GraphEcho: Graph-Driven Unsupervised Domain Adaptation for Echocardiogram Video Segmentation

Jiewen Yang[1] Xinpeng Ding[1] Ziyang Zheng[1†] Xiaowei Xu[2*] Xiaomeng Li[1*]
Hong Kong University of Science and Technology[1]
Guangdong Provincial People's Hospital, Institute of Cardiovascular Diseases, GuangZhou, China[2]
{jyangcu, xdingaf}@connect.ust.hk, xiao.Wei.xu@foxmail.com, eexmli@ust.hk

Abstract

Echocardiogram video segmentation plays an important role in cardiac disease diagnosis. This paper studies the unsupervised domain adaption (UDA) for echocardiogram video segmentation, where the goal is to generalize the model trained on the source domain to other unlabelled target domains. Existing UDA segmentation methods are not suitable for this task because they do not model local information and the cyclical consistency of heartbeat. In this paper, we introduce a newly collected ***CardiacUDA dataset*** *and a novel* ***GraphEcho*** *method for cardiac structure segmentation. Our GraphEcho comprises two innovative modules, the Spatial-wise Cross-domain Graph Matching (SCGM) and the Temporal Cycle Consistency (TCC) module, which utilize prior knowledge of echocardiogram videos, i.e., consistent cardiac structure across patients and centers and the heartbeat cyclical consistency, respectively. These two modules can better align global and local features from source and target domains, leading to improved UDA segmentation results. Experimental results showed that our GraphEcho outperforms existing state-of-the-art UDA segmentation methods. Our collected dataset and code will be publicly released upon acceptance. This work will lay a new and solid cornerstone for cardiac structure segmentation from echocardiogram videos. Code and dataset are available at : https://github.com/xmed-lab/GraphEcho*

1. Introduction

Echocardiography is a non-invasive diagnostic tool that enables the observation of all the structures of the heart. It can capture dynamic information on cardiac motion and function [33, 11, 20], making it a safe and cost-effective option for cardiac morphological and functional analysis. Accurate segmentation of cardiac structure, such as Left Ventricle (LV), Right Ventricle (RV), Left Atrium (LA), and Right Atrium (RA), is crucial for determining essential cardiac functional parameters, such as ejection fraction and myocardial strain. These parameters can assist physicians in identifying heart diseases, planning treatments, and monitoring progress [12, 32]. Therefore, the development of an automated structure segmentation method for echocardiogram videos is of great significance. Nonetheless, a model trained using data obtained from a specific medical institution may not perform as effectively on data collected from other institutions. For example, when a model trained on site G is directly tested on site R, its performance can significantly decrease to 48.5% Dice, which is significantly lower than the performance of a model trained directly on site R, which achieves 81.3% Dice; see results of *Without DA* and *Upper Bound* in Table 2. The result indicates that there are clear domain gaps between echocardiogram videos collected on different sites; see (c-d) in Figure 2. Therefore, it is highly desirable to develop an unsupervised domain adaptation (UDA) method for cardiac structure segmentation from echocardiogram videos.

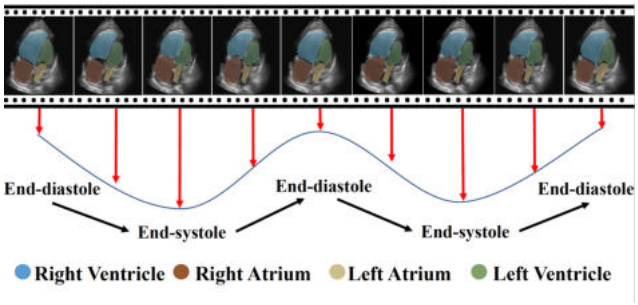
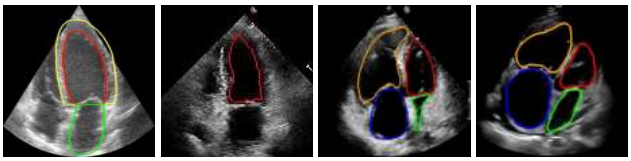


Figure 1. Examples of nine frames from our newly collected **CardiacUDA** dataset, which serves as a new domain adaptation benchmark for cardiac structure segmentation from echocardiogram videos.

To the best of our knowledge, the UDA segmentation for echocardiogram videos has not been explored yet, and the most intuitive way is to adapt existing UDA methods designed for natural image segmentation and medi-

*Corresponding author
†Interning at The Hong Kong University of Science and Technology



(a) CAMUS (b) Echonet (c) Site G (d) Site R

Figure 2. (a-b) two public datasets; (c-d) our newly collected CardiacUDA from two sites: G and R. Red, green, blue, orange and yellow refer to the segmentation contours for left ventricle (LV), left atrium (RA), right Atrium (RA), right ventricle (RV), and epicardium of left ventricle, respectively. Table 1 outlines the advantages of our dataset over CAMUS and Echonet.

Table 1. The comparison of CardiacUDA, CAMUS, and Echonet. †: 5 frames are labelled for each video in the training set, and all frames are labelled for each video in the validation and test dataset.

Dataset	**Our CardiacUDA**	CAMUS [22]	EchoNet [32]
Video Num.	992	500	10,030
Frames Num.	102,796	10,000	1,755,250
Train/Test Labels	5 frames / Full †	2 frames / 2 frames	2 frames / 2 frames
Multiple Centers	✓	×	×
Cardiac Views	4	1	1
Resolution	720p	480p	120p
Annotated Regions	LV, RV, LA, RA	LV, LA	LV

cal image segmentation to our task. In general, existing methods can be grouped into 1). the image-level alignment methods [21, 29, 43, 37] that focus on aligning the style difference to minimize the domain gaps, such as PLCA [21], PixMatch [29] and Fourier-base UDA [43, 37]; 2). feature-level alignment methods [19, 23, 24], such as [23], use global class-wise alignment to reduce the discrepancy between source and target domains. However, applying these methods directly to cardiac structure segmentation in echocardiogram videos generated unsatisfactory performance; see results in Table 2 and Figure 5. We thus consider two possible reasons: (1) Existing UDA methods [23, 21, 29, 43, 37] primarily focused on aligning the global representations between the source and target domain while neglecting local information, such as LV, RV, LA, and RA; see (c-d) in Figure 2. The failure to model local information during adaptation leads to restricted cardiac structure segmentation results. (2) Most existing methods [16, 23, 21, 29, 43, 37, 39, 44, 41] were mainly designed for 2D or 3D images, which does not consider the video sequences and the cyclic properties of the cardiac cycle in our task. Given that heartbeat is a periodically recurring process, it is essential to ensure that the extracted features exhibit cyclical consistency [7].

