without further training in UDA scenarios and as basis for Fine-tuning in SSDA and FSDA scenarios. Still in the field of classical approaches do Transfer Learning, we add as a baseline to our experimental procedure training a DNN From Scratch with the smaller amount of labeled data available for targets datasets in SSDA and FSDA scenarios.

Our main baseline was the D2D approach proposed by Oliveira and dos Santos [21], as it uses a Cycle Consistent GAN with a similar architecture as CoDAGANs. However, instead of performing the supervised learning at the one of the ends of the translation procedure – as most of the literature does [18]–[21] – CoDAGANs optimize the supervised loss using the isomorphic representation as input. Distinctly from Oliveira and dos Santos [21], we employ two separate backbones for D2D: 1) $D2D_M$ that uses the MUNIT [30] architecture as a basis, mixing the source content image with the target style code; and 2) $D2D_U$ that used UNIT [29] as Image-to-Image Translation architecture, without discerning between content and style encodings. One should notice that $D2D_M$ is particularly similar to the method proposed by Yang et al. [22]. Content-only training was further explored by Yang et al. [22] and in early experiments of Oliveira and Santos [21].

Both our method and the chosen baselines use the U-Net [9] as backbone for supervised learning for semantic segmentation. This was a conscious choice based on earlier iterations of this work [21] that compared FCNs [8], U-Nets [9] and SegNets [10] in similar setups and found that U-Nets and SegNets achieved the best results while FCNs generally presented subpar results compared to their Transposed Convolution-based peers. We then narrowed the search due to the larger amount of Skip Connections in this architecture, which mitigates the problem of vanishing gradients by creating backward flow bypasses that help on the training of earlier layers and previous modules with the supervised loss $\mathcal{L}_{sup}$.

VI. RESULTS AND DISCUSSION

Quantitative results presented in this section are divided by domain and type of analysis. Sections VI-A and VI-B presents the UDA, SSDA and FSDA result regarding segmentation of anatomical structures in MXRs and CXRs, respectively. DXRs are evaluated qualitatively, as there is only one labeled dataset per task. In order to mimic the lack of labels in the tasks while still being able to evaluate the performance of our method in UDA and SSDA scenarios, we tested six labels configurations in CXRs and MXRs. Experiments with only source labels (UDA) are referred to as $E_0$ and experiments with the whole range of labels available for training (FSDA) are denominated $E_{100\%}$. Between $E_0$ and $E_{100\%}$, we limited the amount of target labels to 2.5% ($E_{2.5\%}$), 5% ($E_{5\%}$), 10% ($E_{10\%}$) and 50% ($E_{50\%}$), emulating four SSDA scenarios. Each table in Sections VI-A and VI-B attribute one uppercase letter for each dataset, so that they can be more easily be referenced during the discussion.

Qualitative analysis in both unlabeled and labeled data are presented in Section VI-C for all domains. Section VI-D discusses the activations of channels in isomorphic representations, where supervised learning is performed and Domain Generalization is enforced. At last, Section VI-E discusses the distributions of samples across different domains and datasets computed from the isomorphic space $I$.

A. QUANTITATIVE RESULTS FOR MXR SAMPLES

Jaccard average values and standard deviations for MXR tasks are shown in Tables 3 and 4 for pectoral muscle and breast region segmentation, respectively. The first lines in the tables present the label configurations used in the experiments at each column. Results are shown separately for datasets INbreast (A), MIAS (B), DDSM B/C (C), and DDSM A (D) in Table 3 and for datasets INbreast (A) and MIAS (B) in Table 4. Objective results for datasets BCDR (E) and LAPIMO (F) for pectoral muscle and datasets (C)-(F) in breast region segmentation are not possible due to the complete lack of labels in these tasks. We reinforce that only two CoDAGANs ($CoDA_M$ using MUNIT [30] and $CoDA_U$ based on UNIT [29]) were trained for all datasets in each task, as CoDAGANs allow for multi-source and multi-target DA. Thus, repeated columns indicating the results for $CoDA_M$ and $CoDA_U$ are simply reporting the results of the same models for different datasets. All methods besides $CoDA_M$ and $CoDA_U$ indicate whether the source or target data used in the training, as they are neither multi-source nor multi-target, limiting them to pairwise training.
TABLE 3. Jaccard results (in %) for pectoral muscle segmentation DA to and/or from six distinct MXR datasets: INbreast (A), MIAS (B), DDSM B/C (C), DDSM A (D), BCDR (E) and LAPIMO (F). This table shows results for CoDAGANs with backbones based on MUNIT [30] (CoDA\textsubscript{M} and UNIT (CoDA\textsubscript{U})), as well as Domain-to-Domain approaches based on these architectures (D\textsuperscript{2}D\textsubscript{M} and D\textsuperscript{2}D\textsubscript{U}). Pretrained U-Nets [9] and U-Nets trained from scratch on the limited target labels.

| Experiments                  | $E_{0\%}$ | $E_{2.5\%}$ | $E_{5\%}$ | $E_{10\%}$ | $E_{50\%}$ | $E_{100\%}$ |
|-----------------------------|-----------|-------------|-----------|------------|------------|-------------|
| % Labels INbreast (A)       | 100.00%   | 100.00%     | 100.00%   | 100.00%    | 100.00%    | 100.00%     |
| % Labels MIAS (B)           | 100.00%   | 100.00%     | 5.00%     | 10.00%     | 50.00%     | 100.00%     |
| % Labels DDSM B/C (C)       | 0.00%     | 2.50%       | 0.00%     | 0.00%      | 0.00%      | 0.00%       |
| % Labels DDSM A (D)         | 0.00%     | 0.00%       | 0.00%     | 0.00%      | 0.00%      | 0.00%       |
| % Labels BCDR (E)           | 0.00%     | 0.00%       | 0.00%     | 0.00%      | 0.00%      | 0.00%       |
| % Labels LAPIMO (F)         | 0.00%     | 0.00%       | 0.00%     | 0.00%      | 0.00%      | 0.00%       |

1) PECTORAL MUSCLE SEGMENTATION IN MXR IMAGES

For the completely unlabeled case $E_{0\%}$ in pectoral muscle segmentation, CoDA\textsubscript{M} and CoDA\textsubscript{U} achieved $J$ values of 67.61% and 60.01% for the MIAS target dataset, while the best baseline achieved 41.06%. SSDA and FSDA experiments ($E_{2.5\%}$ to $E_{100\%}$) regarding the MIAS dataset show that CoDAGANs achieve considerably better results than all baselines in all but one case. These results evidenced the higher instability of training pairwise translation architectures compared to conditional training. Across the training procedure, Jaccard values for D2D fluctuated by several percentage units, yielding standard deviations of one magnitude or more larger than CoDAGANs.

