# Probabilistic Modeling for Image Registration Using Radial Basis Functions: Application to Cardiac Motion Estimation

Ziyu Gan, Wei Sun, Kaimin Liao, and Xuan Yang

***Abstract*— Cardiovascular diseases (CVDs) are the leading cause of death, affecting the cardiac dynamics over the cardiac cycle. Estimation of cardiac motion plays an essential role in many medical clinical tasks. This article proposes a probabilistic framework for image registration using compact support radial basis functions (CSRBFs) to estimate cardiac motion. A variational inference-based generative model with convolutional neural networks (CNNs) is proposed to learn the probabilistic coefficients of CSRBFs used in image deformation. We designed two networks to estimate the deformation coefficients of CSRBFs: the first one solves the spatial transformation using given control points, and the second one models the transformation using drifting control points. The given-point-based network estimates the probabilistic coefficients of control points. In contrast, the drifting-point-based model predicts the probabilistic coefficients and spatial distribution of control points simultaneously. To regularize these coefficients, we derive the bending energy (BE) in the variational bound by defining the covariance of coefficients. The proposed framework has been evaluated on the cardiac motion estimation and the calculation of the myocardial strain. In the experiments, 1409 slice pairs of end-diastolic (ED) and end-systolic (ES) phase in 4-D cardiac magnetic resonance (MR) images selected from three public datasets are employed to evaluate our networks. The experimental results show that our framework outperforms the state-of-the-art registration methods concerning the deformation smoothness and registration accuracy.**

***Index Terms*— Cardiac motion estimation, compact support radial basis function (CSRBF), deep learning (DL), deformable registration, probabilistic learning.**

## I. Introduction

CARDIOVASCULAR diseases (CVDs), such as ischemic heart disease, lead to abnormal motion of the left ventricular myocardium over the cardiac cycle. Cardiac motion estimation, especially left ventricle (LV) motion estimation, is essential for understanding cardiac mechanics and diagnosing and treating CVDs.

Existing works aiming at estimating cardiac motion fall into two categories: feature-tracking-based methods [1], [2] and deformation-based methods [3], [4]. Two critical disadvantages of feature-tracking-based methods are that the accuracy of feature tracking-based methods highly depends on the accuracy of feature extraction, and inadequate local features may lead to the failure of feature tracking [5]. Although deformation-based methods do not rely on feature extraction, their optimization procedures are usually coupled with high computational complexity. The optimization is ill-posed, which may cause many local minima during the optimization.

With the upsurge of deep learning (DL) techniques, DL-based deformable registration methods are proposed. These methods employ well-trained neural networks to estimate the correspondence and deformation between the image pair. Compared with conventional registration methods, DL-based methods obtain registration results rapidly with the trained network.

According to the training strategy of DL, the DL-based registration methods can be classified as supervised learning and unsupervised learning. Supervised learning methods train the network by minimizing a loss function, which measures the difference between the predicted parameters of the deformation model and the given ground truth. However, due to few ground truths of medical datasets provided by experts, the supervised learning methods are limited by inadequate training samples. On the contrary, unsupervised learning methods train the network without ground truth [6]–[8]. Compared with supervised learning methods, unsupervised learning methods have more potential applications because no annotated data are required, which has gradually become the commonly used DL-based registration method.

However, most existing DL-based algorithms estimated dense displacement vector fields (DVFs) in a nonparametric registration way. Thus, the number of predicted parameters is the number of elements in a dense DVF. Therefore, the number of predicted parameters to estimate DVFs is massive. Furthermore, it is challenging to control the spatial relationship between these parameters, which easily results in nonsmooth or nontopology-preserving DVFs.

In this article, we employ compact support radial basis functions (CSRBFs) to perform a parametric deformation and estimate the coefficients of CSRBFs using networks to tackle the issue in DVFs. CSRBF-based transformation functions interpolate the coefficients of control points to the internal

Manuscript received September 28, 2020; revised April 7, 2021 and August 24, 2021; accepted December 31, 2021. This work was supported by the National Natural Science Foundation of China under Grant 61871269. *(Corresponding author: Xuan Yang.)*

Ziyu Gan, Kaimin Liao, and Xuan Yang are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: ganziyu@email.szu.edu.cn; liaokaimin2019@email.szu.edu.cn; yangxuan@szu.edu.cn).

Wei Sun is with the Radboud University Medical Center, 6500 Nijmegen, The Netherlands (e-mail: sunweidemail@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3141119.

Digital Object Identifier 10.1109/TNNLS.2022.3141119

points via the CSRBFs. The advantages of CSRBF-based deformation include the following: 1) only a limited number of parameters are needed to be predicted because the number of unknown parameters of CSRBF-based transformation only depends on the relatively sparse control points; 2) the deformation can be controlled by control points, which implies the deformation field can be adjusted finely by distributing control points wisely; 3) the BE can be computed in a closed form [9], which can be used to regularize the DVF effectively; and 4) CSRBFs can deform images locally, which is preferred in the deformation of cardiac cine magnetic resonance (MR) images.

Because of the benefits of CSRBFs, this article proposes a DL-based method using CSRBFs for deformable registration to estimate cardiac motion in an unsupervised fashion. Several DL-based methods also use radial basis functions (RBFs) to perform registration. Most of them use B-splines interpolation [10]–[12] and thin plate splines (TPSs) [13]–[16]. Besides, most of these works are either supervised methods or patch-wised registration. To the best of our knowledge, this article is the first work to perform DL-based unsupervised deformable registration combining variational inference and CSRBF-based transformations. We propose a probabilistic framework to estimate the coefficients of CSRBFs by conducting variational inference and use it to estimate cardiac motion. Variational inference is a method for approximating the intractable posterior distribution of latent variables in a generative process. Inspired by variational inference methods [17], [18] and their registration applications [7], [19], we construct a probabilistic generative model for CSRBF-based transformations using an unsupervised learning network.

First, we construct a variational inference framework to estimate the coefficients of CSRBFs. Next, we propose two networks to solve the framework for the given control points scenario and drifting control points scenario, respectively. In general, the spatial transformation using RBFs is usually based on given control points [20], [21]. These given control points are fixed during the registration process. Our first network is proposed for this scenario. To improve registration accuracy further, the regions with large deformation must be covered by control points well. Therefore, our second network is proposed to shift initial control points to proper locations and estimate the corresponding coefficients simultaneously. Moreover, to regularize the coefficients, we derive the BE form in the variational bound using the covariance of latent variables. Finally, our framework is evaluated in cardiac motion estimation using public datasets.

The contributions of our work are summarized in the following.

1) We propose a probabilistic learning model for image registration using CSRBF-based transformations by conducting variational inference. To the best of our knowledge, this is the first work that models CSRBF-based image registration using variational inference. The coefficients of CSRBFs are the latent variables in our model, and their distribution is estimated using a variational autoencoder (VAE). The prior of these latent variables is designed for representing the locations of control points, which enables a closed-form derivation of the BE of the DVF in the evidence lower bound (ELBO) of the variational inference, forcing the DVFs to be smooth. Because of introducing CSRBF-based transformations, our model outperforms the state-of-the-art works regarding deformation smoothness and registration accuracy.
2) Two novel networks, NetGI and NetDC, are proposed to solve the variational inference model. Especially, NetDC is designed to estimate the probabilistic coefficients of CSRBFs and the locations of control points simultaneously. It can be used to shift control points adaptively in CSRBF-based deformations to implement the local deformation via distributing control points unevenly. To the best of our knowledge, this is the first work to position control points adaptively using neural networks. Moreover, this new technique shows its advantage of improving the registration accuracy in the experiments.
3) Registration uncertainties can be predicted using the variance estimated by our networks. The uncertainty measures can be used in various applications, such as qualitative analyses and pathological areas' detection.
4) Our framework is evaluated in a cardiac motion estimation task and outperforms the state-of-the-art methods concerning registration accuracy and smooth DVFs.

We summarize the related works in Section II and provide an overview of our model in Section III-A. Section III-B introduces the spatial transformation function based on CSRBFs. The variational inference model based on CSRBFs is proposed in Section III-C. Two networks to solve our variational inference model are designed in Sections III-D1 and III-D2. In Section IV, the proposed framework is evaluated in cardiac motion estimation. Finally, conclusions are provided in Section V.

## II. Related Works

### A. Conventional Cardiac Motion Estimation

Feature-tracking-based methods commonly contain two steps: detecting features and then tracking feature points, such as the sampled points on the contours [2] and corner points from images [22]. Feature-tracking-based methods aim to search correspondences between point sets or graphs, such as point set matching [1], [23] and graph matching [2].

Deformation-based methods establish a dense nonlinear DVF between the moving image and the fixed image by optimizing an objective function that measures the image-intensity-based similarity between the image pair and imposing smoothness constraints [24] on DVF. For cardiac motion estimation, deformable registration employs a deformation model with parameters to warp the moving image to the fixed image and estimate the displacements of the points of the ventricular wall. The commonly used deformation models in LV motion estimation are free-form deformation model [25], polyaffine model [3], [26], elastic body model [4], and so on.

### B. DL-Based Deformable Registration

DL-based deformable registration methods employ well-trained neural networks to estimate the correspondence and deformation between the image pair. Compared with conventional registration methods, DL-based methods obtain registration results rapidly with an already-trained network. According to the training strategy used in DL, the DL-based methods fall into supervised learning and unsupervised learning.

Supervised learning methods train the network via minimizing an objective function, which calculates the difference between the predicted parameters of the deformation model and the ground truth. Rohé *et al.* [27] established ground-truth deformations between image pairs using segmented shapes and trained a fully convolutional neural network (CNN) for 3-D registration. Sokooti *et al.* [28] trained a CNN using a large set of artificially generated DVFs and did not explicitly define a dissimilarity metric. Eppenhof and Pluim [29] trained a 3-D CNN using synthetic random transformations, and the network was applied to a small set of representative images for the desired application to register inhale–exhale lung CT image pairs.