To address the above limitations, we present a novel graph-driven UDA method, namely **GraphEcho**, for echocardiogram video segmentation. Our proposed GraphEcho consists of two novel designs: ***(1) Spatial-wise Cross-domain Graph Matching (SCGM)*** module and ***(2) Temporal Cycle Consistency (TCC)*** module. SCGM is motivated by the fact that the structure/positions of the different cardiac structures are similar across different patients and domains. For example, the left ventricle's appearance is typically visually alike across different patients; see red contours in Figure 2. Our SCGM approach reframes domain alignment as a fine-grained graph-matching process that aligns both class-specific representations (local information) and the relationships between different classes (global information). By doing so, we can simultaneously improve intra-class coherence and inter-class distinctiveness.

Our TCC module is inspired by the observation the recorded echocardiogram videos exhibit cyclical consistency; see examples in Figure 1. Specifically, our TCC module utilizes a series of recursive graph convolutional cells to model the temporal relationships between graphs across frames, generating a global temporal graph representation for each patient. We then utilized a contrastive objective that brings together representations from the same video while pushing away those from different videos, thereby enhancing temporal discrimination. By integrating SCGM and TCC, our proposed method can leverage prior knowledge in echocardiogram videos to enhance inter-class differences and intra-class similarities across source and target domains while preserving temporal cyclical consistency, leading to a better UDA segmentation result.

In addition, we collect a new dataset, called **CardiacUDA** from two clinical centers. As shown in Table 1, compared to existing publicly available echocardiogram video datasets [22, 32], our new dataset has higher resolutions, greater numbers of annotations, more annotated structure types as well as more scanning views. Our contribution can be summarized as follows:

- We will publicly release a newly collected echocardiogram video dataset, which can serve as a new benchmark dataset for video-based cardiac structure segmentation.
- We propose GraphEcho for cardiac structure segmentation, which incorporates a novel SCGM module and a novel TCC module that are motivated by prior knowledge. These modules effectively enhance both inter-class differences and intra-class similarities while preserving temporal cyclical consistency, resulting in superior UDA results.
- GraphEcho achieved superior performance compared to state-of-the-art UDA methods in both the computer vision and medical image analysis domains.

2. Related Work

2.1. UDA for Segmentation

In this section, we review the existing UDA segmentation methods for natural and medical images separately.

Natural image segmentation. For natural image segmentation, the adversarial-based domain adaptation methods [14, 29, 34, 36, 46] and multi-stage self-training methods, including single stage [5, 26, 28, 40, 47] and multi-stage [23, 24] are the most commonly used training methods. The adversarial method aims to align the distributions and reduce the discrepancy of source and target domains through the Generative Adversarial Networks (GAN) [15] framework. At the same time, the self-training generate and update pseudo label online during training, such as applying data augmentation or domain mix-up.

Medical image segmentation. For medical image segmentation, the UDA segmentation methods can be classified into image-level [1, 39] that use GANs and different types of data augmentation to transfer source domain data to the target domain, and feature-level methods [16, 41], such as feature alignment methods that aim to learn domain-invariant features across domains.

While existing methods tend to overlook the temporal consistency characteristics in heartbeat cycles and the local relationships between different chambers across domains, our proposed GraphEcho method effectively learns both inter-class differences and intra-class coherence while preserving temporal consistency. This leads to superior UDA segmentation results.

2.2. Graph Neural Networks

Graph neural networks (GNNs) have the ability to construct graphical representations to describe irregular objects of data [17]. Also, graphs can iteratively aggregate the knowledge based on the broadcasting of their neighbouring nodes in the graph, which is more flexible for constructing the relationship among different components [17]. The learned graph representations can be used in various downstream tasks, such as classification [17], object detection [25], vision-language [2], etc. Specifically, ViG [17] models an image as a graph and uses GNN to extract high-level features for image classification. Li *et al*. [25] apply graphical representation instead of the feature space to explore multiple long-range contextual patterns from the different scales for more accurate object detection. GOT [2] leverages the graphs to conduct the vision and language alignment for image-text retrieval. There also exist some works that use the graph to conduct cross-domain alignment for object detection [25] and classification [10, 27]. However, these methods only capture the global graph information for images, which is insufficient for video segmentation tasks. In this paper, our proposed GraphEcho learns both local class-wise and temporal-wise graph representations, which can reduce the domain gap in a fine-grained approach and enhance temporal consistency, leading to an enhanced result.

3. Method

As shown in Figure 3, our method consists of three main components. First, a basic segmentation network is used to extract features and obtain prediction masks for both source and target domain data (See Section 3.1). Then, a **S**patial-wise **C**ross-domain **G**raph **M**atching (SCGM) module and a **T**emporal-wise **C**ycle **C**onsistency (TCC) module are designed to reduce the domain gap in both spatial and temporal wise for echocardiogram videos (See Section 3.2 and (See Section 3.3)).

3.1. Basic Segmentation Network

In our UDA echocardiogram segmentation, we denote the source and target domain data as $\{\mathcal{X}^s, \mathcal{Y}^s\}$ and $\mathcal{X}^t$, respectively, where $\mathcal{X}^s$ is the video set in the source domain, and $\mathcal{Y}^s$ is its corresponding label set. Note that the videos set in the target domain $\mathcal{X}^t$ is without the label. For clarity, we sample a video frame with the label $\{\mathbf{x}^s, \mathbf{y}^s\}$ from an example $\{\mathbf{X}^s, \mathbf{Y}^s\}$ of the source domain data, where $\mathbf{X}^s \in \mathcal{X}^s$ is a video from $\mathcal{X}^s$ and $\mathbf{Y}^s \in \mathcal{Y}^s$ is its corresponding label. Similarly, we can also sample a video frame from the target domain, *i.e*., $\mathbf{x}^t$.

The basic segmentation network consists of a feature extractor and a decoder. We first feed the $\mathbf{x}^s$ or $\mathbf{x}^t$ to the feature extractor to obtain $\mathbf{f}^s$ or $\mathbf{f}^s$, followed by a decoder that maps the features $\mathbf{f}^s$ or $\mathbf{f}^t$ to the corresponding prediction mask, *i.e*., $\hat{\mathbf{y}}^s$ or $\hat{\mathbf{y}}^t$. Then, we use the segmentation loss to supervise the model on the pixel classification task with the annotated source domain data as follows:

$$\mathcal{L}_{seg} = \mathcal{L}_{bce}(\hat{\mathbf{y}}^s, \mathbf{y}^s) + \mathcal{L}_{dice}(\hat{\mathbf{y}}^s, \mathbf{y}^s), \tag{1}$$

where $\mathcal{L}_{bce}$ and $\mathcal{L}_{dice}$ are the binary cross-entropy loss and dice loss [30].

3.2. Spatial-wise Cross-domain Graph Matching

In this section, we introduce the spatial-wise cross-domain graph matching (SCGM), which aligns both class-wise representations and their relations across the source and target domains. To this end, we use the graph to model each echocardiogram frame, where the nodes represent the different chambers and the edges illustrate the relations between them. Compared with the convolution neural network, the graph can better construct the relations among different classes explicitly [17].

In the following, we first illustrate how to convert the features of source and target domains *i.e.*, $\mathbf{f}^s$ and $\mathbf{f}^s$ to the corresponding graph representation, which is defined as $\mathbf{g}^s$ and $\mathbf{g}^t$ respectively. After that, we introduce the graph matching method to align the generated graph to reduce the domain gap.