In the case of pectoral muscle for DDSM B/C (C), UDA using CoDA\textsubscript{M} and CoDA\textsubscript{U} achieved 89.99% and 82.45%, with the D2D baseline achieving worse than random results, evidencing its lack of capability to translate between domains (A) and (C). The best baseline in this case was simply the use of Pretrained DNNs in (A) and testing on (C), which achieved 78.22%. Segmentation results for DDSM A (D) were considerably worse for all methods and experiments,
as samples from this subset of images showed an extremely lower contrast compared to the samples of DDSM B/C. Even in this suboptimal case, CoDAGANs achieved much better results than the baseline in UDA. Preprocessing using adaptive histogram equalization in DDSM A (C) samples might improve results, although more empirical evidence is required. As there were only few samples labeled from (C) and (D), only UDA was possible for these datasets in D2D and pretrained baselines, as all labels were kept for testing. However, one can easily see that experiments $E_{2.5\%}$ to $E_{100\%}$ show better results in (C) and (D) as the number of labels from (B) increases, achieving a $J$ of 79.08% with all (B) labels being used in training. This is due to two factors: 1) the larger number of labels achieved with the combination of (A) and (B); and 2) the more similar visual patterns between (B)$\to$(C) and (B)$\to$(D). Similarly to DDSM (C)$\to$(D), MIAS (B) is an older originally analog that was later digitized, while INbreast (A) is a Full Field Digital Mammography (FFDM) dataset.

3) MXR SEGMENTATION CONFIDENCE INTERVALS

Figure 8 show the $J$ values from Tables 3 and 4 with confidence intervals for $p \leq 0.05$ using a t-Student distribution. A first noticeable trait in Figures 8a and 8e is that CoDAGANs maintained their capability to perform inference on the INbreast source dataset for both pectoral muscle and breast region experiments when labels from other sources are added to the procedure. D2D tends to get more unstable when the plots get closer to FSDA ($E_{100\%}$) due to the incongruities in labeling styles from the different datasets. Figures 8b, 8c and 8d clearly show that CoDAGANs outperforms all baselines in UDA for the MIAS (B), DDSM B/C (C) and DDSM A (D) datasets by a large margin for pectoral muscle segmentation. All of these discrepancies between CoDAGANs and baselines are statistically significant, showing a clear superiority of CoDAGANs in UDA scenarios in this task. Another important result is that Figure 8d shows a clear increase in the performance of CoDA$_M$ on dataset (D) when more labels from dataset (B) were allowed to be used – that is, in results close to fully pretrained models with no Fine-tuning already achieved a $J$ value of 75.53%. This easier DA task also benefits from the higher capability of U-Nets to segment details using skip connections between symmetric layers. As CoDAGANs remove the first layers of U-Net’s Encoder to fit the smaller spatial dimensions of the isomorphic representation, the last layers of the network do not receive skip connections from the first layers, allowing for fine object details to be lost. This can be seen as a compromise between generalization capability and fine segmentation details.

### TABLE 4.

| Experiments | $E_{0\%}$ | $E_{2.5\%}$ | $E_{5\%}$ | $E_{10\%}$ | $E_{50\%}$ | $E_{100\%}$ |
|-------------|----------|------------|----------|-----------|----------|-----------|
| % Labels INbreast (A) | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| % Labels MIAS (B) | 0.00% | 2.50% | 5.00% | 10.00% | 50.00% | 100.00% |
| % Labels DDSM BC (C) | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| % Labels DDSM A (D) | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| % Labels BCDR (E) | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| % Labels LAPIMO (F) | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |

CoDA$_M$ | 98.69 ± 0.06 | 98.48 ± 0.15 | 98.59 ± 0.09 | 97.98 ± 0.36 | 98.27 ± 0.41 | 98.11 ± 0.20 |
CoDA$_U$ | 98.29 ± 0.13 | 98.12 ± 0.10 | 98.37 ± 0.15 | 97.79 ± 0.64 | 97.89 ± 0.27 | 98.04 ± 0.18 |
D2D$_M$ (A)$\to$(B) | 98.90 ± 0.09 | 98.93 ± 0.16 | 98.27 ± 0.45 | 98.36 ± 0.82 | 97.36 ± 1.60 | 85.89 ± 7.74 |
D2D$_U$ (A)$\to$(B) | 98.74 ± 0.14 | 98.26 ± 0.31 | 98.92 ± 0.19 | 98.65 ± 0.09 | 97.80 ± 0.52 | 95.01 ± 1.51 |
D2D$_M$ (A)$\to$(C) | 99.00 ± 0.08 | - | - | - | - |
D2D$_U$ (A)$\to$(C) | 98.90 ± 0.09 | - | - | - | - |
D2D$_M$ (A)$\to$(D) | 98.87 ± 0.15 | - | - | - | - |
D2D$_U$ (A)$\to$(D) | 98.83 ± 0.14 | - | - | - | - |
D2D$_M$ (A)$\to$(E) | 99.02 ± 0.05 | - | - | - | - |
D2D$_U$ (A)$\to$(E) | 98.90 ± 0.05 | - | - | - | - |
D2D$_M$ (A)$\to$(F) | 98.11 ± 1.65 | - | - | - | - |
D2D$_U$ (A)$\to$(F) | 98.69 ± 0.15 | - | - | - | - |
From Scratch in (A) | 98.75 ± 0.11 | - | - | - | - |