However, due to the limited ground truth of medical datasets provided by experts, the supervised learning methods are limited by inadequate training samples. A compromised solution is to use transformations computed by state-of-the-art algorithms or synthetic transformations [14], [16], [30]. The drawbacks of supervised learning methods are that registration quality depends on the ground truth quality, and the synthetic transformations are infeasible to simulate the actual deformation, especially for LV motion deformation.

On the contrary, unsupervised learning methods train the network without ground truth [6]–[8]. Due to the spatial transformer network (STN) proposed by Jaderberg *et al.* [31], the moving image can be deformed according to the estimated deformation field. Then, a loss function can be constructed as the similarity between the deformed moving image and the fixed image. Furthermore, additional constraints, such as the smoothness of deformation fields, can be introduced to the loss function. Because the STN is differentiable, the backpropagation algorithm can be performed to train the unsupervised learning network. De Vos *et al.* [11] proposed the DIRnet that combined a CNN regressor and an STN to estimate the local deformation parameters of cubic B-splines [32] by analyzing corresponding image patches from the moving and fixed images. Li and Fan [33] employed the FCN to directly estimate spatial transformations between pairs of images by maximizing an imagewise similarity metric between fixed and deformed moving images in a multi-resolution image registration framework. Sloan *et al.* [34] registered images of neurons using the mean squared error (MSE) between the warped and fixed volumes as the loss function to train a CNN. Balakrishnan *et al.* [35] and [36] proposed the VoxelMorph, a U-net [37] style network, to perform imagewised registration on brain MR images. Qin *et al.* [38] built an unsupervised Siamese style recurrent STN and performed the weakly supervised segmentation by taking advantage of unsupervised features learned in the motion estimation network from a large amount of unannotated cardiac data.

In the unsupervised registration method, generative models have shown unique advantages [6], [7], [39]–[41]. GANs are popular generative models. Unlike unsupervised works based on manually crafted similarity metrics, GAN-based approaches use a generator network to generate the deformed moving image and apply a discriminator network to decide whether the deformed image is similar to the target image [39], [40]. GANs can recover a more complex range of deformations, and the similarity metric is learned automatically based on the discrimination network.

Another kind of generative model in unsupervised image registration is VAE. A VAE aims to avoid overfitting and ensure that the latent space has good properties to generate new data. In image registration, latent variables, such as a set of parameters of deformations or a low-dimensional vector, are defined, and new images can be generated using the sampling of latent variables. Krebs *et al.* [6] and Dalca *et al.* [7] employed VAE to implement image registration.

Unsupervised learning methods show more potential applications than supervised learning methods because no annotated data are required in the training process. However, this kind of approach relies heavily on modeling image similarity. More sophisticated similarity metrics and regularization are needed to improve the performance of unsupervised learning approaches.

Ma *et al.* [42] and Cao *et al.* [43] reviewed a lot of feature-based image matching works and learning-based deformation registration algorithms that include a lot of DL-based methods using RBFs. Most of DL-based methods using RBFs to perform registration employed B-splines [10]–[12] or TPSs [13]–[16] as the interpolation functions. Krebs *et al.* [10] used an artificial agent by choosing from a set of actions to optimize the parameters of cubic B-splines deformation in DL-based ROI-specific deformable registration. De Vos *et al.* [11] and [12] employed a cubic B-splines transformer to generate a DVF in the DIRNet. Wu *et al.* [13] used a convolutional stacked autoencoder to discover deep feature representations and identified their anatomical correspondences by matching the representations to each key point. Then, TPSs were employed to interpolate the dense DVF. Cao *et al.* [14] proposed a CNN regressor to directly learn the parameters of TPS with the equalized active-point guided sampling strategy and an auxiliary contextual cue. Later, this work was extended by proposing a cue-aware deep regression network [15]. Though these works achieved comparable performance to the conventional methods, most of these existed works are either supervised methods or patchwised registration.

### *C. Probabilistic Learning for Image Registration*

Variational inference is a method for approximating the intractable posterior distribution of latent variables in the generative process. Kingma and Welling [17] proposed the VAE to approximate the posterior distribution of latent variables by combining the variational inference and the deep autoencoder. VAE typically consists of a probabilistic encoder and a probabilistic decoder. The encoder approximates the posterior distribution of the latent variable to the prior, and the decoder produces the reconstructed data with the sampled representation. VAE is a generative model and is suitable for a large dataset. The conditional variational autoencoder

(CVAE) [18] extends VAE by introducing conditioned labels, which can be used in semisupervised learning. In theory, image registration can be regarded as a generative process that warps the moving image to match the fixed image statistically using transformation parameters. When the distribution of transformation parameters is estimated, the parameters for the new data can be predicted using the probability distribution.

So far, very few researchers have explored the feasibility of the variational inference to perform medical image registration. Dalca *et al.* [7] made the posterior of the stationary velocity field a multivariate normal distribution and leveraged a CNN with diffeomorphic integration and spatial transform layers. Krebs *et al.* [6] and [19] regularized a low dimensional encoding to approximate multivariate normal distribution of DVFs. However, all these works are based on dense DVFs.

## III. Methods

### A. Image Registration Based on Generative Model

A generative model aims to learn data distribution from training samples using unsupervised learning to generate new data with some variability. Compared with discriminative models, generative models often perform better on smaller datasets because they learn the underlying data structure and place some structure assumptions on models to prevent over-fitting. When the test data are generated by different underlying distributions instead of the training data, it is easier to adjust the generative model to fit this change in an unsupervised learning way. An additional benefit of the generative model is that it can predict the uncertainty of a decision.

In image registration, the training data are usually limited due to the expensive annotation process. Generative models have the potential to learn probabilistic deformation models in image registration using the small training dataset. In addition, the generative model can explain the uncertainty of registration results.

Image registration can be represented as a generative model in the sense of probability. The fixed image $F$ can be regarded as the one generated from the moving image $M$ deformed by a spatial transformation $\boldsymbol{\phi}_z$, where $z$ is the parameter of the spatial transformation. $z$ can also be the latent variable in the generative model. The fixed image $F$ can be regarded as a sample from a random distribution generated from the latent variable $z$ with the condition of the moving image $M$. The generative procedure consists of two steps [17]: 1) a value of the latent variable $z$ is generated from its prior distribution $p(z)$ and 2) the fixed image $F$ is generated from the distribution $p_{\boldsymbol{\theta}}(F|z, M)$ with generative parameters $\boldsymbol{\theta}$. In general, the prior distribution $p(z)$ is assumed given as a Gaussian distribution.

The variational inference technique can be employed to solve the image registration problem based on the generative model. Since $p_{\boldsymbol{\theta}}(F|M) = \int p_{\boldsymbol{\theta}}(F|z, M)p(z)\mathrm{d}z$ is intractable in computation, which also makes the posterior $p_{\boldsymbol{\theta}}(z|F, M) = ((p_{\boldsymbol{\theta}}(F|z, M)p(z))/(p_{\boldsymbol{\theta}}(F|M)))$ intractable, variational inference transforms the marginal likelihood of the fixed image $F$ into

$$\begin{aligned}
&\log p_{\boldsymbol{\theta}}(F|M)\\
&\quad= \mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[\log p_{\boldsymbol{\theta}}(F|M)]\\
&\quad= \mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[\log p_{\boldsymbol{\theta}}(F, z|M) - \log p_{\boldsymbol{\theta}}(z|F, M)\\
&\qquad\qquad - \log q_{\boldsymbol{\beta}}(z|F, M) + \log q_{\boldsymbol{\beta}}(z|F, M)] \quad (1)\\
&\quad= \mathrm{ELBO} + \mathrm{KL}[q_{\boldsymbol{\beta}}(z|F, M) \| p_{\boldsymbol{\theta}}(z|F, M)]
\end{aligned}$$

$$\begin{aligned}
&\mathrm{ELBO}\\
&\quad= \mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[\log p_{\boldsymbol{\theta}}(F, z|M) - \log q_{\boldsymbol{\beta}}(z|F, M)]\\
&\mathrm{KL}[q_{\boldsymbol{\beta}}(z|F, M) \| p_{\boldsymbol{\theta}}(z|F, M)]\\
&\quad= \mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}\left[\log \frac{q_{\boldsymbol{\beta}}(z|F, M)}{p_{\boldsymbol{\theta}}(z|F, M)}\right]. \quad (2)
\end{aligned}$$

In (1), $p_{\boldsymbol{\theta}}(F, z|M)$ represents the distribution of generated image and the latent variable $z$ given the moving image $M$. The posterior $p_{\boldsymbol{\theta}}(z|F, M)$ represents the distribution of the latent variable $z$ given the fixed image $F$ and the moving image $M$. Variational inference introduces the variational posterior $q_{\boldsymbol{\beta}}(z|F, M)$ parametrized by $\boldsymbol{\beta}$ to approximate the posterior $p_{\boldsymbol{\theta}}(z|F, M)$. Therefore, the variational inference aims to maximize $p_{\boldsymbol{\theta}}(F|M)$ and minimize the KL divergence between $q_{\boldsymbol{\beta}}(z|F, M)$ and $p_{\boldsymbol{\theta}}(z|F, M)$. It transforms the maximization of marginal likelihood and minimization of the KL divergence into maximizing ELBO [44]. ELBO is denoted as the evidence lower bound.

Contrary to the mean-field variational inference, no factorial variational distribution is assumed, and no closed-form expectation is needed in this article. Instead, we employ an autoencoder proposed by Kingma and Welling [17] to estimate the optimal parameter of the variational distribution.