Graph construction. The graph construction aims to convert visual features to graphs. Since the construction process of the source domain and the target domain is the same,

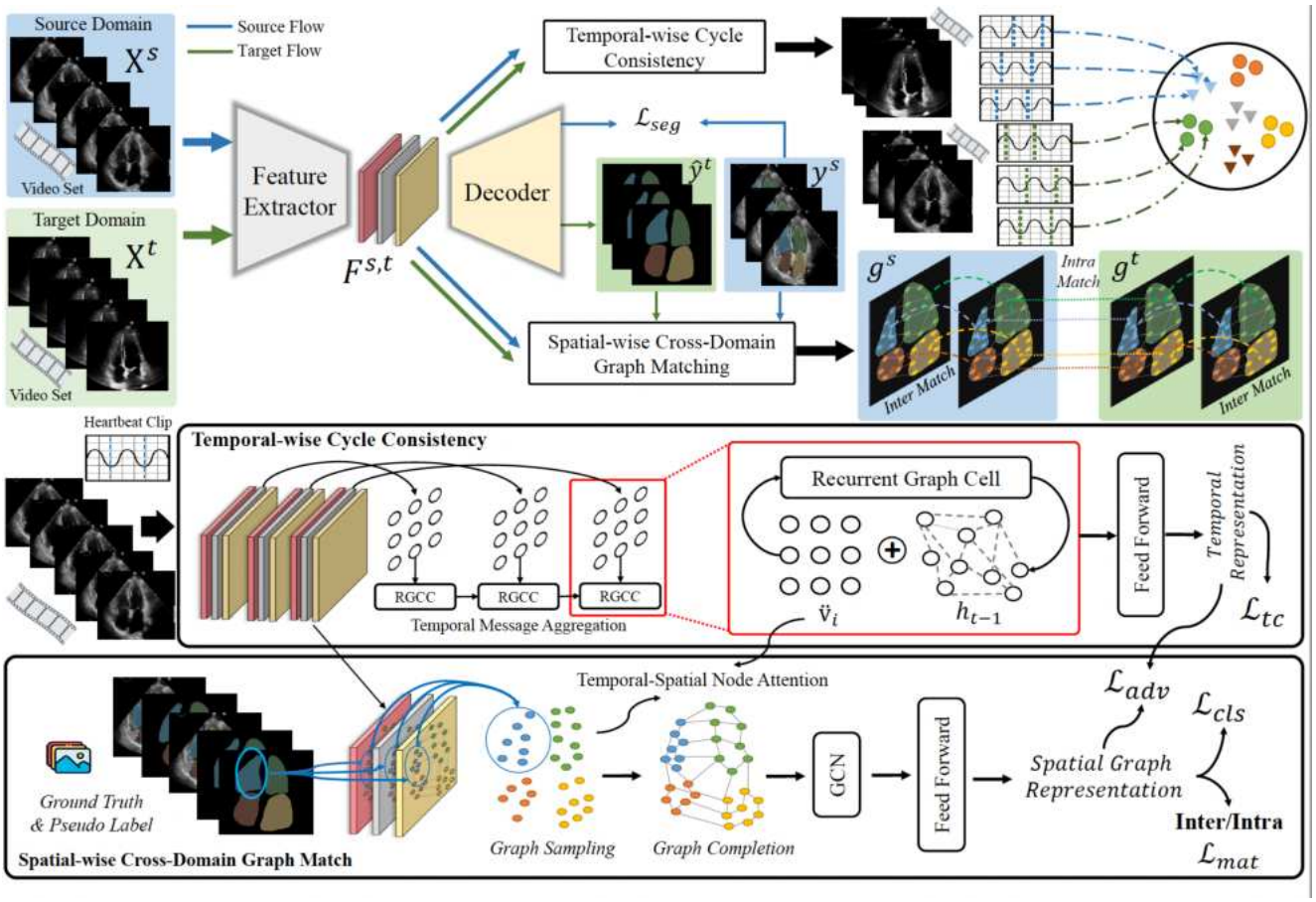


Figure 3. **The overview of our GraphEcho framework.** The source/target videos are fed into a shared backbone, generate feature maps, and produce the segmentation result from the decoder. Spatially, we extract feature nodes from each feature map based on their corresponding ground truth and pseudo label and, subsequently, construct complete semantic graphs suitable for intra-domain and inter-domain matching of cardiac structures. Temporally, we aggregate temporal messages by means of recurrent graph cells, thereby clustering heartbeat representations and enforcing temporal cycle consistency. Additionally, our method incorporates the temporal-spatial node attention module to establish a connection between the spatial and temporal domains. Finally, the trained feature extractor and decoder are used for inference and generating the final result.

we take the source domain for illustration. Formally, given the feature $\mathbf{f}^s$ and its corresponding pseudo labels $\hat{\mathbf{y}}^s$ for a video frame (see definitions in Section 3.1), we conduct graph sampling to sample the graph nodes from $\mathbf{f}^s$ based on $\hat{\mathbf{y}}^s$, as shown in Figure 3. Specifically, we first use the pseudo labels $\hat{\mathbf{y}}^s$ to segment $\mathbf{f}^s$ into different chamber regions, *i.e.*, $\{\mathbf{f}_i^s\}$. Then, in each chamber region, we uniformly sample m feature vectors fed into a projection layer to obtain the node embedding $\mathbf{v}^s$. Based on $\mathbf{v}^s$, we define the edge connections $\mathbf{e}^s$, which is a learned matrix. Finally, the constructed semantic graph can be defined as $g^s = \{\mathbf{v}^s, \mathbf{e}^s\}$. In this same way, we can also obtain the semantic graph for the target domain, *i.e.*, $g^t = \{\mathbf{v}^t, \mathbf{e}^t\}$.

Graph matching. We leverage graph matching to perform the alignment of the source and target domain graph g^s and g^t, thus reducing the domain gap. Since graph matching is an optimization problem for g^s and g^t, the relations between the two graphs are essential for the optimal solution [2]. Hence, we use the self-attention [35, 8] to capture the intra- and inter-domain relations between the source and target graph nodes, *i.e.*, $\mathbf{v}^s$ and $\mathbf{v}^t$, which can be formulated as $\bar{\mathbf{v}}^s, \bar{\mathbf{v}}^t = Attention(concat(\mathbf{v}^t, \mathbf{v}^s))$, where $concat$ indicates the concatenation. To ensure the generated graph nodes are classified into the correct classes, we introduce the classification loss as follows:

$$\mathcal{L}_{cls} = -\alpha \mathbf{y} log(h(\bar{\mathbf{v}}^s)) - \beta \hat{\mathbf{y}} log(h(\bar{\mathbf{v}}^t)), \qquad (2)$$

where h is the classifier head followed by a softmax, and α, β are the weights for two domains.

