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

1. Algorithm Pipeline

Based on the GraphEcho that we have presented in Section 3, the SGCM and TCC module can be formulated as Algorithm 1 and Algorithm 2. Note that all the superscript s and t of all variables represent both source and target domains. For instance, $\mathbf{x}^{s,t}$ indicate the input from both source $\mathbf{x}^s$ and target $\mathbf{x}^t$ domains.

2. Visualization for sequences of echocardiogram videos

In this supplementary, we provide more visualization results (see figure 1 and 2) for the sequences of echocardiogram videos.

3. Code available

Our codes are available in our supplementary material. For the detail of the training and evaluation, please see the **ReadMe.md** in our **ID2623_code.zip** attachment.

4. The example of our dataset CardiacUDA

We provide some examples of our CardiacUDA dataset, including a training example with annotation and a testing example with annotation. Please see our attachment **id2623_dataset_example.zip**.

References

[1] 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.

[2] 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.

---

[*] Corresponding author

[†] Interning at The Hong Kong University of Science and Technology

Algorithm 1: Spatial-wise Cross-domain Graph Matching Module (SGCM)

```
Output:
    $\mathcal{L}_{SCGM}$ : The overall loss of the SGCM;
    $\mathcal{L}_{cls}$ : The classification loss of the SGCM;
    $\mathcal{L}_{mat}$ : The graph matching loss of the SGCM;
    $\mathcal{L}_{seg}$ : Supervised segmentation loss;
    $\mathcal{L}_{bce}$ : Binary cross-entropy loss;
    $\mathcal{L}_{dice}$ : Dice loss [2].

Input:
    $\mathbf{x}^{s,t}$ : Input video frames from source and target domains;
    $\mathbf{y}^{s}$ : The ground truth annotation of the source domain;
    $m$ : The number of the sampling feature;
    $C$ : The total classes number of segmentation region;
    $\alpha, \beta$ : The classification loss weight for source and target domains;
    $\text{Encoder}(\cdot)$ : The Parameter shared feature extractor;
    $\text{Decoder}(\cdot)$ : The Parameter shared decoder for generating the segmentation result;
    $\mathbb{I}(\cdot)$ : Indicator function.

$\mathbf{f}^{s,t} \leftarrow \text{Encoder}(\mathbf{x}^{s,t})$
$\hat{\mathbf{y}}^{s,t} \leftarrow \text{Decoder}(\mathbf{f}^{s,t})$
$\mathcal{L}_{seg} \leftarrow \mathcal{L}_{bce}(\hat{\mathbf{y}}^{s}, \mathbf{y}^{s}) + \mathcal{L}_{dice}(\hat{\mathbf{y}}^{s}, \mathbf{y}^{s})$

(1).Node Sampling:
for $i = 1$ to in $C$ do
    $\{\mathbf{f}_i^{s,t}\} \leftarrow$ Get different chamber region $\{\mathbf{f}_i^{s,t}\}$ according to $\mathbf{y}^{s}$ and $\hat{\mathbf{y}}^{t}$.
    $\mathbf{v}^{s,t} \leftarrow$ Uniformly sample $m$ feature vectors from $\{\mathbf{f}_i^{s,t}\}$ as the node embedding $\mathbf{v}^{s,t}$.
end for

(2).Node Classification:
$\bar{\mathbf{v}}^{s,t} \leftarrow Attention(concat(\mathbf{v}^{s}, \mathbf{v}^{t}))$
$\mathcal{L}_{cls} \leftarrow -\alpha \mathbf{y} log(h(\bar{\mathbf{v}}^{s})) - \beta \hat{\mathbf{y}} log(h(\bar{\mathbf{v}}^{t}))$

(3).Graph Matching:
$g^{s,t} \leftarrow$ Add the learned matrix as the edge connections $\mathbf{e}^{s,t}$ to $\mathbf{v}^{s,t}$ and constructed semantic graph $g^{s,t}$.
$\mathbf{A} \leftarrow$ Obtain adjacency matrix $\mathbf{A}$ from $g^{s}$ and $g^{t}$.
$\vec{\mathbf{A}} \leftarrow Sinkhorn(\mathbf{A})$: Obtain transport cost matrix of graphs among chambers.
Minimize the transport distance of $p$-th row and $q$-th column element on $\vec{\mathbf{A}}$.
    $\mathcal{L}_{mat} \leftarrow \sum_{p,q} \mathbb{I}(\mathbf{y}_p^{s} = \hat{\mathbf{y}}_q^{t}) \cdot min(\vec{\mathbf{A}}(p,q)) + \mathbb{I}(\mathbf{y}_p^{s} \neq \hat{\mathbf{y}}_q^{t}) \cdot max(\vec{\mathbf{A}}(p,q))$.

Overall Loss:
$\mathcal{L}_{SCGM} = \mathcal{L}_{cls} + \mathcal{L}_{mat}$.
```