### B. Spatial Transformation Based on CSRBFs

In this article, we employ the CSRBF-based transformation function to deform images. Given the coefficients of control points, the CSRBF-based transformation interpolates the DVF using CSRBFs. Let $\boldsymbol{\phi}_z : \mathbb{R}^2 \to \mathbb{R}^2$ be the spatial transformation between the moving image $M$ and the fixed image $F$. The transformation function $\boldsymbol{\phi}_z$ is composed of $\boldsymbol{\phi}_{z,x}$ and $\boldsymbol{\phi}_{z,y}$, which are transformation functions in $x$- and $y$-directions, respectively. Taking the $x$-direction as an example, given a point set of $n$ control points $P = \{p_i\}_{i=1}^n$, the transformation function along the $x$-direction is defined as

$$\boldsymbol{\phi}_{z,x}(\boldsymbol{o}) = \boldsymbol{o}_x + \sum_{i=1}^{n} z_{i,x}\,\boldsymbol{\psi}\left(\frac{\|\boldsymbol{o} - p_i\|}{r}\right) \quad (3)$$

where $\boldsymbol{o} = (\boldsymbol{o}_x, \boldsymbol{o}_y) \in \mathbb{R}^2$ is the pixel; $z_{i,x}$ is the coefficient corresponding to the control point $p_i$ along the $x$-direction. For the $y$-direction, the transformation function $\boldsymbol{\phi}_{z,y}$ is the same to that of $x$-direction, only different in $\boldsymbol{o}_y$ and the coefficient $z_{i,y}$. $\| \boldsymbol{o} - p_i \|$ is the Euclidean distance between $\boldsymbol{o}$ and $p_i$; $\boldsymbol{\psi}(\cdot)$ is the CSRBF with support $r$. The CSRBF-based transformation uses the proper coefficients to interpolate a dense DVF and controls the deformation locally. We employ the Wendland function $\boldsymbol{\psi}(\xi) = (1 - \xi)_+^4(4\xi + 1)$ in (3), where $(x)_+ = \max(0, x)$. The support $r$ is set as

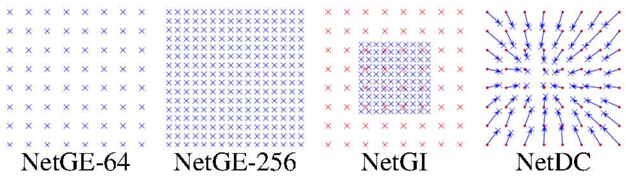


Fig. 1. Distributions of control points. From left to right: evenly distributed control points containing 64 and 256 control points, respectively; the given nonevenly distributed control points; the drifting control points, where the red and blue ones are initial and shifted control points, respectively; and the arrows mark the shifting of each control point. The bottom line marks the corresponding network.

$r = 2\max_{p_i \in P} \min_{p_j \in P-\{p_i\}} \|p_i - p_j\|$ to make $r$ adaptive with respective to the distribution of control points.

Fig. 1 shows the various distribution of control points used in our model. The first two schemes show 64 and 256 evenly spaced control points, respectively. The third one is the nonevenly distributed control points. CSRBF-based transformations with nonevenly spaced control points are flexible when control points are positioned densely in the regions with large deformation. They can achieve more accurate results than the transformations with evenly spaced control points. However, how to place control points properly is an issue to be handled. The last one in Fig. 1 demonstrates the drifting control points, where the red ones are initial control points, and the arrows mark the shifting of each control point. The shifted positions marked by blue are drifting control points used in CSRBF-based transformations. We propose an approach to drift control points to proper positions adaptively using a neural network; details can be referred to in Section III-D.

### *C. Probabilistic Model*

The registration process can be treated as a generative process to find the distribution of $z_i = \{z_{i,x}, z_{i,y}\}$ for each control point $p_i$. Especially for drifting control points, finding the location and coefficient distribution for each control point simultaneously is required.

We consider the coefficient vector $z = \{z_{1,x}, z_{2,x}, \ldots, z_{n,x}, z_{1,y}, z_{2,y}, \ldots, z_{n,y}\}$ as the latent variable in the generative process and approximate its posterior $p(z|F, M)$ using variational inference. We assume that the prior $p(z)$ is a multivariate normal distribution with zero mean and a covariance $\mathbf{\Sigma}_P$

$$p(z) = \mathcal{N}(z; 0, \mathbf{\Sigma}_P) \tag{4}$$

where $\mathbf{\Sigma}_P$ is defined as follows in (5) and (6), as shown at the bottom of the page, where $\mathbf{0}$ is the zero matrix. The prior $p(z)$ is different for different control point sets.

VAE is a variational Bayesian method with a multivariate distribution as prior and a posterior approximated by an artificial neural network. A typical VAE has an encoder $q_{\boldsymbol{\beta}}(z|F, M)$ with variational parameters $\boldsymbol{\beta}$ and a decoder $p_{\boldsymbol{\theta}}(F|z, M)$ with generative parameters $\boldsymbol{\theta}$. The encoder produces a distribution over the latent variable $z$ from which the fixed image $F$ is likely to be generated with the condition of the moving image $M$. The decoder produces a distribution over the possible fixed image $F$ corresponding to the latent variable $z$ and the condition $M$. To generate $z$ corresponding to $F$ and $M$, the posterior $p_{\boldsymbol{\theta}}(z|F, M)$ needs to be calculate. Following the variational method [17], we introduce a variational distribution $q_{\boldsymbol{\beta}}(z|F, M)$ with parameters $\boldsymbol{\beta}$ to approximate the intractable posterior $p(z|F, M)$. Let the approximation of $q_{\boldsymbol{\beta}}(z|F, M)$ be a multivariate normal distribution

$$q_{\boldsymbol{\beta}}(z|F, M) = \mathcal{N}(z; \boldsymbol{\mu}(F, M), \mathbf{\Sigma}(F, M)) \tag{7}$$

where the mean $\boldsymbol{\mu}(F, M)$ and the covariance $\mathbf{\Sigma}(F, M)$ are related to the moving image $M$ and the fixed image $F$. To simplify the analysis and avoid massive computation, the covariance $\mathbf{\Sigma}(F, M)$ is assumed as a diagonal matrix.

The encoder using a CNN with parameters $\boldsymbol{\beta}$ estimates the mean $\boldsymbol{\mu}(F, M)$ and the diagonal covariance $\mathbf{\Sigma}(F, M)$ of $q_{\boldsymbol{\beta}}(z|F, M)$. Our model aims to find the optimal parameters $\boldsymbol{\beta}^*$. To make the process of sampling differentiable, the latent variable $z$ is sampled using the reparameterization trick as $z = \boldsymbol{\mu}(F, M) + \mathbf{\Sigma}(F, M) * \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$. Furthermore, when the distribution of latent variable $z$ is known, the distribution $p_{\boldsymbol{\theta}}(F|z, M)$ is referred as the pseudo-decoder to warp the moving image $M$. Note that additional network layers with parameters $\boldsymbol{\theta}$ in the decoder are not needed in our framework because the spatial transformation is known when the latent variable $z$ is given. Therefore, $p_{\boldsymbol{\theta}}(F|z, M)$, $p_{\boldsymbol{\theta}}(F|M)$, $p_{\boldsymbol{\theta}}(z|F, M)$ are denoted as $p(F|z, M)$, $p(F|M)$, $p(z|F, M)$, and $p(F|M) = \int p(F|z, M)p(z)dz$ is a constant with respect to $q_{\boldsymbol{\beta}}(z|F, M)$. Correspondingly, maximizing ELBO is equivalent to minimizing $\mathrm{KL}[q_{\boldsymbol{\beta}}(z|F, M) \| p(z|F, M)]$. It is an essential feature of our framework because no additional network layers in the decoder imply no training for the decoder parameters. It simplifies the training network and makes the network lightweight.

$$\mathbf{\Sigma}_P^{-1} = \begin{bmatrix} \boldsymbol{B} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B} \end{bmatrix} \tag{5}$$

$$\boldsymbol{B} = \begin{bmatrix} \psi\left(\frac{\|p_1 - p_1\|}{r}\right) & \psi\left(\frac{\|p_1 - p_2\|}{r}\right) & \cdots & \psi\left(\frac{\|p_1 - p_n\|}{r}\right) \\ \psi\left(\frac{\|p_2 - p_1\|}{r}\right) & \psi\left(\frac{\|p_2 - p_2\|}{r}\right) & \cdots & \psi\left(\frac{\|p_2 - p_n\|}{r}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \psi\left(\frac{\|p_n - p_1\|}{r}\right) & \psi\left(\frac{\|p_n - p_2\|}{r}\right) & \cdots & \psi\left(\frac{\|p_n - p_n\|}{r}\right) \end{bmatrix} \tag{6}$$

We maximize the ELBO to derive the loss function

$$\begin{aligned}
&\max \mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[\log p(F,z|M) - \log q_{\boldsymbol{\beta}}(z|F,M)]\\
&\Rightarrow \min \mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[\log q_{\boldsymbol{\beta}}(z|F,M) - \log p(F,z|M)]\\
&\Rightarrow \min \mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[\log q_{\boldsymbol{\beta}}(z|F,M) - \log p(F|z,M)\\
&\qquad\qquad - \log p(z|M)]\\
&\Rightarrow \min \mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[\log q_{\boldsymbol{\beta}}(z|F,M) - \log p(F|z,M)\\
&\qquad\qquad - \log p(z)]\\
&\Rightarrow \min \mathrm{KL}[q_{\boldsymbol{\beta}}(z|F,M)\|p(z)] - \mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[\log p(F|z,M)].
\end{aligned} \tag{8}$$