Then, graph matching can be implemented by maximising the similarity of graphs (including nodes and edges) belonging to the same class but from two different domains. Specifically, we first obtain the adjacency matrix $\mathbf{A}$ from g^s and g^t following to represent the relations of graph nodes. Then, the maximizing process can be transferred into opti-

mizing the transport distance of $\mathbf{A}$. To this end, we use the Sinkhorn algorithm [6] to obtain the transport cost matrix of graphs among chambers, defined as $\vec{\mathbf{A}} = Sinkhorn(\mathbf{A})$. Then, our optimization target can be formulated as follows:

$$\mathcal{L}_{mat} = \sum_{p,q} \mathbb{I}(\mathbf{y}_p^s = \hat{\mathbf{y}}_q^t) \cdot min(\vec{\mathbf{A}}(p,q)) + \mathbb{I}(\mathbf{y}_p^s \neq \hat{\mathbf{y}}_q^t) \cdot max(\vec{\mathbf{A}}(p,q)), \tag{3}$$

where $\mathbf{A}(p,q)$ is the p-th row and q-th column element on $\mathbf{A}$, $\mathbb{I}(\cdot)$ is the indicator function. Eq. 3 aims to minimize the distance between samples of the same class across different domains while increasing the distance between samples of different classes across domains, thus eliminating the influence of domain shift. Finally, $\mathcal{L}_{SCGM} = \mathcal{L}_{cls} + \mathcal{L}_{mat}$ is the overall loss of module SCGM.

3.3. Temporal-wise Cycle Consistency

In this section, we propose the **T**emporal **C**ycle **C**onsistency (TCC) module to enhance the temporal graphic representation learning across frames, by leveraging the temporal morphology of echocardiograms, *i.e.*, the discriminative heart cycle pattern across different patients. The proposed TCC consists of three parts: a temporal graph node construction to generate a sequence of temporal graph nodes for each video; a recursive graph convolutional cell to learn the global graph representations for each video; a temporal consistency loss to enhance the intra-video similarity and reduce the inter-video similarity. Note that TCC is applied to both source and target domains; we take the source domain for clarity.

Temporal graph node construction. Given a video $\mathbf{X}^s$, we defines its features for frames as $\{\mathbf{f}_i^s\}_{i=1}^N$, where $\mathbf{f}_i$ is the feature of the i-th frame and N is the number of frames in $\mathbf{X}^s$. Considering the computation cost, we use an average pooling layer to compress the size of $\{\mathbf{f}_i^s\}_{i=1}^N$. For each compressed feature $\mathbf{f}_i^s$, we flatten it and treat its pixels as graphical nodes, *i.e.*, $\ddot{\mathbf{v}}_i^s$. Then, the temporal graph nodes for the video $\mathbf{X}^s$ can be defined as $\{\ddot{\mathbf{v}}_i^s\}_{i=1}^N$.

Recursive graph convolutional cell. Inspired by [42], we propose the recursive graph convolutional cell to aggregate the semantics of the temporal graph nodes $\{\ddot{\mathbf{v}}_i^s\}_{i=1}^N$ for obtaining the global temporal representation of each video. For the p-th node $\ddot{\mathbf{v}}_i^s(p)$ at $\ddot{\mathbf{v}}_i^s$, we find its K nearest neighbors $\mathcal{N}(p)$ on the hidden state $\mathbf{h}_t$ [1], where $\mathcal{N}(p) \in \mathbf{h}_t$. Then the edge $\ddot{\mathbf{e}}_i^s(q,p)$ can be added directed from $\mathbf{h}_t(q)$ to $\ddot{\mathbf{v}}_i^s(p)$ for all $\mathbf{h}_t(q) \in \mathcal{N}(p)$. After obtaining the edge $\ddot{\mathbf{e}}_i^s$ for $\ddot{\mathbf{v}}_i^s$, the message broadcast from the i-th graph to the $i+1$-th graph can be defined as follows:

$$\mathbf{h}_{t+1} = \sigma \mathbf{w}_{gcn}(\ddot{\mathbf{v}}_i^s, \ddot{\mathbf{e}}_i^s) + \mathbf{b}_{gcn}, \tag{4}$$

where the σ indicates the activation function, $\mathbf{w}_{gcn}$ and $\mathbf{b}_{gcn}$ are the graph convolution weight and bias, respectively. We conduct this message broadcast for $\{\ddot{\mathbf{v}}_i^s, \ddot{\mathbf{e}}_i^s\}_{i=1}^N$, and obtain the final hidden state $\mathbf{h}_N$. The global representation for video $\mathbf{X}^s$ is the $\mathbf{o}^s$, obtained by $\mathbf{o}^s = FFN(\mathbf{h}_N)$, where FFN is a feed forward network. Hence, the whole process of recursive graph convolutional cell can be formulated as $\mathbf{o}^s = RGCC(\mathbf{X}^s)$. Similarly, we can obtain the temporal representation for the target video $\mathbf{X}^t$ by $\mathbf{o}^t = RGCC(\mathbf{X}^t)$.

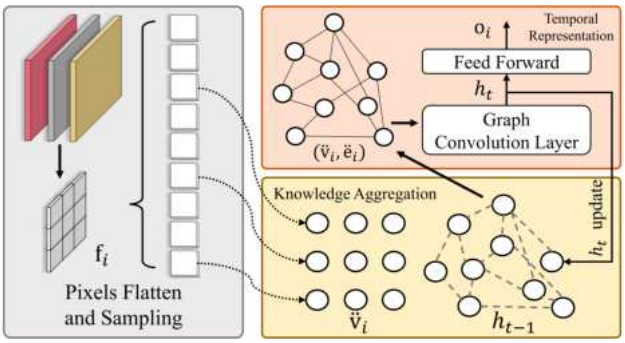


Figure 4. This figure illustrates the workflow of the Recursive Graph Convolutional Cell (RGCC), which receives the current input $\ddot{\mathbf{v}}_i^s$ and the hidden reference graph h_{t-1}. Complete the knowledge aggregation operation through the K-nearest neighbour algorithm that builds the edge $\ddot{\mathbf{e}}_i^s$ for $\ddot{\mathbf{v}}_i^s$. Finally, a graph convolution layer for nodes representation $\bar{x}_t$ calculation.