CoDA$_M$ | 68.96 ± 1.57 | 88.13 ± 3.81 | 91.97 ± 0.68 | 93.11 ± 2.01 | 96.53 ± 0.45 | 97.19 ± 0.10 |
CoDA$_U$ | 69.72 ± 0.30 | 90.12 ± 0.60 | 91.97 ± 0.61 | 95.28 ± 0.25 | 95.85 ± 0.50 | 97.21 ± 0.15 |
D2D$_M$ (A)$\to$(B) | 5.02 ± 2.51 | 69.26 ± 14.74 | 5.01 ± 2.64 | 13.78 ± 3.77 | 59.20 ± 20.80 | 58.70 ± 29.26 |
D2D$_U$ (A)$\to$(B) | 9.63 ± 2.49 | 15.72 ± 2.49 | 66.02 ± 26.19 | 90.70 ± 1.82 | 95.93 ± 0.77 | 80.00 ± 13.50 |
Pretrained (A)$\to$(B) | 75.53 | 91.94 ± 0.28 | 93.21 ± 0.35 | 94.06 ± 1.66 | 96.64 ± 0.13 | 97.40 ± 0.08 |
From Scratch in (B) | - | 91.38 ± 0.28 | 93.27 ± 0.25 | 95.69 ± 0.22 | 97.10 ± 0.25 | 97.37 ± 0.15 |

VOLUME 8, 2020
supervised learning with labels from (B). Figure 8f shows the UDA, SSDA and FSDA results for CoDAGANs and baselines on the target MIAS (B) dataset in the task of breast region segmentation. CoDAGANs yield considerably higher results than D2D, even though a Pretrained U-Net surpassed all methods in this task for UDA. Domain shifts between the MIAS and INbreast datasets are probably considerably small. Pretrained U-Nets might not be universally better than CoDAGANs in UDA, though, as it is usually unable to compensate for large domain shifts. This trend was shown in Figures 8b, 8c and 8d and will be further reinforced in Section VI-B.

B. QUANTITATIVE RESULTS FOR CXR SAMPLES

CXR results can be seen in Tables 5 and 6 for lungs, heart and clavicle segmentations. The JSRT (A), OpenIST (B), Shenzhen (C), Montgomery (D) and Chest X-Ray 8 (E) datasets are objectively evaluated in the lung field segmentation task, as shown in Table 5, while PadChest (F), NLMCXR (G) and OCT CXR (H) do not possess pixel-level ground truths for quantitative assessment. In heart and clavicle segmentation, apart from the source JSRT (A) dataset, only OpenIST (B) contains a subset of 15 labeled samples for these two task. Therefore, we reserved the labeled samples for testing and trained on the remaining samples for UDA quantitative assessment, as shown in Table 6. Analogously to Section VI-A, bold values in Tables 5 and 6 represent the best overall results in a given label configuration for a specific dataset.

1) LUNG SEGMENTATION IN CXR IMAGES

In the task of lung segmentation in CXRs (Table 5), baselines showed considerably poor results for target datasets (B)-(D) in UDA experiments. Following the results from Sections VI-A.1 and VI-A.2, D2D with a small amount of target labels proved to be highly unstable, yielding worse results and considerably higher standard deviations, when compared with CoDAGANs. CoDA_M and CoDA_U achieve the best UDA results in (B), (C) and (D), surpassing all baselines by

FIGURE 8. Confidence Intervals for MXRs according to the values shown in Tables 3 and 4. Methods shown in these plots include CoDAGANs (CoDA_M and CoDA_U), Domain-To-Domain translation (D2D_M and D2D_U), pretrained U-Nets and U-Nets trained from scratch for SSDA and FSDA.
TABLE 5. Jaccard results (in %) for lung field segmentation DA to and/or from eight distinct CXR datasets: JSRT (A), OpenIST (B), Shenzhen (C), Montgomery (D), Chest X-ray 8 (E), PadChest (F), NLMCXR (G) and OCT CXR (H). This table shows results for CoDAGANs with backbones based on MUNIT [30], CoDA\(_M\) and CoDA\(_U\), as well as Domain-to-Domain approaches based on these architectures (D2D\(_M\) and D2D\(_U\)), Pretrained U-Nets [9] and U-Nets trained from scratch on the limited target labels.

| Experiments       | \(E_{0.0\%}\) | \(E_{2.5\%}\) | \(E_{5.0\%}\) | \(E_{10.0\%}\) | \(E_{50.0\%}\) | \(E_{100.0\%}\) |
|-------------------|----------------|---------------|---------------|----------------|----------------|-----------------|
| % Labels JSRT (A) | 95.27 ± 0.07   | 94.08 ± 0.44  | 94.41 ± 0.86  | 94.74 ± 0.12  | 94.87 ± 0.39  | 95.31 ± 0.17   |
| % Labels OpenIST (B) | 95.55 ± 0.07   | 95.16 ± 0.12  | 94.77 ± 0.17  | 94.83 ± 0.08  | 94.56 ± 0.99  | 94.97 ± 0.62   |
| % Labels Shenzhen (C) | 96.16 ± 0.35   | 95.44 ± 0.16  | 95.33 ± 0.04  | 95.26 ± 0.17  | 95.22 ± 0.29  | 95.20 ± 0.19   |
| % Labels Montgomery (D) | 96.45 ± 0.02   | 96.44 ± 0.05  | 96.07 ± 0.61  | 96.43 ± 0.10  | 96.29 ± 0.07  | 96.20 ± 0.10   |
| % Labels ChestX-Ray\(8\) (E) | 96.10 ± 0.51   | 96.34 ± 0.03  | 96.04 ± 0.35  | 96.33 ± 0.07  | 96.23 ± 0.04  | 96.35 ± 0.06   |
| % Labels PadChest (F) | 96.23 ± 0.21   | 96.22 ± 0.08  | 96.20 ± 0.09  | 96.24 ± 0.09  | 96.16 ± 0.21  | 95.84 ± 0.70   |
| % Labels NLMCXR (G) | 96.26 ± 0.11   | 96.21 ± 0.06  | 96.22 ± 0.13  | 96.21 ± 0.17  | 96.22 ± 0.10  | 96.02 ± 0.13   |
| % Labels OCT CXR (H) | 96.35 ± 0.19   | 96.42 ± 0.09  | 96.38 ± 0.15  | 96.38 ± 0.15  | 96.38 ± 0.15  | 96.38 ± 0.15   |

(A) CoDAGANs maintain state-of-the-art results in SSDA experiments with small amount of labels, surpassing baselines in most datasets for \(E_{5.0\%}\) and \(E_{10.0\%}\). In \(E_{50.0\%}\) and \(E_{100.0\%}\) state-of-the-art results are achieved mainly by From Scratch training in the target domain due to label abundance. Similarly, D2D methods are only able to achieve stable results, after \(E_{10.0\%}\). As in MXRs, D2D underperformed in UDA settings compared to CoDAGANs, even though it presented considerably better results than Pretrained DNNs.