Algorithm 2: Temporal-wise Cycle Consistency Module (TCC)

```
Output:
    $\mathcal{L}_{tc}^{s,t}$ : The temporal consistency loss for the source and target domains;
    $\mathcal{L}_{TCC}$ : The overall loss of temporal consistency.
Input:
    $\mathbf{X}^{s,t}$ : Input video from source and target domains.
    $\mathcal{L}_{adv}$ : The adversarial methods [1] to eliminate the domain gap with global feature alignment;
    $N$ : The number of frames in $\mathbf{X}^{s,t}$;
    $\mathbf{h}_t$ : The hidden state, and the $h_0$ is learned parameters with all zero in initial state;
    $\mathbf{w}_{gcn}, \mathbf{b}_{gcn}$ : The graph convolution weight and bias;
    $\sigma$ : The activation function;
    $FFN$ : Feed forward network;
    Encoder($\cdot$) : The Parameter shared feature extractor;
    RGCC($\cdot$) : Recursive graph convolutional cell.

    (1).Temporal Graph Node Construction:
    $\{\mathbf{f}_i^{s,t}\}_{i=1}^{N} \leftarrow$ Encoder($\mathbf{X}^{s,t}$), Where $\mathbf{f}_i$ is the feature of the $i$-th frame.
    $\{\ddot{\mathbf{v}}_i^{s,t}\}_{i=1}^{N} \leftarrow$ Apply average pooling and node sampling (see algorithm 1.(1)) to feature map $\{\mathbf{f}_i^{s,t}\}_{i=1}^{N}$.

    (2).Recursive Graph Convolutional Cell (RGCC):
    for $t = 0$ to $N$ do
        $\{\ddot{\mathbf{v}}_t^{s,t}, \ddot{\mathbf{e}}_t^{s,t}\} \leftarrow$ Find $K$ nearest neighbors on the hidden state $\mathbf{h}_t$ for each node at $\ddot{\mathbf{v}}_t^{s,t}$.
        $\mathbf{h}_{t+1}^{s,t} \leftarrow \sigma \mathbf{w}_{gcn}(\ddot{\mathbf{v}}_i^{s,t}, \ddot{\mathbf{e}}_i^{s,t}) + \mathbf{b}_{gcn}$
    end for
    $\mathbf{o}^{s,t} \leftarrow FFN(\mathbf{h}_N^{s,t})$

    (3).Temporal consistency loss (source domain as an example):
    $\mathcal{L}_{tc}^{s} \leftarrow -\sum_{\{\mathbf{o}_k^s, \mathbf{o}_+^s\} \in \mathcal{P}^s} log \frac{\exp(\mathbf{o}_k^s \cdot \mathbf{o}_+^s)}{\sum_{\mathbf{o}_-^s \in \mathcal{B}} \exp(\mathbf{o}_k^s \cdot \mathbf{o}_-^s)}$
    where $\mathcal{P}^s$ is the set of positive pairs, $\mathbf{o}_k^s$ and $\mathbf{o}_+^s$ are representations that randomly sampled from a video $\mathbf{X}^s$ as positive pairs. For the negative samples, we maintain a memory bank $\mathcal{B}$ consisting of representations of clips sampled from different videos.

    Overall Loss:
    $\mathcal{L}_{TCC} = \mathcal{L}_{tc}^{s,t} + \mathcal{L}_{adv}$
```