Here, we assume that $z$ and $M$ are independent of each other. The first term $\mathrm{KL}[q_{\boldsymbol{\beta}}(z|F,M)\|p(z)]$ forces the variational distribution $q_{\boldsymbol{\beta}}(z|F,M)$ to approximate to the prior $p(z)$, which is computed in closed form as follows:

$$\begin{aligned}
&\mathrm{KL}[q_{\boldsymbol{\beta}}(z|F,M)\|p(z)]\\
&\quad= \frac{1}{2}\big( - \log|\boldsymbol{\Sigma}(F,M)| + \log|\boldsymbol{\Sigma_P}| + \mathrm{tr}\big(\boldsymbol{\Sigma}_P^{-1}\boldsymbol{\Sigma}(F,M)\big)\\
&\qquad\qquad + \boldsymbol{\mu}(F,M)^T\boldsymbol{\Sigma}_P^{-1}\boldsymbol{\mu}(F,M) - n\big).
\end{aligned} \tag{9}$$

Since all diagonal elements of $\boldsymbol{\Sigma}_P^{-1}$ are 1, and $\boldsymbol{\Sigma}(F,M)$ is a diagonal matrix, $\mathrm{tr}(\boldsymbol{\Sigma}_P^{-1}\boldsymbol{\Sigma}(F,M)) = \mathrm{tr}(\boldsymbol{\Sigma}(F,M))$. Then, (9) can be simplified as

$$\begin{aligned}
&\mathrm{KL}[q_{\boldsymbol{\beta}}(z|F,M)\|p(z)]\\
&\quad= \frac{1}{2}\big( - \log|\boldsymbol{\Sigma}(F,M)| - \log\big|\boldsymbol{\Sigma}_P^{-1}\big| + \mathrm{tr}(\boldsymbol{\Sigma}(F,M))\\
&\qquad\qquad + \boldsymbol{\mu}(F,M)^T\boldsymbol{\Sigma}_P^{-1}\boldsymbol{\mu}(F,M)\big) + \mathrm{const}.
\end{aligned} \tag{10}$$

Note that $\boldsymbol{\mu}(F,M)$ is the mean of $z$, which can be regarded as the value of $z$ during registration procedure; $\boldsymbol{\mu}(F,M)^T\boldsymbol{\Sigma}_P^{-1}\boldsymbol{\mu}(F,M)$ is similar to the BE defined on transformation functions based on CSRBF, which can be used as the BE constraint imposing on DVFs [9]. It is an essential constraint of the loss function that can make the DVF smooth and topology-preserving. Furthermore, it also shows the advantage of introducing CSRBF-based transformations in the probabilistic model. That is, the BE of DVFs can be constrained in a close-formed way.

The distribution $p(F|z,M)$ in the second term in (8) describes the probability of the moving image $M$ deformed to be similar to the fixed image $F$ using latent parameter $z$. It can be represented by the similarity between the warped moving image and the fixed image. When the similarity is high, $p(F|z,M)$ is high; otherwise, it is low. Similar to Krebs *et al.* [19], we employ a local cross correlation (LCC) Boltzmann distribution to be $p(F|z,M)$ with a balance factor $\lambda$: $p(F|z,M) \sim \exp(-\lambda(1-\mathrm{LCC}(F, M(\boldsymbol{\phi}_z))))$. $\mathrm{LCC}(F, M(\boldsymbol{\phi}_z))$ is defined as follows:

$$\frac{1}{N}\sum_{\boldsymbol{o}\in\Omega}\frac{\big(\sum_{\boldsymbol{o}_k\in L_{\boldsymbol{o}}}\overline{F}(\boldsymbol{o}_k)\overline{M}(\boldsymbol{\phi}_z(\boldsymbol{o}_k))\big)^2}{\big(\sum_{\boldsymbol{o}_k\in L_{\boldsymbol{o}}}\overline{F}(\boldsymbol{o}_k)^2\big)\big(\sum_{\boldsymbol{o}_k\in L_{\boldsymbol{o}}}\overline{M}(\boldsymbol{\phi}_z(\boldsymbol{o}_k))^2\big)} \tag{11}$$

where $N$ is the number of pixels, $\Omega$ is the image field, and $\overline{F}(\boldsymbol{o}_k)$ and $\overline{M}(\boldsymbol{\phi}_z(\boldsymbol{o}_k))$ are intensities subtracted by average intensities over the local region $L_{\boldsymbol{o}}$ centered at the pixel $\boldsymbol{o}$.

The second term $\mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[\log p(F|z,M)]$ in (8) can be approximated using the Monte Carlo method

$$\begin{aligned}
&\mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[\log p(F|z,M)]\\
&\quad= \mathbb{E}_{q_{\boldsymbol{\beta}}(z|F,M)}[-\lambda(1 - \mathrm{LCC}(F, M(\boldsymbol{\phi}_z)))]\\
&\quad\simeq \frac{\lambda}{K}\sum_{k=1}^{K}\mathrm{LCC}(F, M(\boldsymbol{\phi}_{z^k})) + \mathrm{const}
\end{aligned} \tag{12}$$

where $z^k$ is the $k$th sampled value of $z$ using reparameterization trick, and $K$ is the total number of samples, which is set to 1 in our experiments. By eliminating the constant, the total loss function $\mathcal{L}_{\mathrm{LCC}}(F,M,P)$ is defined as

$$\begin{aligned}
&\frac{1}{2}\big( - \log|\boldsymbol{\Sigma}(F,M)| + \mathrm{tr}(\boldsymbol{\Sigma}(F,M)) - \log\big|\boldsymbol{\Sigma}_P^{-1}\big|\\
&\qquad + \boldsymbol{\mu}(F,M)^T\boldsymbol{\Sigma}_P^{-1}\boldsymbol{\mu}(F,M)\big) - \frac{\lambda}{K}\mathrm{LCC}(F, M(\boldsymbol{\phi}_{z^k})).
\end{aligned} \tag{13}$$

Besides, for comparison, we also try to use a MSE Boltzmann distribution to be $p(F|z,M)$ with a balance factor $\lambda$: $p(F|z,M) \sim \exp(-\lambda\mathrm{MSE}(F, M(\boldsymbol{\phi}_z)))$. The total loss function $\mathcal{L}_{\mathrm{MSE}}(F,M,P)$ can be derived as

$$\begin{aligned}
&\frac{1}{2}\big( - \log|\boldsymbol{\Sigma}(F,M)| + \mathrm{tr}(\boldsymbol{\Sigma}(F,M)) - \log\big|\boldsymbol{\Sigma}_P^{-1}\big|\\
&\qquad + \boldsymbol{\mu}(F,M)^T\boldsymbol{\Sigma}_P^{-1}\boldsymbol{\mu}(F,M)\big) + \frac{\lambda}{K}\mathrm{MSE}(F, M(\boldsymbol{\phi}_{z^k})).
\end{aligned} \tag{14}$$

Note that this loss function is used for comparison with the loss function $\mathcal{L}_{\mathrm{LCC}}(F,M,P)$. Our models are trained with $\mathcal{L}_{\mathrm{LCC}}(F,M,P)$ in default unless specified.

### *D. Networks for Estimating Probabilistic Coefficients and Drifting Control Points*

Considering that the latent variables correspond to the coefficients of CSRBFs, we construct a pseudoautoencoder architecture network to estimate the probabilistic parameters $\boldsymbol{\mu}(F,M)$ and $\boldsymbol{\Sigma}(F,M)$ of coefficients. The encoder is a CNN to predict the distribution parameters of coefficients, and the decoder consists of a CSRBF-based transformation layer that interpolates the DVFs and an STN to warp the moving image $M$ using the interpolated DVFs. We design two networks for the scenario of fixed control points and drifting control points, respectively. For comparison, we also develop two other networks for the cases with evenly spaced control points.

*1) Networks With Fixed Control Points:* For the scenario of fixed control points, as shown in Fig. 1, 64 evenly spaced control points are placed globally, and 100 densely spaced control points are placed in the central region used to deform the local area imaging the cardiac structure delicately. To estimate the probabilistic parameters $\boldsymbol{\mu}(F,M) = \{\boldsymbol{\mu}_g, \boldsymbol{\mu}_c\}$ and $\boldsymbol{\Sigma}(F,M) = \{\boldsymbol{\Sigma}_g, \boldsymbol{\Sigma}_c\}$, where $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ are parameters for global control points, and $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are parameters for control points located in the central region, a network named NetGI is constructed. NetGI consists of Module A and Module B that aim to predict the parameters of 64 global control points and 100 local control points, respectively.

As shown in Fig. 2, Module A takes the image pair of the moving image $M$ and the fixed image $F$ as the inputs and contains two convolutions with 16 kernels, followed by

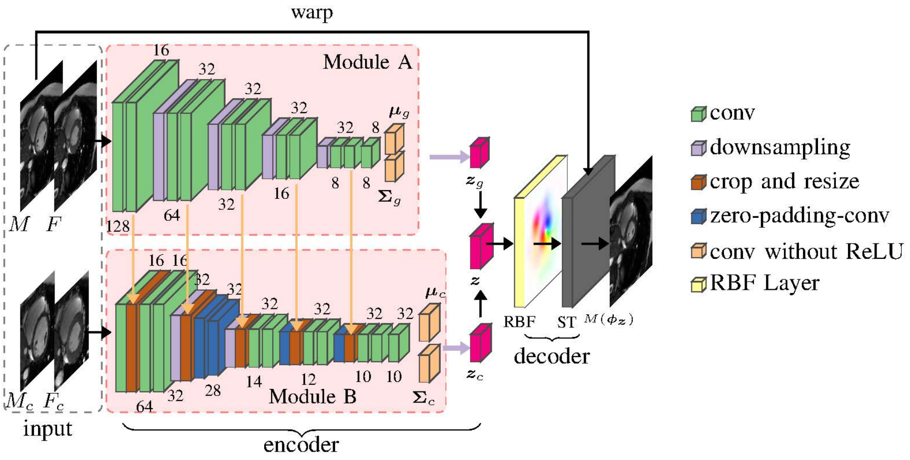


Fig. 2. Architecture of network NetGI. The encoder consists of Module A and Module B. Module A and Module B are used to predict the probabilistic parameters of global control points and central control points, respectively. The decoder consists of an RBF-based transformation layer and a spatial transform layer STN. $M$ and $F$ are the inputs of Module A. $M_c$ and $F_c$ are the cropped image pair, which is the inputs of Module B.

four blocks consisting of a downsampling convolution with 32 kernels and a stride of 2, and two convolutions with 32 kernels. A convolution with eight kernels follows them, and two convolutions output the mean $\boldsymbol{\mu}_g$ and the covariance $\boldsymbol{\Sigma}_g$ of the global control points, respectively.