We also show that the source dataset presented little to no deterioration in segmentation quality when segmented by CoDAGANs compared to D2D and From Scratch training on (A). D2D from translations (A)→(B) to (A)→(H) present remarkably similar results in UDA, SSDA and FSDA, achieving state-of-the-art results in all cases. It is noticeable that CoDAGANs achieved no superiority in the source domain, as it aims for generalization and does not focus in fine-grained...
segmentation. However, the difference of Jaccard values between CoDAGANs and baseline methods that only consider a pair of domains or even only the source domain (From Scratch) remained limited to between 1% and 2%.

2) HEART AND CLAVICLE SEGMENTATION IN CXR IMAGES

As shown in Table 6, heart and clavicle segmentation proved to be harder tasks than lung field segmentation. Both tasks only count with the JSRT dataset as fully labeled, with OpenIST having only 15 images with pixel-level annotations for both heart and clavicles. We therefore used these samples only for evaluating UDA in a target dataset, as the small number of samples would not allow for proper SSDA and FSDA. We consider a pair of domains or even only the source domain (From Scratch) remained limited to between 1% and 2%.

Table 6. Jaccard results (in %) for heart and clavicle segmentation DA to and/or from eight distinct CXR datasets: JSRT (A), OpenIST (B), Shenzhen (C), Montgomery (D), Chest X-ray 8 (E), PadChest (F), NLMCXR (G) and OCT CXR (H). This table shows results for CoDAGANs with backbones based on MUNIT [30] CoDA (M) and UNIT (CoDA (U)), as well as Domain-to-Domain approaches based on these architectures (D2DM and D2DU). Pretrained U-Nets [9] and U-Nets trained from scratch on the limited target labels.

| Experiments | E0.5% (Heart) | E0.5% (Clavicles) |
|-------------|--------------|-----------------|
| % Labels JSRT (A) | 100.00% | 100.00% |
| % Labels OpenIST (B) | 0.00% | 0.00% |
| % Labels Shenzhen (C) | 0.00% | 0.00% |
| % Labels Montgomery (D) | 0.00% | 0.00% |
| % Labels Chest X-ray8 (E) | 0.00% | 0.00% |
| % Labels PadChest (F) | 0.00% | 0.00% |
| % Labels NLMCXR (G) | 0.00% | 0.00% |
| % Labels OCT CXR (H) | 0.00% | 0.00% |

(A) CoDA (M) 89.86 ± 0.29 77.31 ± 0.37
CoDA (U) 89.89 ± 0.32 76.03 ± 1.06
D2DM (A) → (B) 90.68 ± 0.19 87.76 ± 0.18
D2DU (A) → (B) 90.97 ± 0.10 87.96 ± 0.14
D2DM (A) → (C) 91.16 ± 0.18 87.20 ± 0.22
D2DU (A) → (C) 90.70 ± 0.14 87.09 ± 0.61
D2DM (A) → (D) 90.65 ± 0.17 84.78 ± 0.33
D2DU (A) → (D) 90.67 ± 0.23 84.46 ± 1.24
D2DM (A) → (E) 90.97 ± 0.20 87.38 ± 0.21
D2DU (A) → (E) 91.11 ± 0.12 87.02 ± 0.63
D2DM (A) → (F) 89.74 ± 2.32 87.65 ± 0.35
D2DU (A) → (F) 90.63 ± 0.37 87.39 ± 0.47
D2DM (A) → (G) 91.15 ± 0.18 87.59 ± 0.69
D2DU (A) → (G) 90.90 ± 0.10 87.37 ± 0.59
D2DM (A) → (H) 91.33 ± 0.16 86.46 ± 0.62
D2DU (A) → (H) 90.22 ± 0.94 84.49 ± 0.64
From Scratch in (A) 88.91 ± 0.48 76.07 ± 0.41

(B) CoDA (M) 64.63 ± 1.28 61.94 ± 1.03
CoDA (U) 63.71 ± 0.95 57.59 ± 1.26
D2DM (A) → (B) 54.91 ± 2.35 57.11 ± 0.96
D2DU (A) → (B) 54.50 ± 0.84 48.53 ± 0.89
Pretrained (A) → (B) 0.0 ± 0.0 0.24 ± 0.40

Image Translation does not compromise performance in the source domain. CoDAGANs closely followed the performance of D2D in the source dataset (A) for heart segmentation, but again showed considerably worse performance in clavicle segmentation. This underperformance of CoDAGANs in clavicle segmentation for both datasets is probably explained by the higher imbalance of this task. Clavicles cover a much smaller area in a CXR than lungs or a heart, and therefore, are more susceptible to low performance in segmentation DNNs that contain fewer skip connections, as the case of the truncated asymmetrical U-Net configured to receive data from the isomorphic representation I in CoDAGANs.

3) CXR SEGMENTATION CONFIDENCE INTERVALS

Figure 9 shows the confidence intervals for p ≤ 0.05 in lung segmentation for both the source JSRT dataset (Figure 9a) and the target image sets (Figures 9b, 9c, 9d and 9e). Figures 9f and 9g show the results for heart and clavicle segmentation respectively.

One can see by Figures 9a and 9f that segmentation in the source dataset is preserved even when labels from other datasets are introduced in the training procedure. Figures 9b, 9c, 9d and 9g show the UDA, SSDA and FSDA efficiency of CoDAGANs in the fully or partially labeled target datasets, that is, OpenIST, Shenzhen, Montgomery and Chest X-Ray 8 for lung segmentation and only OpenIST for heart and clavicles.