Temporal consistency loss. For better representation learning, we leverage temporal consistency loss to make features from the same video similar and features from different videos dissimilar. In this paper, we use contrastive learning [18, 9], a mainstream method to pull close the positive pairs and push away negative ones, to achieve this goal. We regard the two consequent clips $\mathbf{X}_k^s$ and $\mathbf{X}_+^s$ that randomly sampled from a video $\mathbf{X}^s$ as positive pairs. Then, we feed the positive clips to the recursive graph convolutional cell to obtain the global representations, *i.e.*, $\mathbf{o}_k^s = RGCC(\mathbf{X}_k^s)$ and $\mathbf{o}_+^s = RGCC(\mathbf{X}_+^s)$. For negative pairs, we maintain a memory bank $\mathcal{B}$ consisting of representations of clips sampled from different videos. Then, the temporal consistency loss for the source domain is defined as follows:

$$\mathcal{L}_{tc}^s = -\sum_{\{\mathbf{o}_k^s, \mathbf{o}_+^s\} \in \mathcal{P}^s} log \frac{\exp(\mathbf{o}_k^s \cdot \mathbf{o}_+^s)}{\sum_{\mathbf{o}_-^s \in \mathcal{B}} \exp(\mathbf{o}_k^s \cdot \mathbf{o}_-^s)}, \tag{5}$$

where $\mathcal{P}^s$ is the set of positive pairs. We here use the dot product to measure the similarity and use InfoNCE [31] as the specific contrastive learning objective. Similarly, we can define the temporal consistency loss for the target domain as $\mathcal{L}_{tc}^t$, and the total temporal consistency loss is $\mathcal{L}_{tc} = \mathcal{L}_{tc}^s + \mathcal{L}_{tc}^t$.

Since $\mathcal{L}_{tc}$ is applied to two domains independently, a gap between source and target domains still exists for the learned global representation, *i.e.*, $\mathbf{o}^s$ or $\mathbf{o}^t$. Hence, we leverage the adversarial methods [13] to eliminate the gap between $\mathbf{o}^s$ and $\mathbf{o}^t$, which can be formulated as $\mathcal{L}_{adv}$. The

[1] h_t is the learned parameters and the initial hidden state h_0 is all zero.

overall loss of temporal consistency is $\mathcal{L}_{TCC} = \mathcal{L}_{tc} + \mathcal{L}_{adv}$, where $\mathcal{L}_{adv}$ is the global domain-adversarial loss in our TCC module. To summarize, the final loss of GraphEcho is $\mathcal{L}_{All} = \mathcal{L}_{SCGM} + \mathcal{L}_{TCC} + \mathcal{L}_{seg}$, and the network is trained in end-to-end.

4. Experiments

4.1. Datasets.

We evaluate our method on three datasets, including our collected dataset (CardiacUDA) and two public datasets (CAMUS [22] and Echonet Dynamic [32]). Table 1 shows the dataset details.

CardiacUDA. We collect CardiacUDA from our two collaborating hospitals: site G and site R. In order to guarantee all echocardiogram videos are standards-compliant, all cases of CardiacUDA are collected, annotated and approved by 5-6 experienced physicians. For ethical issues, we have required approval from medical institutions.

Each patient underwent four views during scanning, which included parasternal left ventricle long axis (LVLA), pulmonary artery long axis (PALA), left ventricular short-axis (LVSA), and apical four-chamber heart (A4C), resulting in four videos per patient. The resolution of each video was either 800x600 or 1024x768, depending on the scanner used (Philips or HITACHI). A total of 516 and 476 videos were collected from Site G and Site R, respectively, from approximately 100 different patients. Each video consists of over 100 frames, covering at least one heartbeat cycle.

We have provided pixel-level annotations for each view, including masks for the left ventricle (LV) and right ventricle (RV) in the LVLA view, masks for the pulmonary artery (PA) in the PALA view, masks for the LV and RV in the LVSA view, and masks for the LV, RV, left atrium (LA), and right atrium (RA) in the A4C view. The videos in both Site R and Site G were divided into a ratio of 8:1:1 for training, validation, and testing, respectively. To lower annotation costs in the training set, only five frames per video are provided with pixel-level annotation masks. To better measure the model performance, we provide pixel-level annotations for every frame in each video in the validation and testing sets.

CAMUS [22] consists of 500 echocardiogram videos with pixel-level annotations for the left ventricle, myocardium, and left atrium. To save the annotation cost, only 2 frames (end diastole and end systole) are labelled in each video. We randomly split the dataset into $8:1:1$ for training, validation, and testing.

Echonet Dynamic [32] is the largest echocardiogram video dataset, including 10,030 videos with human expert annotations. Similarly, we split videos into $8:1:1$ for training, validation, and testing, respectively.

4.2. Implementation Details

Training. All methods are built on the "DeepLabv3" [3] backbone for fair comparison. We trained the model using the stochastic gradient descent (SGD) optimizer with a weight decay of 0.0001 and a momentum of 0.9. The model was trained for a total of 400 epochs with an initial learning rate of 0.02, and the learning rate was decreased by a factor of 0.1 every 100 epochs. The batch size was set to 4. For spatial data augmentation, each frame was resized to 384×384 and then randomly cropped to 256×256. The frames were also randomly flipped vertically and horizontally. As for temporal data augmentation, we randomly selected 40 frames from an echocardiogram video and sampled 10 frames as input equidistantly. We followed the same training and data augmentation approach for the CAMUS and Echonet dynamic datasets as we did for our dataset.

Validation and Testing. We chose the model with the highest performance on the validation set and reported its results on the testing set. During the inference stage, we only used center cropping as the preprocessing.

4.3. Comparison with the State-of-the-Art Methods

Results on CardiacUDA. We compare our method with existing state-of-the-art UDA methods [23, 24, 19, 21, 29, 43, 43, 37, 40] in the computer vision domain. Furthermore, considering the similar visual appearances of different domains, we also compare our method with several state-of-the-art semi-supervised segmentation methods [4, 45, 38], where images in the source domain are considered as labelled images, and images from target domain are treated as unlabelled images.

The performance is evaluated on two settings, as shown in Table 2. Our method demonstrated superior performance compared to the best-performing method [37], achieving a 3.9% and 6.2% improvement on averaged Dice under two settings, respectively. Notably, our method can surpass the best semi-supervised segmentation methods [38, 45] by 10% and 11.6% on averaged Dice under two settings, respectively. This comparison further highlights the significant domain gaps between site G and site R, demonstrating the effectiveness of our developed UDA method. Figure 5 shows the visualization of the segmentation results, where our method outperformed the other methods.

Results on our CardiacUDA, CAMUS, and Echonet. Table 3 shows the results of our UDA methods under six settings with three datasets. $a \rightarrow b$ indicates that a is the source domain and b is the target domain. We can see that our method can achieve excellent performance under six settings. Notably, as shown in Echo $\rightarrow$ CAMUS, our method can achieve 87.6% and 82.4% on Dice for EDV and ESV, respectively, which are very close to the upper bound of this setting. We also compare our method with state-of-the-art methods on different settings in Table 3, which

Table 2. Results on CardiacUDA dataset. "*Without DA*": evaluating the model trained on the *source domain* directly on the target domain. "*Upper Bound*": evaluating the model trained on the *target domain* with labels directly on the same domain. "Avg." refers to the averaged Dice score over four views, including LVLA, PALA, LVSA, and A4C. All results are reported in Dice score (%)