Module B learns the probabilistic parameters $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ corresponding to the control points located in the central region. We crop the central parts of the image pair in $64 \times 64$ to be the inputs of Module B. Similar to Module A, Module B also contains five blocks. The first block starts with the convolution of input, concatenated by Module A's cropped and resized feature map output, followed by two convolutions. The other blocks are similar to the first block except in feature map dimensions. The convolutions without padding are used in the second block, and no padding is performed in the downsampling in the last two blocks. The dimensions of convolutions in the first blocks and other blocks are 16 and 32, respectively. Module B ends with a convolution with 8 kernels and predicts the mean $\boldsymbol{\mu}_c$ and the covariance $\boldsymbol{\Sigma}_c$ of the central control points. A leaky ReLu layer follows all convolutions except the last one.

All the predicted parameters $\boldsymbol{\mu}_g, \boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_g, \boldsymbol{\Sigma}_c$ are concatenated as the distribution parameters of latent variable $z$. After sampling $z$ using the reparameterization trick, we use a CSRBF-based transformation layer to generate $\boldsymbol{\phi}_z$ and then use an STN to warp the moving image $M$ using $\boldsymbol{\phi}_z$, which is the decoder network.

The training procedure of NetGI is presented in Algorithm 1. In addition, for comparison, we use Module A to perform registration using 64 and 256 evenly spaced control points, denoted as NetGE-64 and NetGE-256, respectively. Noted that the last block in Module A is removed to be NetGE-256.

**Algorithm 1** Training NetGI for Image Registration

```
$\boldsymbol{\beta} \leftarrow$ Initialize parameters
**repeat**
    $F, M \leftarrow$ Random minibatch of pairs of the fixed image
        and the moving image
    $\boldsymbol{\mu}(F, M), \boldsymbol{\Sigma}(F, M) \leftarrow$ Encoder of NetGI $\mathrm{E}_{\boldsymbol{\beta}}(F, M)$
    $z \leftarrow \boldsymbol{\mu}(F, M) + \boldsymbol{\Sigma}(F, M) * \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$
    $\boldsymbol{\phi}_z, M(\boldsymbol{\phi}_z) \leftarrow$ Decoder of NetGI
    $\mathrm{g} \leftarrow \nabla_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}; F, M, \boldsymbol{\epsilon})$
    $\boldsymbol{\beta} \leftarrow$ Update parameters using gradients g
**until** convergence
**return** $\boldsymbol{\beta}$
```

*2) Network With Drifting Control Points:* For the case of drifting control points, the distribution parameters $\boldsymbol{\mu}(F, M), \boldsymbol{\Sigma}(F, M)$, and locations of control points are estimated simultaneously. We assume that the number of control points is fixed, and the initial control points are evenly spaced on grids. A CNN named NetDC is designed with two branches to estimate the displacements of initial control points and corresponding probabilistic parameters simultaneously. The critical part of NetDC is how to obtain the probabilistic parameters corresponding to these drifting control points. We employ a spatial transformer layer to warp the probabilistic parameter feature maps to match drifting control points. The architecture of NetDC is illustrated in Fig. 3, which is also composed of an encoder and a decoder. The encoder of NetDC has two kinds of modules: 1) Module F outputs features of the probabilistic parameters and the features of drifting control points and 2) Module O outputs displacements of initial control points and the probabilistic parameters of these control points.

The architecture of Module F is illustrated in Fig. 3(a), which consists of two branches. Both branches take the image pair of $M$ and $F$ as inputs. The bottom branch extracts features of the probabilistic parameters. It contains two convolutions with $d_i$ kernels and a downsampling convolution with a stride of 2. The top branch analyses features of the displacements of initial control points. Two convolutions with $d_i$ kernels and convolutions with two kernels follow the input. Note that the convolution with two kernels generates the DVF. Next, a spatial transformer layer warps the feature generated in the bottom branch using the DVF. Finally, the warped feature

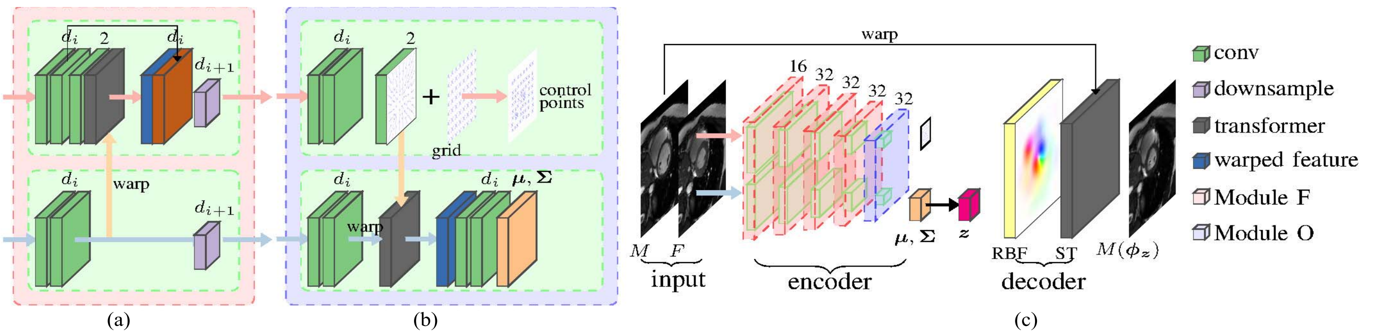


Fig. 3. Network for RBF mesh deformation with a drifting distribution of control points (NetDC). NetDC takes the image pair of $M$ and $F$ as the inputs. The encoder of NetDC is composed of four Module F's and one Module O. Module F analyzes the features of correspondence between the image pair and the features of the offsets of control points. Module O outputs the offsets of control points and the coefficients of control points concerning the control points. (a) Module F. (b) Module O. (c) Architecture of NetDC.

and the original feature in the top branch are concatenated. A downsampling convolution with $d_{i+1}$ follows to output the features of the displacements of initial control points, which also is the input of the next module.

Module O is illustrated in Fig. 3(b). Similar to Module F, there are also two branches. The top branch includes two convolutions with $d_i$ kernels and convolution with two kernels to output the displacements of initial control points. By adding the displacements and positions of initial control points, the drifting control points are obtained. In the bottom branch, two convolutions with $d_i$ kernels analyze the features of the probabilistic parameters. Next, a spatial transformer layer, followed by two convolutions, maps the displacements of initial control points to align them with features of drifting control points. Finally, a convolution layer is used to predict probabilistic parameters concerning these drifting control points.

The whole network architecture is illustrated in Fig. 3(c), which contains four successive Module F, one Module O, and a decoder. Here, the decoder is similar to that in NetGI. In our implementation, the number of the drifting control points is 64, and $d_i \in \{16, 32, 32, 32, 32\}$, $i = 0, 1, 2, 3, 4$. A leaky ReLu layer follows all the convolutions except the convolution layer generating the displacements or probabilistic parameters. The training procedure of NetDC is described in Algorithm 2.

**Algorithm 2** Training NetDC for Image Registration

```
$\boldsymbol{\beta} \leftarrow$ Initialize parameters
$P^0 \leftarrow$ Initialize control points
**repeat**
    $F, M \leftarrow$ Random minibatch of pairs of the fixed image and the moving image
    $\boldsymbol{\mu}(F, M), \boldsymbol{\Sigma}(F, M), \boldsymbol{u} \leftarrow$ Encoder of NetDC $\mathrm{E}_{\boldsymbol{\beta}}(F, M)$
    $\boldsymbol{z} \leftarrow \boldsymbol{\mu}(F, M) + \boldsymbol{\Sigma}(F, M) * \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$
    $P \leftarrow P^0 + d(P^0)$ (offsets of control points)
    $\boldsymbol{\phi}_z, M(\boldsymbol{\phi}_z) \leftarrow$ Decoder of NetDC
    $\mathrm{g} \leftarrow \nabla_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}; F, M, \boldsymbol{\epsilon}, P^0)$
    $\boldsymbol{\beta} \leftarrow$ Update parameters using gradients g
**until** convergence
**return** $\boldsymbol{\beta}$
```

## IV. EXPERIMENTS

### A. Implementation

*1) Data:* Three public datasets were employed in the experiments to evaluate the proposed methods, including the MICCAI2009 challenge dataset provided by the Sunnybrook Health Sciences Center [45], the York dataset provided by the Department of Diagnostic Imaging of the Hospital for Sick Children in Toronto (York) [46], and the dataset from automatic cardiac diagnosis challenge (ACDC) at STACOM 2017 [47]. There are in total 228 cardiac short-axis cine-MR image cases in three datasets. Table I lists the details of three datasets.

*2) Implementation Details:* In our experiments, two image slices at the end-systolic (ES) phase and the end-diastolic (ED) phase in one cardiac cycle were registered to each other. The image at ED was the moving image, and the image at ES was the fixed image. 1257 image pairs from 136 randomly selected cases were the training samples, 130 image pairs from 16 randomly selected cases were the evaluation samples, and 698 image pairs from the remaining 76 cases were the testing samples. The image slices were cropped as the size of 128 × 128 covering the cardiac structure well. Data augmentation was performed by rotation, scaling, flipping, and shifting with random parameters. The size of the local area in LCC was 9 × 9. The output of our networks $\boldsymbol{\mu}(F, M)$ was used as the coefficients to estimate DVFs.