Figures 9b through 9g show the progression of CoDAGAN and baseline methods in distinct target CXR datasets according to the different label configurations in our experimental procedure. While most methods converge to similar efficiencies in scenarios closer to FSDA (E50% and E100%), baselines start considerably worse than CoDAGANs in most cases where there is scarcity of target labels (E0% and E2.5%). D2DM and D2DU also yield highly unstable predictions in scenarios between these two extremities (E5% and E10%), with much larger confidence intervals resulting from larger standard deviations than their counterparts.

Another interesting phenomenon can be seen in Figure 9e, where CoDAGANs start worse in UDA than both D2DM and D2DU for target samples from dataset (E), but CoDA (M) improves as the experiments get closer to E100%. One should notice that in experiments E2.5% through E100% no labels from (E) are being used at any time during the training procedure of CoDAGANs, and even still the objective evaluation for dataset (E) improves. This serves as yet another evidence that CoDAGANs are able to acquire semantic information for one dataset (Chest X-Ray 8) by using labels from others; in this case, JSRT, OpenIST, Montgomery and Shenzhen.

C. SEGMENTATION QUALITATIVE ANALYSIS

Figures 10, 11 and 12 show segmentation qualitative results for two tasks of mammographic image segmentation, two tasks of DXR segmentation and three tasks of...
FIGURE 9. Confidence Intervals for CXRs according to the values shown in Tables 5 and 6. Methods shown in these plots include CoDAGANs (CoDA\textsubscript{M} and CoDA\textsubscript{U}), Domain-To-Domain translation (D\textsuperscript{2}D\textsubscript{M} and D\textsuperscript{2}D\textsubscript{U}), pretrained U-Nets and U-Nets trained from scratch for SSDA and FSDA.
CXR segmentation, respectively. Figure 10a presents predictions for pectoral muscle segmentation $E_{0\%}$, while Figure 10b shows breast region segmentation on experiment $E_{0\%}$. Experiment $E_{0\%}$ for lung field segmentation can be seen in Figure 12a, while heart and clavicle segmentation DA experiments ($E_{0\%}$) are shown respectively on Figures 12b and 12c. At last, DXR segmentations from UDA experiments can be seen in Figure 11 for both teeth (Figure 11a) and mandible (Figure 11b) segmentation. Columns for all figures present the original sample, the ground truth segmentation for the specific task for this sample when available and predictions from Pretrained U-Nets, D2D and CoDAGANs for visual comparison.

Each row in Figures 10a and 10b highlights one sample from each one of the six MXR datasets used in our experiments. One can see in both figures that D2D underperformed in most cases, failing to predict any pectoral muscle pixel as positive in multiple samples from target datasets. UDA for breast region segmentation also proved to be a hard task for D2D, as in most samples it segmented either only the pectoral muscle or background. While in the pectoral muscle segmentation task most methods were able to successfully ignore the labels in the background of some digitized datasets such as DDSM, MIAS and LAPIMO, these artifacts were shown to be harder to compensate for on breast region segmentation, as all baselines and CoDAGANs wrongly and
FIGURE 12. Qualitative segmentation results in CXR images for three distinct tasks: $E_{\theta}\%$ lungs (a), $E_{\theta}\%$ heart (b) and $E_{\theta}\%$ clavicles (c).
frequently segmented them as part of the breast. We observed overwhelmingly better results in our qualitative assessment from CoDAGANs, when compared to all other baselines. CoDAGAN superiority proved to be stable both in easier target datasets such as MIAS or BCDR and in more difficult ones as DDSM A and LAPIMO, which contain extremely low contrast and large digitization artifacts, respectively. At last, as the breast boundary contour is fuzzy and extremely hard to segment even for humans in non-FFDM datasets, all methods either underrepresent or overrepresent positive breast pixels in these regions in most samples and datasets.

Figure 11a shows teeth segmentation predictions for both source (IvisioLab) and target (Panoramic X-ray) datasets, while Figure 11b presents DXR mandible segmentations using Panoramic X-ray as source and IvisioLab as target. DXR results show that Pretrained U-Nets and D2D, as expected from a supervised setting, yield mostly predictions in the source dataset for both tasks. However, both methods underperform in the target datasets, missing the segmentation of several teeth and mislabeling mandible regions as background. CoDAGANs achieve much more consistent results in the target datasets, once again evidencing the method’s capabilities in UDA. However, CoDAGAN predictions were observed to be less robust for modeling sharp corners in the shapes probably due to the smaller spatial resolution of representation $I$ when compared to the images themselves, which may lead to loss of small detail and slightly smoother shape contours. This issue might be fixed by passing the outputs of the encoder layers in $G_{\beta}$ to the supervised model $M$ in order to preserve spatial information, much like a skip connection does.

Figure 12a shows DA results for lung field segmentation in 4 fully labeled datasets (JSRT, OpenIST, Shenzhen and Montgomery), 1 partially labeled dataset (Chest X-Ray 8) and 3 other target unlabeled datasets (PadChest, NLMCXR and OCT CXR). We reiterate that one single CoDAGAN was trained for all datasets and made all predictions contained in the last column of Figure 12a. One should notice that the target datasets in this case are considerably harder than the source ones due to poor image contrast, presence of unforeseen artifacts as pacemakers, rotation and scale differences and a much wider variety of lung sizes, shapes and health conditions. Yet, the DA procedure using CoDAGANs for lung segmentation was adequate for the vast majority of images, only presenting errors in distinctly difficult images. As the source dataset (JSRT) has completely distinct visual patterns when compared to the target datasets, both Pretrained U-Nets and D2D are not able to properly compensate for domain shift in these cases, yielding grossly wrong predictions.