Method	site G (*source*) $\longrightarrow$ site R (*target*)					site R (*source*) $\longrightarrow$ site G (*target*)				
	LVLA	PALA	LVSA	A4C	Avg.	LVLA	PALA	LVSA	A4C	Avg.
Semi-Supervised Segmentation Methods										
CPS [4]	63.2±0.9	60.5±1.1	57.0±0.8	64.2±1.0	61.2±0.7	64.9±1.4	63.3±1.1	59.8±0.6	61.0±1.9	62.2±1.4
PC^{2}Seg [45]	64.1±0.8	58.2±0.6	70.2±0.9	63.9±1.2	64.1±1.1	63.7±1.0	64.4±1.0	65.2±1.3	67.0±0.9	65.1±0.8
U^{2}PL [38]	66.2±1.2	62.9±1.5	69.1±0.9	64.4±1.4	65.6±1.6	62.1±1.2	61.5±1.1	61.9±1.0	65.0±0.5	62.7±0.9
Unsupervised Domain Adaptation Methods										
Without DA	53.5±0.4	42.8±0.7	47.6±0.6	50.1±1.0	48.5±0.7	55.2±0.5	47.4±0.6	49.1±0.3	52.8±0.9	51.1±0.5
CCM [23]	13.5±2.6	10.3±3.8	13.0±3.2	20.7±2.9	22.3±3.5	17.8±4.0	14.2±3.1	25.7±2.9	16.6±3.2	18.6±3.0
Caco [19]	40.7±2.4	39.9±2.6	28.4±2.1	34.1±1.9	35.8±2.3	36.2±3.0	35.9±2.6	27.0±1.8	38.3±2.5	32.7±2.8
RIPU [40]	36.0±2.3	34.9±2.8	25.8±3.0	31.1±1.9	31.9±2.2	27.4±2.1	36.2±1.7	32.5±1.9	34.9±2.6	32.8±2.4
CPSL [24]	35.4±1.4	45.1±1.6	39.7±1.5	51.2±1.3	42.6±2.0	44.2±1.6	53.0±2.3	39.5±1.5	42.6±2.0	44.8±1.9
PLCA [21]	58.2±1.7	21.0±4.9	40.2±3.6	60.3±2.3	44.9±3.1	60.1±2.9	38.8±3.0	42.9±2.7	59.4±2.6	50.3±2.9
PixMatch [29]	60.8±1.8	52.7±1.6	56.0±1.8	66.5±1.5	59.0±1.7	62.9±2.0	49.0±3.2	63.2±2.1	69.9±1.8	61.3±1.8
FDA [43]	67.3±2.0	65.5±1.5	54.8±2.3	64.3±1.9	63.0±1.5	65.8±1.7	63.2±1.8	61.9±2.1	64.5±1.5	63.9±1.8
FDA-MBT [43]	64.4±0.9	65.1±0.8	61.7±1.1	70.1±1.3	65.3±1.2	66.3±0.9	64.9±1.4	67.2±0.6	71.3±0.9	67.4±0.8
FADA [37]	70.1±1.2	68.3±1.4	76.1±0.7	72.4±0.6	71.7±0.8	69.9±0.9	67.7±1.0	74.5±1.4	70.0±0.5	70.5±1.1
Ours	**73.9±1.2**	**75.5±1.3**	**76.8±0.4**	**76.3±0.7**	**75.6±0.9**	**73.3±1.0**	**74.9±1.2**	**80.2±0.3**	**78.2±0.5**	**76.7±0.5**
Upper Bound	79.1±0.4	82.4±0.6	82.1±1.0	81.4±1.2	81.3±0.5	80.5±1.4	79.2±0.8	83.3±1.6	83.9±0.2	81.7±0.8

Table 3. Results on CAMUS, Echonet dynamic and CardiacUDA datasets. As only LV segmentation labels are provided in these three datasets, we report the results on the dice score of LV segmentation. "EDV" and "ESV" refers to the Dice score of LV segmentation results at end-systole and end-diastole frames, respectively. All results are reported in Dice score (%)

Method	CAMUS→Echo		Echo→CAMUS		Ours→Echo		Ours→CAMUS		Echo→Ours	CAMUS→Ours
	EDV	ESV	EDV	ESV	EDV	ESV	EDV	ESV	Avg.	Avg.
Without DA.	69.2±1.4	66.2±2.2	64.3±0.9	59.6±1.7	34.1±1.4	33.8±2.2	31.0±0.9	32.4±1.7	22.9±1.6	19.2±1.4
U^{2}PL [38]	63.2±0.9	67.8±0.9	57.2±1.1	60.1±1.2	49.5±0.9	51.3±0.7	43.1±0.9	46.7±1.2	36.5±1.0	34.3±0.8
Caco [19]	55.9±1.5	56.0±1.3	47.3±1.3	49.0±1.2	38.6±3.4	40.8±2.6	46.1±1.9	45.5±2.2	29.6±1.8	26.8±2.3
RIPU [40]	64.3±1.2	67.7±1.0	70.2±0.6	68.2±0.9	46.9±0.7	47.4±0.7	56.0±1.2	51.5±1.4	36.2±1.7	31.7±2.0
PLCA [21]	71.1±0.4	69.3±0.5	72.9±0.8	68.3±0.7	51.9±0.9	49.7±0.9	52.4±1.4	52.1±1.2	35.3±1.1	36.1±0.8
FDA [43]	78.8±1.1	75.4±1.0	76.2±0.4	74.1±0.5	55.6±0.4	54.0±0.6	56.8±0.5	56.1±0.3	38.4±1.5	37.9±1.2
FADA [37]	77.5±0.8	76.5±0.5	78.6±0.8	76.6±0.6	54.1±0.6	52.0±1.0	57.4±1.0	55.2±0.8	41.7±1.1	39.0±0.9
Ours	**83.4±0.7**	**81.8±0.9**	**87.6±0.4**	**82.4±1.0**	**61.2±0.5**	**61.8±0.7**	**66.3±0.3**	**64.9±0.4**	**46.2±1.3**	**44.0±1.6**
Upper Bound	93.4±0.6	90.5±1.3	89.3±1.1	87.9±0.8	93.4±0.6	90.5±1.3	89.3±1.1	87.9±0.8	81.3±0.9	81.3±0.6

shows our method outperforms all other methods with significant improvement.

4.4. Ablation Study

Effectiveness of SCGM and TCC. Table 4 shows the effectiveness of our proposed SCGM and TCC. "Base" indicates the basic segmentation network. The results show that adopting SCGM can largely improve the base model from 48.5% to 74.3% under setting G $\rightarrow$ R. However, only applying TCC shows limited improvements over the base model. This is mainly because the TCC is designed to jointly train unlabelled data and construct better graphical representation in a temporal manner, which does not include any operation that focuses on narrowing the domain discrepancy, leading to limited adaptation results. The combination of SCGM and TCC can achieve the best performance.

Ablation study of SCGM. Since there are two loss functions, *i.e.*, $\mathcal{L}_{cls}$ (Eq. 2) and $\mathcal{L}_{mat}$ (Eq. 3) in SCGM, we ablate their effects in Table 5. The results illustrate that using $\mathcal{L}_{cls}$ and $\mathcal{L}_{mat}$ alone can only achieve limited improvements. This is because only using $\mathcal{L}_{cls}$ can not align the representations from different domains well while only using $\mathcal{L}_{mat}$ may perform the erroneous alignment, *e.g.*, align the features of LV to those of RV. By combining two losses, we can conduct the correct class-wise alignment and achieve significant improvement.