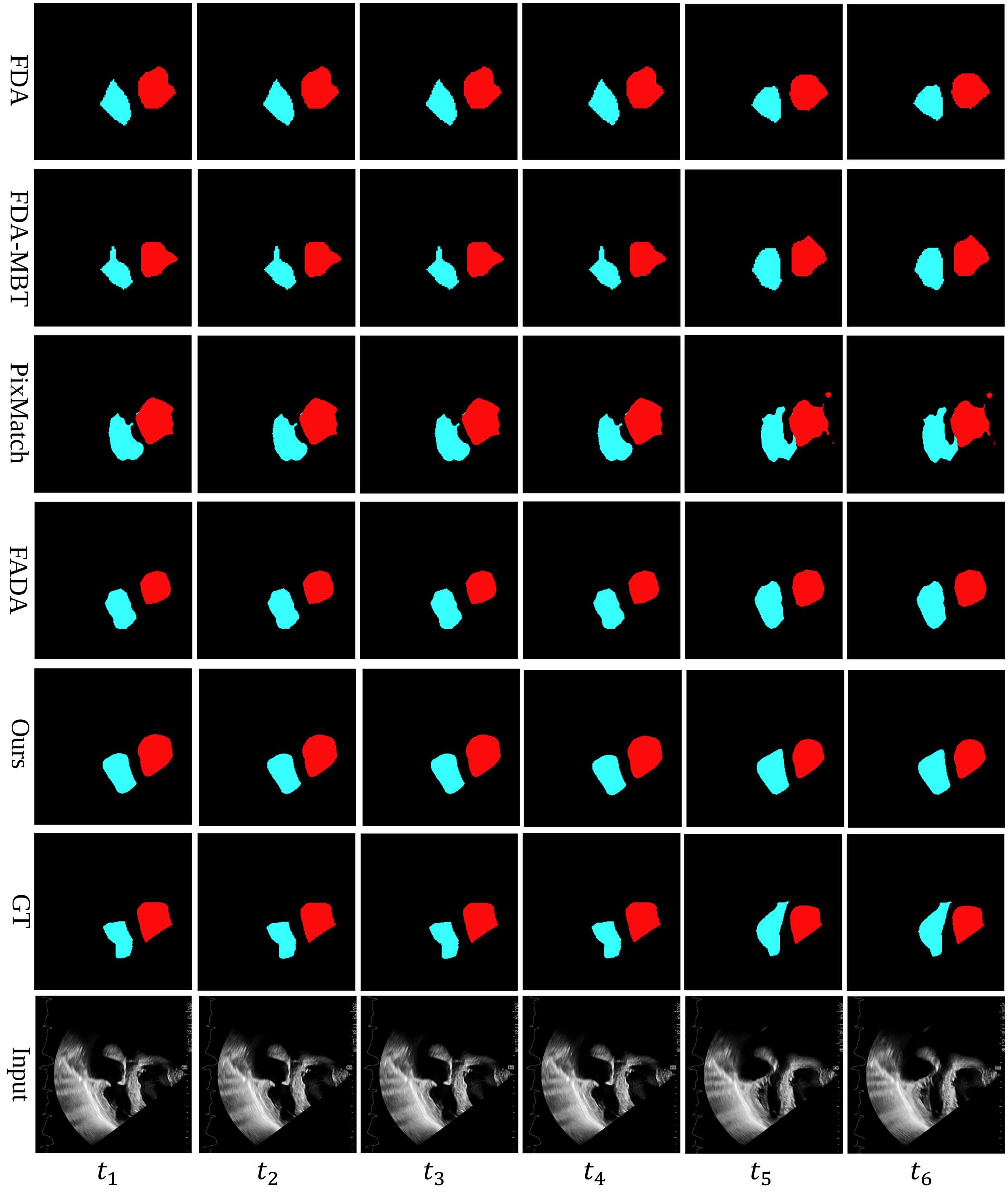

Figure 1. We visualize a video sequence of parasternal left ventricle long axis view to show the segmentation results (GT denotes the ground truth segmentation result). Red and cyan indicate refer to the segmentation regions for the right Atrium (RA) and left atrium (LV), respectively. The $\{t1, ..., t6\}$ denotes the frame order of a video sequence.