Heart and clavicle segmentation (Figures 12b and 12c) are harder tasks than lung segmentation due to heart boundary fuzziness and a high variability of clavicle sizes, shapes and positions. In addition, clavicle segmentation is a highly unbalanced task. Those factors, paired with the fact that the well-behaved samples from the JSRT dataset are the only source of labels to this task contributed to higher segmentation error rates mainly in clavicle segmentation. Results for heart and clavicles are presented for the same 8 datasets as lung segmentation, but only a small subset of OpenIST contains labels for clavicles and heart. Even with all these hampers, CoDAGANs still yielded consistent prediction maps for hearts and clavicles across all target datasets, while baselines are, again, unable to compensate for domain shifts.

Figure 13 presents a visual assessment of segmentation errors in CXR (Figure 13a), DXR (13b) and MXR (Figure 13c) tasks for some samples of target datasets in UDA scenarios. A full assessment of results and both CoDAGAN and baseline errors can be seen in this project’s webpage.

One can see that several lung predictions by CoDAGANs yielded small isles of false positives in other bony areas of CXRs (Figure 13) as well as in the background of the images due to wrongly compensated domain shifts. While most of these errors can be corrected by simply filtering for keeping only the larger contiguous areas lung field segmentation and heart segmentation, this would be harder to implement for clavicles due to their smaller relative sizes in CXR exams. Extremely low contrast images as the NLMCXR sample presented in the fourth row of Figure 13 presented a challenge for CoDAGANs on all CXR tasks, being the most common source of missed predictions for our method.

We noticed that there were large inter-dataset labeling differences for all CXR tasks. For instance, several OpenIST heart labels contain larger heart delineations than JSRT labels, which led to a larger number of false negatives on OpenIST, as can be seen in the fifth row of Figure 13. Also, clavicle labels on JSRT delineate only pixels within lung borders, while OpenIST labels delineate the whole pair of bones both inside and outside the lung fields. We employed a binary mask between clavicles and lungs for each labeled OpenIST sample in order to fix this discrepancy in labeling characteristics.

Even though both Pretrained U-Nets and D2D yielded worse general results in DXR tasks, CoDAGANs still missed a considerable number of teeth, failed to separate the upper and lower dental arches and wrongfully split mandibles, as shown in Figure 13b. At last, MXR prediction errors can be seen in Figure 13c, mainly in denser breasts, which hamper the differentiation between pectoral muscle and breast tissue and due to fuzzy breast-boundary borders. Some of the non-FFDM datasets also contain digitization artifacts in the background, which were frequently misclassified as breast pixels. Therefore, there is still a lot of room for improvement in CoDAGAN’s domain shift compensation capabilities.

D. QUALITATIVE ANALYSIS OF ISOMORPHIC REPRESENTATIONS

Another important qualitative assessment to be performed in CoDAGANs is to visually assess that the same objects in distinct datasets are represented similarly in $I$-space. This is shown in Figure 14 for three different activation channels in MXRs (Figure 14a), DXRs (Figure 14b) and CXRs (Figure 14c) from distinct datasets.
In Figure 14a, high density tissue patterns and important object contours in the images from INbreast, MIAS, DDSM BC, DDSM A, BCDR and LAPIMO are encoded similarly by CoDAGANs. Breast boundaries are also visually similar across samples from all mammographic datasets, as CoDAGANs are able to infer that this information is semantically similar despite the differences in the visual patterns of the images. Visual patterns that compose the patient’s anatomical structures, such as ribs and lung contours, in Figure 14c are visibly similar in the samples from all eight CXR datasets: JSRT, OpenIST, Shenzhen, Montgomery, Chest X-Ray 8, PadChest, NLMCXR and OCT CXR. The third radiological domain used in our comparisons is composed of two different DXRs datasets: IvisionLab and Panoramic X-Ray (Figure 14b). It is easy to notice the common patterns encoded by CoDAGANs for the same semantic areas of the distinct images such as the teeth edges and mandible contours. One should notice that despite the clear visual distinctions between the original samples from the different datasets in all domains, the isomorphic representations were visually alike across samples from the domains. These results show that CoDAGANs successfully create a joint representation for high semantic-level information which encodes analogous visual patterns across datasets in a similar manner. In other words, different convolutional channels in $I$ activate visual patterns with the same semantic information from the distinct datasets in a similar manner. This feature of encoding a joint distribution between domains by looking only to the marginal distributions of the samples is what allows CoDAGANs to perform UDA, SSDA and FSDA with high accuracy.

**E. LOW DIMENSIONALITY**

In order to view the data distributions of samples from the different datasets in the $I$-space of CoDAGAN representations, we reduced the dimensionality of $I$ to a 2D visualization using Principal Component Analysis (PCA) and the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm [76]. First, in order to reduce computational requirements, we reduced the original 524, 288 dimensions of $I$ to a 200-dimensional space using PCA and applied t-SNE on the remaining components. We also fit a gaussian on the data distributions for each dataset using the Gaussian Mixture Model (GMM) from sklearn\(^\text{20}\). The resulting 2D

\(^{20}\text{https://scikit-learn.org/stable/}
visualizations of the mammography and CXR datasets can be seen in Figure 15. Figures 15a and 15b show respectively the original 2D representation of the $I$-space from PCA/t-SNE and the GMM.
FIGURE 15. 2D projections of in \( I \)-space for MXRs in pectoral muscle segmentation (a, b) and CXRs in the lung segmentation task (c, d). The original 2D-projected samples after PCA/t-SNE (a, c) are shown conjointly with gaussian fits over the data (b, d) for both domains.

Not coincidentally, these artifacts in LAPIMO samples hampered CoDAGAN abilities to compensate for domain shift and severely hampered the segmentation quality in all baselines.

A similar pattern can be seen in Figures 15c and 15d, which show respectively the 2D projections of CXR datasets and the GMM fits for these data. JSRT samples have the most standardized data among all CXR datasets, containing only high visual quality samples with fixed posture, high contrasts between anatomical structures (i.e. lungs, ribs, etc) and no major lung shape-distorting illnesses (i.e. pneumonia, tuberculosis, etc). Other datasets – such as Chest X-Ray 8, Montgomery and Shenzhen – present more real-world scenarios with a high variety of lung shapes and sizes and smaller control over patient’s position during the exam, that is, higher rotation, scale and translations in these images. Thus samples from the JSRT dataset in Figure 15d are clustered in a small region in the 2D projection of \( I \)-space, while the other datasets contain more spread samples in this projection. This result evidences that the use of distinct sources of data should better enforce satisfactory Domain Generalization for the supervised model \( M \) in CoDAGANs.