Ablation study of TCC. We explore the effects of two loss functions ($\mathcal{L}_{tc}$ (Eq. 5) and $\mathcal{L}_{adv}$) in TCC in Table 6. In this ablation study, we use SCGM as the baseline model, which has been ablated. We can see that both $\mathcal{L}_{tc}$ and $\mathcal{L}_{adv}$ can benefit the model, and using two losses can achieve the best performance. For the visualisation of the effectiveness of

Table 4. Effectiveness of SCGM and TCC. Results report in averaged Dice score.

	SCGM	TCC	Dice Scores (%)	
			G→R	R→G
Base	✗	✗	48.5±0.7	51.1±0.5
Base + SCGM	✓	✗	74.3±1.0	71.3±0.7
Base + TCC	✗	✓	55.3±0.8	53.0±1.2
Ours	✓	✓	**75.6±0.9**	**76.7±0.5**

Table 5. Effect of $\mathcal{L}_{cls}$ (Eq. 2) and $\mathcal{L}_{mat}$ (Eq. 3) in SCGM.

$\mathcal{L}_{cls}$	$\mathcal{L}_{mat}$	Averaged Dice Score (%)	
		G→R	R→G
✗	✗	48.5±0.7	51.1±0.5
✓	✗	51.9±1.2	53.7±1.3
✗	✓	53.4±1.6	54.0±1.5
✓	✓	74.3±1.0	71.3±0.7

Table 6. Effect of $\mathcal{L}_{tc}$ (Eq. 5) and $\mathcal{L}_{adv}$ in TCC.

$\mathcal{L}_{tc}$	$\mathcal{L}_{adv}$	Averaged Dice Score (%)	
		G→R	R→G
✗	✗	74.3±0.7	71.3±0.5
✓	✗	74.4±1.1	75.6±0.9
✗	✓	74.1±0.9	73.5±1.0
✓	✓	**75.6±0.9**	**76.7±0.5**

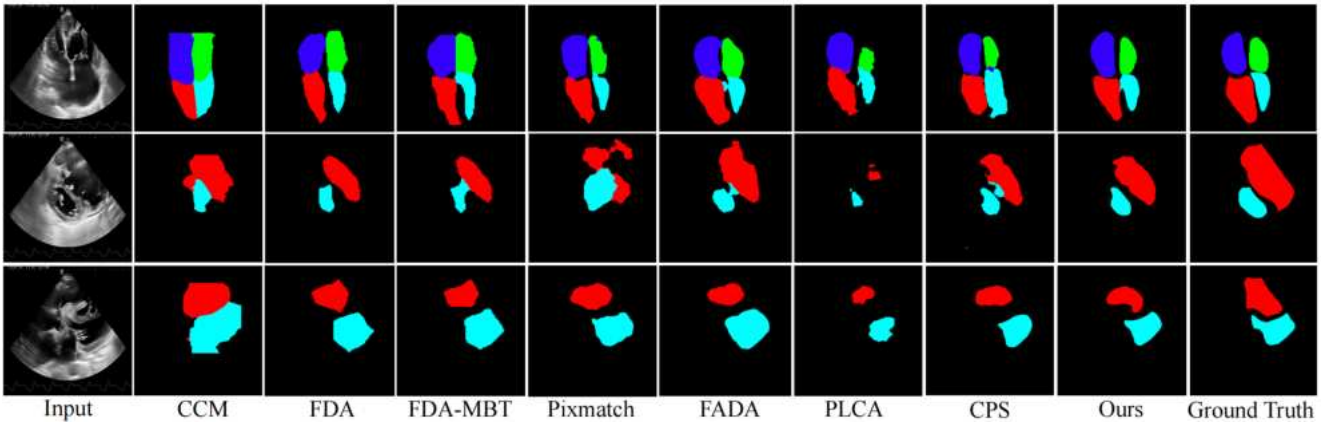


Figure 5. We visualize three video frames to show the segmentation results. Red, green, blue, and cyan indicate refer to the segmentation regions for the right Atrium (RA), left ventricle (LV), right ventricle (RV), and left atrium (LV), respectively.

Table 7. Analysis of different attentions. "None" denotes that no attention module is applied in our framework, while the "Inter", "Intra", and "Inter-Intra" refers to cross-domain, internal domain, and dual (cross+internal) attention, respectively.

Method	Site G ⟶ Site R				
	LVLA	PALA	LVSA	A4C	Avg.
None	68.1±2.2	69.8±0.9	71.4±1.3	73.3±0.4	70.7±1.2
Inter	70.5±1.5	72.6±1.3	72.0±0.8	73.5±0.4	72.2±1.0
Intra	72.1±0.5	71.5±0.8	75.2±1.7	74.6±0.7	73.4±0.9
Inter-Intra	**73.9±1.2**	**75.5±1.3**	**76.8±0.4**	**76.3±0.7**	**75.6±0.9**

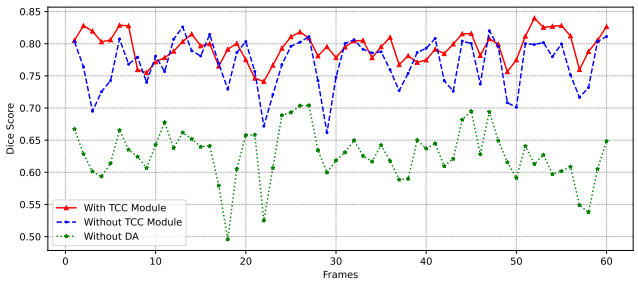


Figure 6. Dice score of the segmentation result for each frame in a video example. The x-axis represents the frame indexes in the video, and the y-axis is the corresponding Dice score.

the TCC module, figure 5 illustrates the segmentation result generated by our framework with the TCC module is able to present more consistent performance (marked by the red line) in a video. The results without the TCC module or disabling the domain adaptation perform worse on segmentation consistency.

Effect of the types of attention. As shown in Table 7, we compare different node attention methods. The results show that inter-intra attention achieves the best performance in our datasets, which indicates the relations between different domains are important to improve the performance.

TCC can learn temporal information. Figure 5 shows the Dice score for each frame in a video example. Compared to results without using TCC, our method produces better results with enhanced temporal consistency, showing the effectiveness of the TCC module in learning temporal information.

5. Conclusion and Limitation

In this paper, we introduced a newly collected CardiacUDA dataset and a novel GraphEcho method for cardiac structure segmentation from echocardiogram videos. Our GraphEcho involved two innovative modules, the Spatial-wise Cross-domain Graph Matching (SCGM) and the Temporal Cycle Consistency (TCC) module. These two modules are motivated by the fact that the structure of different cardiac structures is similar across different patients and domains and the cyclical consistency property of echocardiogram videos. Our approach enables improved UDA segmentation results by effectively aligning global and local features from both source and target domains, thereby preserving both inter-class differences and intra-class similarities. Experimental results showed that our GraphEcho outperforms existing state-of-the-art UDA segmentation methods. In our future work, we will explore how to represent objects with complex contours in other medical domains with more efficient representation and conducted the graph-based method on more complicated scenarios such as CT and MRI in future work.