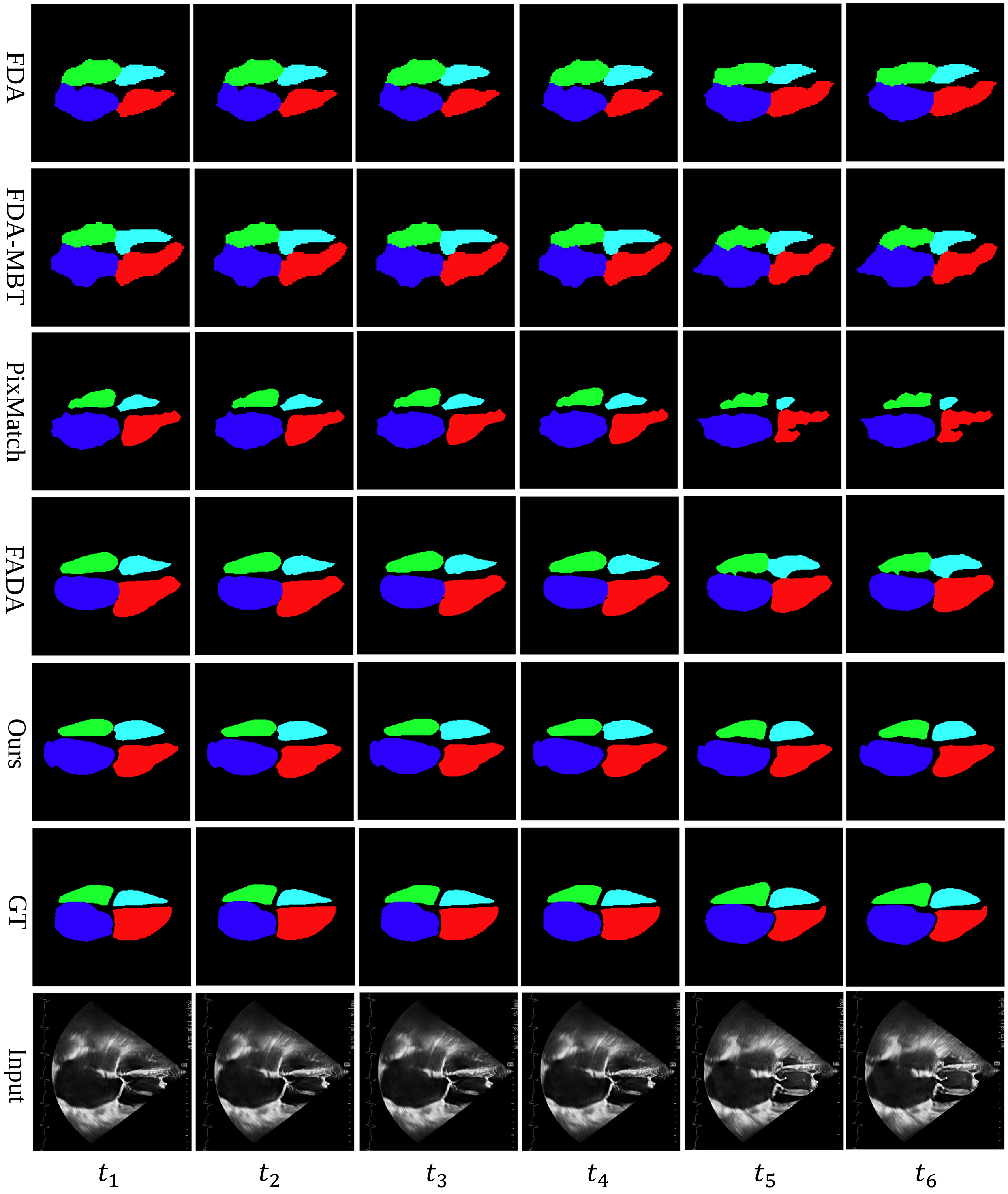

Figure 2. We visualize a video sequence of an apical four-chamber heart view to show the segmentation results. Red, green, blue, and cyan indicate refer to the segmentation regions for the right Atrium (RA), left ventricle (LV), right ventricle (RV), and left atrium (LV), respectively. The $\{t1, ..., t6\}$ denotes the frame order of a video sequence.