Another visibly distinct cluster in Figure 15d is formed of samples pertaining to the OCT CXR set. Samples from this dataset were noticeably harder to segment due to their
smaller contrast range. OCT CXR patients also performed the exam on a distinct position with their arms pointing upward, contrary to all other CXR data used in our experiments. These visual features reinforce this dataset’s distinction from other CXRs in our experiments and explain its homogeneity in the 2D projections of Figure 15d.

Another use for these 2D projections could be to perform inference from datasets that were never trained by the algorithm, effectively achieving Domain Generalization [12] for new samples. This Domain Generalization CoDAGAN could find the natural cluster closer to the new data according to a dissimilarity metric and assign the novel samples to the cluster. This approach could, therefore, personalize the One-Hot-Encoding so that it better captures the particular visual patterns of previously unseen data.

**VII. CONCLUSION**

This paper proposed and validated a method that covers the whole spectrum of UDA, SSDA and FSDA in dense labeling tasks for multiple source and target biomedical datasets. We performed an extensive quantitative and qualitative experimental evaluation on several distinct domains, datasets and tasks, comparing CoDAGANs with both traditional and recent baselines in the DA literature. CoDAGANs were shown to be an effective DA method that could learn a single model that performs satisfactory predictions for several different datasets in a domain, even when the visual patterns of these data were clearly distinct. The proposed method was able to gather both labeled and unlabeled data in its inference process, making it highly adaptable to a wide variety of data scarcity scenarios in SSDA.

We showed that CoDAGANs can be built upon two distinct Unsupervised Image-to-Image Translation methods (UNIT [29] and MUNIT [30]), evidencing its agnosticism to the underlying image translation architecture. It is also evident in both our background comparisons (Section III) and in our experimental evaluation (Section VI) that CoDAGANs are distinct from simpler D2D approaches, which are recurrent in the literature of image translation for DA [15], [18], [21], [47], [48]. One possible explanation for the inferior performance of D2D is the lack of Domain Generalization – that is, the lack of multi-source information being incorporated to the model resulting from the limitations of pairwise training. Another important remark should be noticed regarding the performance of pairwise approaches and CoDAGANs. While being multi-source and multi-target allows for one single CoDAGAN to be trained to perform inference over multiple domains, D2D approaches can only try to achieve this generality by choosing one source domain – usually the one with the larger number of labels – and training multiple distinct models to each target. Therefore, CoDAGANs are considerably more efficient for multi-target DA. Time comparisons for experiments in Section VI can be seen in the Supplementary Material uploaded at the project’s webpage.

It was observed in Sections VI-A and VI-B that CoDAGANs achieve results in fully unsupervised settings that are comparable to supervised DA methods – such as Fine-tuning to new data. These experiments also showed that both Pre-trained DNNs and D2D approaches were ineffective in scarce labeling scenarios. CoDAGANs presented significantly better Jaccard values in most experiments where labeled data was scarce in the large variety of target datasets studied. Fine-tuning and From Scratch training were only able to achieve good objective results when labeled data was abundant – that is, in scenarios closer do $E_{100\%}$. It is important to reiterate that label scarcity is a major problem in real world biomedical image tasks, mainly for dense labeling tasks.

CoDAGANs were observed to perform satisfactory DA even when the labeled source dataset was considerably simpler than the target unlabeled datasets, as presented in Section VI. In experiment $E_{0\%}$ for CXR lung, clavicle and heart segmentations, the JSRT source dataset contains images acquired in a much more controlled environment than all of the target datasets. In addition, experiment $E_{0\%}$ only used INbreast samples for training and was able to perform DA for more real-world scenario datasets, such as DDSM, BCDR and LAPIMO, albeit with some segmentation artifacts in many samples due to poor variability in training data. Another evidence of the capabilities of CoDAGANs is the good performance in DA tasks even for highly imbalanced classes, as the case of clavicle segmentation, wherein the Region of Interest in the images represents only a tiny portion of the pixels.

One should notice that CoDAGANs, despite being tested only for segmentation tasks in this paper, are not conceptually limited to dense labeling tasks nor to biomedical images. One of the main future works for CoDAGANs is comprised of testing their UDA, SSDA and FSDA in sparse labeling tasks, such as classification and regression and other kinds of dense labeling tasks, such as detection. We also intend to test CoDAGANs in other image domains, such as traditional Computer Vision datasets, Remote Sensing data and other kinds of biomedical images, as samples from Magnetic Resonance Imaging (MRI), Computerized Tomography (CT scans) and Positron Emission Tomography (PET scans). At last, CoDAGANs, due to their ability to transfer knowledge between several source and target datasets, can be adapted for Domain Generalization, that is, when there are neither labels nor data for the target domain to train the DA algorithm. This shall also be explored in future iterations of this work.

**ACKNOWLEDGMENT**

The authors would like to thank NVIDIA for the donation of the GPUs that allowed the execution of all experiments in this article.

**REFERENCES**

[1] S. Masood, M. Sharif, A. Masood, M. Yasmin, and M. Raza, “A survey on medical image segmentation,” *Current Med. Imag. Rev.*, vol. 11, no. 1, pp. 3–14, Apr. 2015.

[2] A. Elnakib, G. Gimel’farb, J. S. Suri, and A. El-Baz, “Medical image segmentation: A brief survey,” in *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*. New York, NY, USA: Springer, 2011, pp. 1–39.
S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for data: Ground

D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang,

D. Demner-Fushman, M. D. Kohli, R. Chen, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in Proc. IEEE Conf. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 3213–3223.

S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wäng, P.-X. Lu, and G. Thoma, “Two public chest X-ray datasets for computer-aided screening of pulmonary diseases,” Quant. Imag. Med. Surg., vol. 4, no. 4, p. 475, 2014.

Y. Zhang, S. Miao, T. Mansi, and R. Liao, “Task driven generative modeling for unsupervised domain adaptation: Application to X-ray image segmentation,” in Proc. Int. Conf. Med. Image Comput. Assist. Intervent. Cham, Switzerland: Springer, 2018, pp. 599–607.