The proposed four networks, NetGE-64, NetGE-256, NetGI, and NetDC, were trained using training samples. The control points for four networks were 64, 256, 164, and 64, respectively. Our networks were trained using PyTorch [48] on a computer equipped with a Xeon(R) W-2123 CPU and Nvidia GTX 1080Ti GPU. The Adam optimizer [49] with a learning rate of $1e^{-4}$ was employed.

*3) Evaluation Metrics:* To evaluate the proposed method, the contours at the ED phase provided by experts were mapped to the contours at the ES phase using estimated DVFs, and the mapped contours were compared with the ground-truth contours at ES using various metrics. Since the ground truths of different datasets are different from each other, the evaluation contours included the blood pool of LV (LV-BP), the myocardium of LV (LV-Myo), the epicardium of LV (LV-Epi),

TABLE I
DETAILS OF MICCAI2009, YORK, AND ACDC

| | MICCAI2009 | York | ACDC |
|---|---|---|---|
| Number of cases | 15 training cases, 15 testing cases and 15 online cases | 33 cases | 150 training cases and 50 testing cases |
| Number of slices per case | 6 to 12 | 8 to 15 | 8 to 12 |
| Slice Size | 256 × 256 | 256 × 256 | The size is unfixed |
| Number of phases | 20 phases | 20 phases | 28 to 40 phases |
| Slice spacing | 1.21 to 1.56 mm | 0.9375 to 1.6406 mm | 0.8333 to 1.9531 mm |
| Slice thickness | 6 to 13 mm | 6 to 10 mm | 5 or 10 mm |
| ED and ES phase | labeled by experts | labeled by experts | labeled by experts |
| Annotations | endocardium and epicardium of LV at ED, endocardium at ES | endocardium and epicardium of LV at all phases | epicardium of right ventricle (RV), endocardium and epicardium of LV at ES and ED |
| Training data | 15 training cases and 15 online cases | 22 cases | 50 training cases and 50 testing cases |
| Testing data | 15 testing cases | 11 cases | 50 training cases |
| Evaluation contours | LV-BP | LV-BP, LV-Myo and LV-Epi | LV-BP, LV-Myo, LV-Epi and RV-Epi |

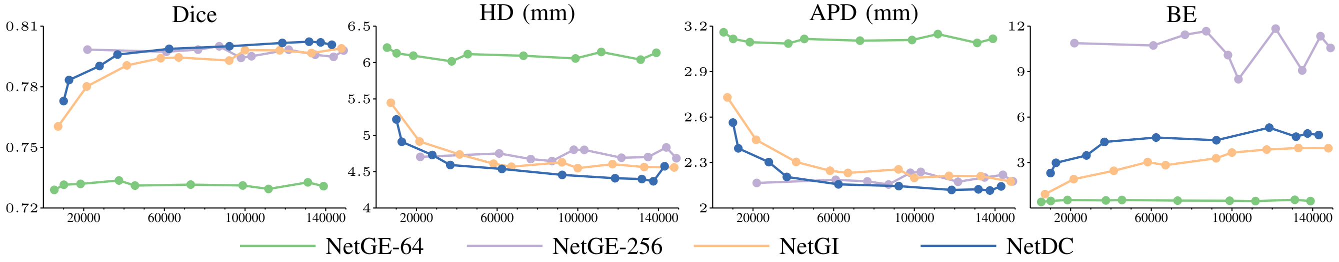


Fig. 4. Evaluation results for different networks with different values of $\lambda$.

and the epicardium of the right ventricle (RV-Epi). Details of evaluation contours for three datasets are also listed in Table I.

The mapped contours were evaluated by three measures, including the Dice score [50], the average perpendicular distance (APD, in mm), and 95%-tile Hausdorff Distance (HD, in mm). The Dice score measures the overlap between two areas enclosed by two contours; APD and HD measure the distance between two contours.

Moreover, the number of nonpositive Jacobian determinants ($|J_{\boldsymbol{\phi}_z}| \leq 0$) was counted to validate the topology-preserving deformation fields. The Jacobian matrix at pixel $\boldsymbol{o} \in \mathbb{R}^2$ is defined as follows:

$$J_{\boldsymbol{\phi}_z}(\boldsymbol{o}) = \begin{bmatrix} \frac{\partial \boldsymbol{\phi}_{z,x}}{\partial x}, \frac{\partial \boldsymbol{\phi}_{z,y}}{\partial x} \\ \frac{\partial \boldsymbol{\phi}_{z,x}}{\partial y}, \frac{\partial \boldsymbol{\phi}_{z,y}}{\partial y} \end{bmatrix}. \tag{15}$$

The Jacobian determinant at a position in a DVF describes how a small local region changes after deformation [7], [51]. When the nonlinear transformation function $\boldsymbol{\phi}_z : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ deforms a small square to a parallelogram, the Jacobian matrix can describe the parallelogram, and the Jacobian determinant provides the ratio of the area of the parallelogram to that of the original square. A negative determinant of the Jacobian matrix around a pixel implies that the mapping may not be one-to-one and noninvertible, which means that the mapping is not topology-preserving. For cardiac motion, a region with negative determinants means the deformation is abnormal and not realistic. A small number of nonpositive Jacobian determinants of a DVF are preferred.

Besides, the BE of DVFs was calculated to measure the smoothness of deformation fields. Taking the $x$-direction for example, the BE in the $x$-direction is calculated as follows:

$$\frac{1}{N}\int\int\left(\left(\frac{\partial^2 \boldsymbol{\phi}_{z,x}}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 \boldsymbol{\phi}_{z,x}}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 \boldsymbol{\phi}_{z,x}}{\partial y^2}\right)^2\right) dxdy. \tag{16}$$

The bending energies in all directions are summed together to be the BE of a DVF.

### *B. Results and Analysis*

*1) Hyperparameter:* The hyperparameter $\lambda$ was used to control the smoothness of the deformation field. Thus, the lower $\lambda$ is, the stronger enforcement to the smoothness is. However, the lower $\lambda$ may degenerate results in registration accuracy. To show the influence of $\lambda$ and explore the relationship between $\lambda$ and registration accuracies, an experiment was performed by setting different values of $\lambda$ to train the networks. The evaluation dataset was tested using four evaluation metrics: Dice, APD, HD, and BE. Fig. 4 shows evaluation results of NetDC, NetGI, NetGE-64, and NetGE-256. It can be seen that, with the increase in the hyperparameter $\lambda$, the registration accuracy tended to improve. However, the smoothness of deformation fields became progressively worse. When $\lambda$ is large to a certain extent, the registration accuracy cannot be improved significantly.

Furthermore, it can be observed that higher registration accuracy can be achieved using more control points by comparing the performance of NetGE-256 and NetGE-64. Although the control points' numbers for NetGE-64 and

NetDC were the same, the registration accuracy of NetDC was much higher than that of NetGE-64. It implies that proper locations of control points greatly influence registration performance, which validates our motivation to shift control points adaptively. NetGE-256 outperformed NetGI in the Dice score and APD because more control points were used for NetGE-256, but NetGI achieved lower HD and generated smoother DVFs. Although the above conclusion is drawn on the evaluation dataset, it is consistent with the conclusion of test samples. The optimal $\lambda$ was 147 000 for NetGI and 130 000 for NetDC based on the best registration accuracy of the evaluation dataset.

*2) Registration:* To compare the performance of our methods with state-of-the-art DL methods, three unsupervised registration networks, KrebsDiff [19], DalcaDiff [7], and VoxelMorph [36], were implemented. KrebsDiff and DalcaDiff are two networks of probabilistic diffeomorphic registration by conducting variational inference, and VoxelMorph is a registration network using unsupervised learning to perform imagewise registration.

We compared our networks, NetGI and NetDC, with KrebsDiff, DalcaDiff, and VoxelMorph. The optimal parameters were used for these networks, such as $\lambda = 60\,000$, and the size of latent variable was 64 for KrebsDiff, $\lambda = 50$ and $\sigma = 0.03$ for DalcaDiff, and $\lambda = 1$ for VoxelMorph. The mean and standard deviations of all metrics for registration results using different networks for datasets, MICCAI2009, York, and ACDC, are listed in Table II. On the MICCAI and York, NetDC was the best in registration performances, and NetGI was second only to NetDC. Although DalcaDiff obtained the best Dice score on the ACDC dataset, NetDC was better on HD and APD. In terms of BE and the number of nonpositive Jacobian determinants, NetGI and NetDC showed better performances on all three datasets. To compare the performance of networks, the mean and standard deviations of all metrics were averaged for all three datasets, as listed in Table II. It is observed that our NetDC achieved the best registration accuracy and generated smooth DVFs, which were nearly one-to-one mappings. Moreover, NetGI and KrebsDiff achieved better HD and APD and generated more regular DVFs than VoxelMorph. However, VoxelMorph received higher Dice scores compared with KrebsDiff.

To summarize the registration results in detail, the boxplots of all metrics are shown in Figs. 5–7. It can be seen that our NetDC and NetGI outperformed other methods significantly in respect of DVF smoothness, which shows the advantages of introducing the smoothing constraint of the CSRBF-based transformation function. From the results of ACDC, we also found that all methods performed better on the registration of LV than on RV because RV is more flexible in shape.