Acknowledgement

This work was supported by a research grant from the Beijing Institute of Collaborative Innovation (BICI) under collaboration with the Hong Kong University of Science and Technology under Grant HCIC-004.

References

[1] Cheng Chen, Qi Dou, Hao Chen, Jing Qin, and Pheng Ann Heng. Unsupervised bidirectional cross-modality adaptation via deeply synergistic image and feature alignment for medical image segmentation. *IEEE TMI*, 39(7):2494–2505, 2020.

[2] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*, pages 1542–1553. PMLR, 2020.

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[4] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

[5] Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. Self-training avoids using spurious features under domain shift. *Advances in Neural Information Processing Systems*, 33:21061–21071, 2020.

[6] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[7] Weihang Dai, Xiaomeng Li, Xinpeng Ding, and Kwang-Ting Cheng. Cyclical self-supervision for semi-supervised ejection fraction prediction from echocardiogram videos. *arXiv preprint arXiv:2210.11291*, 2022.

[8] Xinpeng Ding and Xiaomeng Li. Exploring segment-level semantics for online phase recognition from surgical videos. *IEEE Transactions on Medical Imaging*, 41(11):3309–3319, 2022.

[9] Xinpeng Ding, Nannan Wang, Shiwei Zhang, De Cheng, Xiaomeng Li, Ziyuan Huang, Mingqian Tang, and Xinbo Gao. Support-set based cross-supervision for video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11573–11582, 2021.

[10] Zhengming Ding, Sheng Li, Ming Shao, and Yun Fu. Graph adaptive knowledge transfer for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 37–52, 2018.

[11] Pamela S Douglas, Mario J Garcia, David E Haines, Wyman W Lai, Warren J Manning, Ayan R Patel, Michael H Picard, Donna M Polk, Michael Ragosta, R Parker Ward, et al. 2011 appropriate use criteria for echocardiography: a report of the american college of cardiology foundation appropriate use criteria task force. *Journal of the American College of Cardiology*, 57(9):1126–1166, 2011.

[12] Andrew J Fletcher, Winok Lapidaire, and Paul Leeson. Machine learning augmented echocardiography for diastolic function assessment. *Frontiers in Cardiovascular Medicine*, page 879, 2021.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[14] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2477–2486, 2019.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[16] Dayan Guan, Jiaxing Huang, Aoran Xiao, and Shijian Lu. Domain adaptive video segmentation via temporal consistency regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8053–8064, 2021.

[17] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. *arXiv preprint arXiv:2206.00272*, 2022.

[18] Xiaoting Han, Lei Qi, Qian Yu, Ziqi Zhou, Yefeng Zheng, Yinghuan Shi, and Yang Gao. Deep symmetric adaptation network for cross-modality medical image segmentation. *IEEE TMI*, 41(1):121–132, 2021.

[19] Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1203–1214, 2022.

[20] J Weston Hughes, Neal Yuan, Bryan He, Jiahong Ouyang, Joseph Ebinger, Patrick Botting, Jasper Lee, John Theurer, James E Tooley, Koen Nieman, et al. Deep learning evaluation of biomarkers from echocardiogram videos. *EBioMedicine*, 73:103613, 2021.

[21] Guoliang Kang, Yunchao Wei, Yi Yang, Yueting Zhuang, and Alexander Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. *Advances in Neural Information Processing Systems*, 33:3569–3580, 2020.

[22] Sarah Leclerc, Erik Smistad, Joao Pedrosa, Andreas Østvik, Frederic Cervenansky, Florian Espinosa, Torvald Espeland, Erik Andreas Rye Berg, Pierre-Marc Jodoin, Thomas Grenier, et al. Deep learning for segmentation using an open large-scale dataset in 2d echocardiography. *IEEE transactions on medical imaging*, 38(9):2198–2210, 2019.

[23] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *European conference on computer vision*, pages 440–456. Springer, 2020.

[24] Ruihuang Li, Shuai Li, Chenhang He, Yabin Zhang, Xu Jia, and Lei Zhang. Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11593–11603, 2022.

[25] Wuyang Li, Xinyu Liu, and Yixuan Yuan. Sigma: Semantic-complete graph matching for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5291–5300, 2022.

[26] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021.

[27] Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8266–8276, 2019.

[28] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *European conference on computer vision*, pages 415–430. Springer, 2020.

[29] Luke Melas-Kyriazi and Arjun K Manrai. Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12435–12445, 2021.

[30] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[32] David Ouyang, Bryan He, Amirata Ghorbani, Neal Yuan, Joseph Ebinger, Curtis P Langlotz, Paul A Heidenreich, Robert A Harrington, David H Liang, Euan A Ashley, et al. Video-based ai for beat-to-beat assessment of cardiac function. *Nature*, 580(7802):252–256, 2020.

[33] Alexander Papolos, Jagat Narula, Chirag Bavishi, Farooq A Chaudhry, and Partho P Sengupta. Us hospital use of echocardiography: insights from the nationwide inpatient sample. *Journal of the American College of Cardiology*, 67(5):502–511, 2016.

[34] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[36] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.

[37] Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *European conference on computer vision*, pages 642–659. Springer, 2020.

[38] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4248–4257, 2022.

[39] Fuping Wu and Xiahai Zhuang. Unsupervised domain adaptation with variational approximation for cardiac segmentation. *IEEE Transactions on Medical Imaging*, 40(12):3555–3567, 2021.

[40] Binhui Xie, Longhui Yuan, Shuang Li, Chi Harold Liu, and Xinjing Cheng. Towards fewer annotations: Active learning via region impurity and prediction uncertainty for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8068–8078, 2022.

[41] Qingsong Xie, Yuexiang Li, Nanjun He, Munan Ning, Kai Ma, Guoxing Wang, Yong Lian, and Yefeng Zheng. Unsupervised domain adaptation for medical image segmentation by disentanglement learning and self-training. *IEEE Transactions on Medical Imaging*, 2022.

[42] Jiewen Yang, Xingbo Dong, Liujun Liu, Chao Zhang, Jiajun Shen, and Dahai Yu. Recurring the transformer for video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14063–14073, 2022.

[43] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020.

[44] Huifeng Yao, Xiaowei Hu, and Xiaomeng Li. Enhancing pseudo label quality for semi-supervised domain-generalized medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3099–3107, 2022.

[45] Yuanyi Zhong, Bodi Yuan, Hong Wu, Zhiqiang Yuan, Jian Peng, and Yu-Xiong Wang. Pixel contrastive-consistent semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7273–7282, 2021.

[46] Ronghang Zhu, Xiaodong Jiang, Jiasen Lu, and Sheng Li. Cross-domain graph convolutions for adversarial unsupervised domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[47] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019.