Y. Tang, Y. Tang, J. Xiao, and R. M. Summers, “XLSor: A robust and accurate lung segmentor on chest X-rays using Criss-cross attention and customized radiorealistic abnormalities generation,” in Proc. Int. Conf. Med. Image Comput. Deep Learn., 2019, pp. 457–467.

Y. Tang, Y. Tang, V. Sandfort, J. Xiao, and R. M. Summers, “TUNA-Net: Task-oriented USupervised adversarial network for disease recognition in cross-domain chest X-rays,” in Proc. Int. Conf. Med. Image Comput. Assist. Intervent. Cham, Switzerland: Springer, 2019, pp. 431–440.

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

X. Yao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “Least squares generative adversarial networks,” in Proc. IEEE Int. Conf. Vis. Comput. Vis. (ICCV), Oct. 2017, pp. 2813–2821.

J. Shinrai, S. Katsuragawa, J. Ikekze, T. Matsumoto, T. Kobayashi, K.-I. Komatsu, M. Matsui, H. Fujita, Y. Kodera, and K. Doi, “Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of Radiologists’ detection of pulmonary nodules,” Amer. J. Roentgenol., vol. 174, no. 1, pp. 71–74, Jan. 2000.

X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “ChestX-ray8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3462–3471.

A. Bustos, A. Pertusa, J.-M. Salinas, and M. de la Iglesia-Vayá, “Pad-Chest: A large chest X-ray image dataset with multi-label annotated reports,” 2019, arXiv:1901.07441. [Online]. Available: http://arxiv.org/abs/1901.07441

D. Demner-Fushman, M. D. Kohli, M. B. Rosenman, S. E. Shooshan, L. Rodriguez, S. Antani, G. R. Thoma, and C. J. McDonald, “Preparing a collection of radiology examinations for distribution and retrieval,” J. Amer. Med. Inform. Assoc., vol. 23, no. 2, pp. 304–310, Mar. 2016.

D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, S. L. Baxter, A. McKeeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. L. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, and K. Zhang, “Identifying medical diagnoses and treatable diseases by image-based deep learning,” Cell, vol. 172, no. 5, pp. 1122–1131, Feb. 2018.

J. C. Moreira, I. Amaral, I. Domingues, A. Cardoso, M. J. Cardoso, and J. S. Cardoso, “INbreast: Toward a full-field digital mammographic database,” Academic Radiol., vol. 19, no. 2, pp. 236–248, 2012.

J. Suckling, J. Parker, D. Dance, S. Astley, I. Hutt, C. Boggis, I. Ricketts, E. Stamatakis, N. Cerneau, and S. Kok, “Mammographic Image Analysis Society (MIAS) database v1. 21,” Mammographic Image Anal. Soc. Tech. Rep., 2015. [Online]. Available: https://www.repository.cam.ac.uk/handle/1810/250394

M. Heath, K. Bowyer, D. Kopans, R. Moore, and W. P. Kegelmeyer, “The database of digital mammography,” in Proc. Int. Workshop Digit. Mammography, Jun. 2000, pp. 212–218.

M. G. Lopez, N. Posada, D. C. Moura, R. R. Pollán, J. M. F. Valiente, C. S. Ortega, M. Solar, G. Diaz-Herrero, I. Ramos, and J. Loureiro, “BCDR: A breast cancer digital repository,” in Proc. 15th Int. Conf. Exp. Mech., 2012, pp. 1–5.

B. R. N. Matheus and H. Schiabel, “Online mammographic images database for development and comparison of CAD schemes,” J. Digit. Imag., vol. 24, no. 3, pp. 500–506, Jun. 2011.

G. Silva, L. Oliveira, and M. Pithon, “Automatic segmenting teeth in X-ray images: Trends, a novel data set, benchmarking and future perspectives,” Expert Syst. Appl., vol. 107, pp. 15–31, Oct. 2018.

A. H. Abdi, S. Kassaei, and M. Mehdizadeh, “Automatic segmentation of mandible in panoramic X-ray,” J. Med. Imag., vol. 2, no. 4, Nov. 2015, Art. no. 044003.

A. Rampun, P. J. Morrow, B. W. Scottney, and J. Winder, “Fully automated breast boundary and pectoral muscle segmentation in mammograms,” Artif. Intell. Med., vol. 79, pp. 28–41, Jun. 2017.

B. van Ginneken, M. B. Stegmann, and M. Loog, “Segmentation of anatomical structures in chest radiographs using supervised methods: A comparative study on a public database,” Med. Image Anal., vol. 10, no. 1, pp. 19–40, Feb. 2006.

L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, pp. 2579–2605, Nov. 2008. 

HUGO N. OLIVEIRA received the B.Sc. and M.Sc. degrees in computer science from the Universidade Federal da Paraíba (UFPB), 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree in computer science with the Universidade Federal de Minas Gerais (UFMG). He is also a Professor at the Technical College (COLTEC/UFMG) and a member of the Pattern Recognition and Earth Observation (PATREO) Laboratory. His research interests include machine learning, deep learning, biomedical images, domain adaptation, open set recognition, meta-learning, few-shot learning, data compression, and information theory.

EDEMIR FERREIRA received the B.Sc. degree in computational mathematics and the M.Sc. degree in computer science from the Universidade Federal de Minas Gerais, in 2013 and 2016, respectively. He has experience in geoscience with a focus on remote sensing problems.

JEFESSON A. DOS SANTOS received the Ph.D. degree in computer science from the Université de Cergy-Pontoise, Cergy, France, and the University of Campinas, Campinas, Brazil, in 2013. He is currently an Adjunct Professor with the Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil. He is the Founder and a Coordinator of the Laboratory of Pattern Recognition and Earth Observation, PATREO. He has published several articles in journals with high impact factor and selective editorial policy. He has also published articles in important conferences of image processing and computer vision. He has experience in coordinating research with Brazilian funding agencies and research and development projects with companies in those topics. He has been holding the Research Productivity Grant from the Brazilian Research Council (CNPq), since 2016.