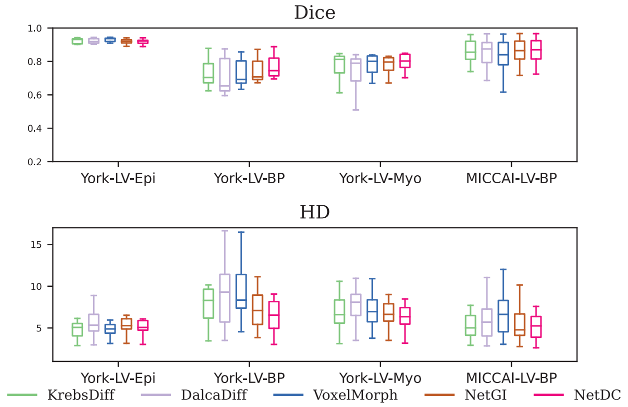


Fig. 5. Boxplots of Dice scores and HDs of registration results using different networks on MICCAI and York datasets.

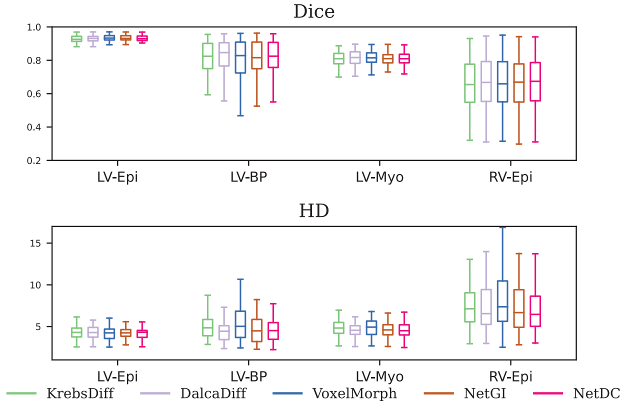


Fig. 6. Boxplots of Dice scores and HDs of registration results using different networks on the ACDC dataset.

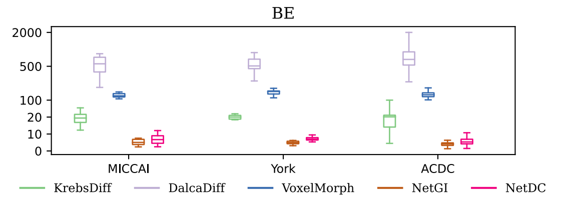


Fig. 7. Boxplots of BEs of registration results using different networks on the ACDC dataset.

The BE of the deformation field is related to the support radius of CSRBFs, the distribution of control points, and the displacements of control points. The number of control points for NetGI was 164, where dense control points were placed in the central area with large deformation. For a fair comparison, we increased the number of control points for NetDC as 13 × 13, denoted as NetDC-169. Correspondingly, the average support radius of NetDC-169 is provided in Table III. Moreover, to compare the BE of the deformation field using different support radius, experimental results of NetGI with $r = 18$ and $r = 32$ are listed in Table III. It can be seen that, although the numbers of control points and support radius of NetGI and NetDC-169 were similar to each other ($r = 18$), NetGI produced smoother deformation fields because of the evenly distributed control points. Besides, generally, larger $r$ improved the smoothness of the deformation field. Based on the results of Table III, it can be concluded that NetGI with 164 control points generated a smooth deformation field because of the evenly distributed control points and larger support radius.

To demonstrate registration results, the deformed images and DVFs for a registration case in ACDC are shown in Fig. 8. The DVFs are visualized using grids and color, respectively. The colored DVFs were drawn using the coding

TABLE II

MEANS AND STANDARD DEVIATIONS OF DICE SCORES, HDS, APDS, BES ($\times 10^{-4}$), AND THE NUMBER OF NONPOSITIVE JACOBIAN DETERMINANTS FOR ALL METHODS

| Dataset | Method | BE[1] | $\lvert J_{\boldsymbol{\phi}_{\boldsymbol{z}}} \rvert \leq 0$ | Dice | HD | APD |
|---|---|---|---|---|---|---|
| MICCAI | KrebsDiff [19] | 26.20(13.80) | 0.33(0.67) | 0.857(0.074) | 5.57(2.16) | 2.60(1.09) |
| | DalcaDiff [7] | 644.84(264.76) | 24.04(21.78) | 0.847(0.090) | 5.78(2.25) | 2.87(1.40) |
| | VoxelMorph [36] | 154.58(24.63) | 22.58(12.87) | 0.829(0.105) | 6.55(2.58) | 3.18(1.57) |
| | NetGI | **5.16**(1.72) | **0** | 0.858(0.081) | 5.48(2.07) | 2.54(1.13) |
| | NetDC | 6.92(2.85) | 1.08(3.37) | **0.861**(0.076) | **5.34**(2.03) | **2.48**(1.09) |
| York | KrebsDiff [19] | 22.65(6.89) | **0** | 0.793(0.093) | 6.64(2.40) | 2.98(1.14) |
| | DalcaDiff [7] | 673.65(348.41) | 47.21(40.85) | 0.776(0.100) | 7.32(2.51) | 3.23(1.23) |
| | VoxelMorph [36] | 194.51(36.04) | 38.06(17.89) | 0.795(0.083) | 7.20(2.36) | 3.05(1.13) |
| | NetGI | **4.89**(0.92) | **0** | 0.798(0.078) | 6.59(2.17) | 2.92(1.07) |
| | NetDC | 6.94(1.50) | **0** | **0.812**(0.078) | **6.25**(2.18) | **2.73**(1.01) |
| ACDC | KrebsDiff [19] | 23.96(16.91) | 1.50(10.47) | 0.800(0.086) | 5.49(1.63) | 2.24(0.67) |
| | DalcaDiff [7] | 895.89(513.56) | 62.61(63.32) | **0.808**(0.087) | 5.33(1.95) | 2.16(0.79) |
| | VoxelMorph [36] | 166.72(37.50) | 27.84(26.08) | 0.804(0.096) | 5.81(2.22) | 2.29(0.98) |
| | NetGI | **4.14**(1.51) | **0.12**(0.75) | 0.798(0.093) | 5.45(1.99) | 2.27(0.87) |
| | NetDC | 5.57(2.52) | 0.16(0.72) | 0.805(0.087) | **5.18**(1.71) | **2.14**(0.72) |
| All | KrebsDiff [19] | 24.21(15.29) | 1.05(8.52) | 0.802(0.132) | 5.65(2.25) | 2.36(0.99) |
| | DalcaDiff [7] | 814.17(466.72) | 52.77(56.60) | 0.806(0.135) | 5.62(2.53) | 2.34(1.17) |
| | VoxelMorph [36] | 168.35(37.03) | 28.28(23.39) | 0.804(0.139) | 6.04(2.81) | 2.45(1.31) |
| | NetGI | **4.45**(1.55) | **0.08**(0.61) | 0.802(0.137) | 5.60(2.42) | 2.37(1.17) |
| | NetDC | 6.03(2.55) | 0.32(1.65) | **0.810**(0.131) | **5.33**(2.17) | **2.24**(0.99) |

[1] Bending Energy to measure the smoothness of deformation.

TABLE III

MEANS AND STANDARD DEVIATIONS OF DICE SCORES, HDS, APDS, BES ($\times 10^{-4}$), AND THE NUMBER OF NONPOSITIVE JACOBIAN DETERMINANTS USING NETDC AND NETGI WITH DIFFERENT SUPPORTS $r$

| Method | $r$ | BE | $\lvert J_{\boldsymbol{\phi}_{\boldsymbol{z}}} \rvert \leq 0$ | Dice | HD | APD |
|---|---|---|---|---|---|---|
| NetGI | 32 | 4.46(1.68) | 0 | 0.805(0.137) | 5.50(2.37) | 2.33(1.12) |
| NetGI | 18 | 7.44(3.22) | 0.50(2.81) | 0.808(0.133) | 5.52(2.39) | 2.29(1.07) |
| NetDC | 26.90 | 6.03(2.55) | 0.32(1.65) | 0.810(0.131) | 5.33(2.18) | 2.24(0.99) |
| NetDC-169 | 18.24 | 14.96(6.66) | 4.87(9.05) | 0.810(0.132) | 5.71(2.53) | 2.32(1.13) |

TABLE IV

MEANS AND STANDARD DEVIATIONS OF DICE SCORES, HDS, APDS, BES ($\times 10^{-4}$), AND THE NUMBER OF NONPOSITIVE JACOBIAN DETERMINANTS FOR DALCADIFF, KREBSDIFF, AND NETDCS TRAINED WITH LCC AND L2 LOSS, RESPECTIVELY

| Method | Similarity Loss | BE | $\lvert J_{\boldsymbol{\phi}_{\boldsymbol{z}}} \rvert \leq 0$ | Dice | HD | APD |
|---|---|---|---|---|---|---|
| KrebsDiff [19] | LCC | 24.21(15.29) | 1.05(8.52) | 0.802(0.132) | 5.65(2.25) | 2.36(0.99) |
| DalcaDiff [7] | MSE | 814.17(466.72) | 52(57) | 0.806(0.135) | 5.62(2.53) | 2.34(1.17) |
| NetDC | LCC | 6.03(2.55) | 0.32(1.65) | 0.810(0.131) | 5.33(2.18) | 2.24(0.99) |
| NetDC | MSE | 9.80(5.46) | 13(27) | 0.803(0.133) | 5.66(2.27) | 2.32(0.95) |

provided by Butler *et al.* [52]. Different colors illustrate different displacements; for example, the green color represents the displacement of the lower left direction. It is observed that our networks produced smoother DVFs compared to that of KrebsDiff, DalcaDiff, and VoxelMorph. This demonstration and comparison of BE in Table II implies that our networks outperformed other networks in terms of the smoothness of DVFs. It validates the advantage of introducing CSRBF-based transformation in the network. That is, the close-formed formulation of BE can be precisely embedded in the cost function.

*3) Loss Comparison:* We applied the MSE loss function to train the network NetDC with different $\sigma$'s. Experimental results showed that, when $\sigma = 3 \times 10^6$, the registration accuracy on the validation dataset was the best. For a detailed comparison, we summarize the results of NetDCs trained with LCC and MSE, respectively, in Table IV. Moreover, results of DalcaDiff [7] and KrebsDiff [19] are also provided, where DalcaDiff was trained with the MSE loss, and KrebsDiff was trained with the LCC loss. It can be seen that the networks trained with LCC loss generated smooth deformation fields and robust registration results (low standard deviations), compared with results obtained by networks trained with MSE loss. Moreover, although DalcaDiff and NetDC used the MSE loss function, NetDC generated smoother and topology-preserving deformation fields, which means that our model has the advantage in deformation smoothness.

*4) Runtime:* We list the runtime of different methods for registering an image pair using GPU and CPU, respectively, in Table V. We implemented KrebsDiff [19], VoxelMorph [36], DalcaDiff [7], and our method using PyTorch [48]. From Table V, we can see that the computation times of KrebsDiff [19], VoxelMorph [36], NetGI, and NetDC all were around 0.01 s on GPU. This means that our model can perform image registration at a time comparable to other methods on the GPU platform. Although our network NetDC spent more computation resources and time on the CPU platform than other methods, the computation time was still less than 6 s, which is acceptable for registration applications. Moreover, this computation time was far less than several

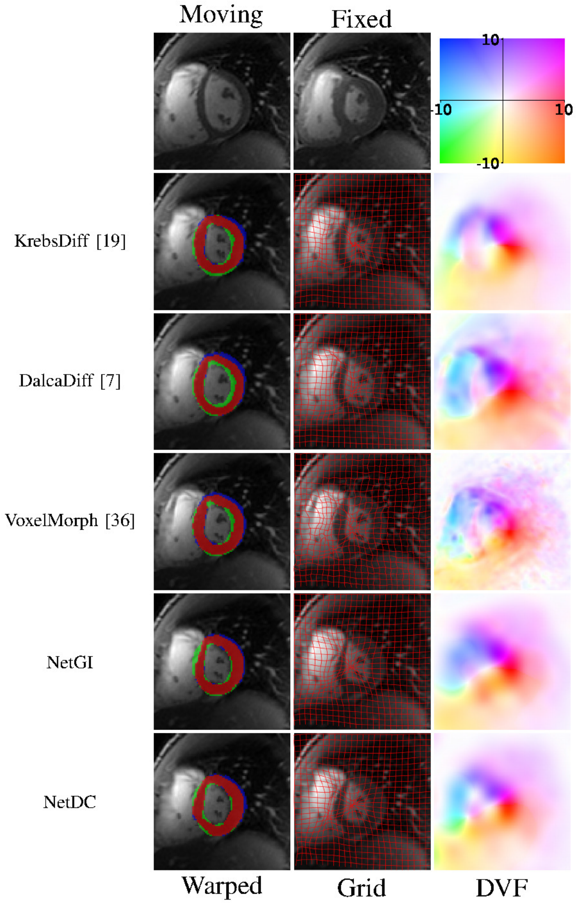


Fig. 8. Demonstration of registration results using different networks. The first row lists the moving image and the fixed image. From left to right: deformed moving images overlapped by the myocardium of the fixed image and the warped myocardium, and visualized DVFs. The myocardium of the fixed image is marked by green, and the warped myocardium of the moving image is marked by blue. The overlap of the myocardium is marked by red.

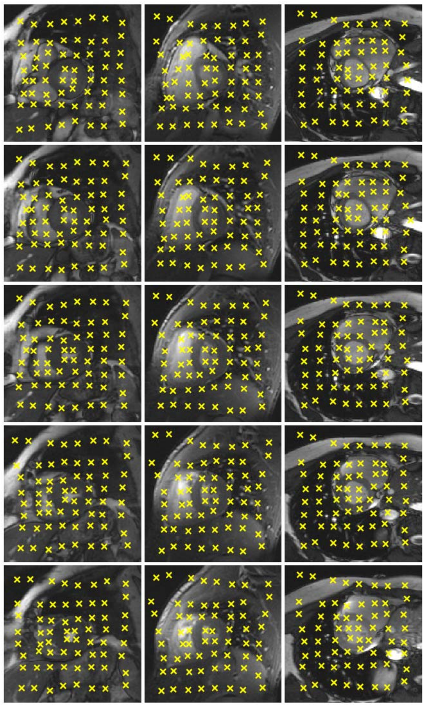


Fig. 9. Control points estimated by NetDC.

TABLE V

COMPARISON OF THE RUNTIME OF REGISTERING AN IMAGE PAIR USING KREBSDIFF, VOXELMORPH, NETGI, AND NETDC

| Method | GPU (sec.) | CPU (sec.) |
|---|---|---|
| KrebsDiff [19] | 0.008(0.007) | 0.457(0.149) |
| DalcaDiff [7] | 0.003(0.006) | 0.169(0.064) |
| VoxelMorph [36] | 0.004(0.007) | 0.378(0.152) |
| NetGI | 0.009(0.008) | 0.737(0.208) |
| NetDC | 0.012(0.008) | 5.956(1.778) |

hours spent by conventional methods. It also shows that the advantage of DL-based approaches was that the registration process is very efficient.

*5) Drifting Control Points:* Since our NetDC can adaptively shift control points, we illustrate the positions of control points after shifting in Fig. 9. NetDC estimated these drifting control points for three cases selected from MICCAI2009, York, and ACDC, respectively. Slices from the base to the apex are shown from top to bottom for each case, and yellow crosses mark the drifting control points. Note that the evenly spaced initial control points moved from their initial positions to the central area containing the heart. This implies that the control points tended to shift to the regions with large deformation, indicating that the drifting control points estimated by NetDC might be used as a feature point at some computer vision tasks.

*6) Uncertainty Estimation:* One advantage of generative models is that they can explain the uncertainty of predictions. Hub *et al.* [53] measured the uncertainty of an elastic registration method based on B-splines transformation by introducing noise to the B-splines coefficients. This uncertainty is based on the stochastic variation of control points, whereas our method explicitly provides the variance of each transformation parameter corresponding to each control point. Simpson *et al.* [54] estimated the uncertainties of the transformation parameters in the form of a covariance matrix. However, complicated closed-form derivation based on mean-field approximation is employed in their method. In our method, CNNs are employed to estimate the probabilistic parameter of transformation coefficients, making the parameters' estimation easy and direct.

In this experiment, we computed the uncertainties of the displacement vectors in the DVFs. The square root of the sum of the variances in $x$- and $y$-directions is the uncertainty of a displacement vector. To estimate the uncertainties of DVFs, random coefficients were sampled 500 times from $q_{\boldsymbol{\beta}}(z|F,M)$, and corresponding displacement vectors were obtained using (3). The uncertainties of the DVFs were computed and visualized in Fig. 10. It is observed that the displacement uncertainties of control points were more significant than that of their surrounding points. Here, we provided a simple analysis to explain the above phenomenon. For a given point $\boldsymbol{o}$, its displacement $d(\boldsymbol{o})$ along the $x$-direction is $\sum_{i=1}^{n} z_{i,x}\boldsymbol{\psi}((\|p_i-\boldsymbol{o}\|)/r)$, where $z_{i,x} \sim N(\mu_i,\Sigma_i)$. $\boldsymbol{\psi}((\|p_i-\boldsymbol{o}\|)/r)$ can be regarded as the weight of Gaussian distribution. Then, the distribution of $d(\boldsymbol{o})$ is the Gaussian $\mathcal{N}(\sum_{i=1}^{n}\boldsymbol{\psi}((\|p_i-\boldsymbol{o}\|)/r)\mu_i, \sum_{i=1}^{n}\boldsymbol{\psi}^2((\|p_i-\boldsymbol{o}\|)/r)\Sigma_i)$. Considering that $r = 2\max_{p_i\in P}$

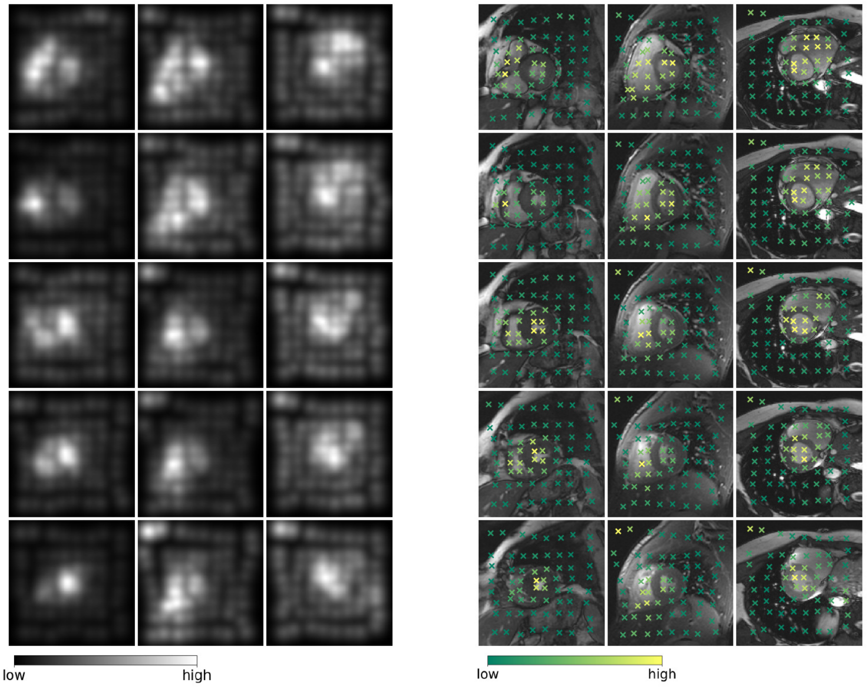


Fig. 10. Uncertainties of DVFs (left) and transformation coefficients (right).

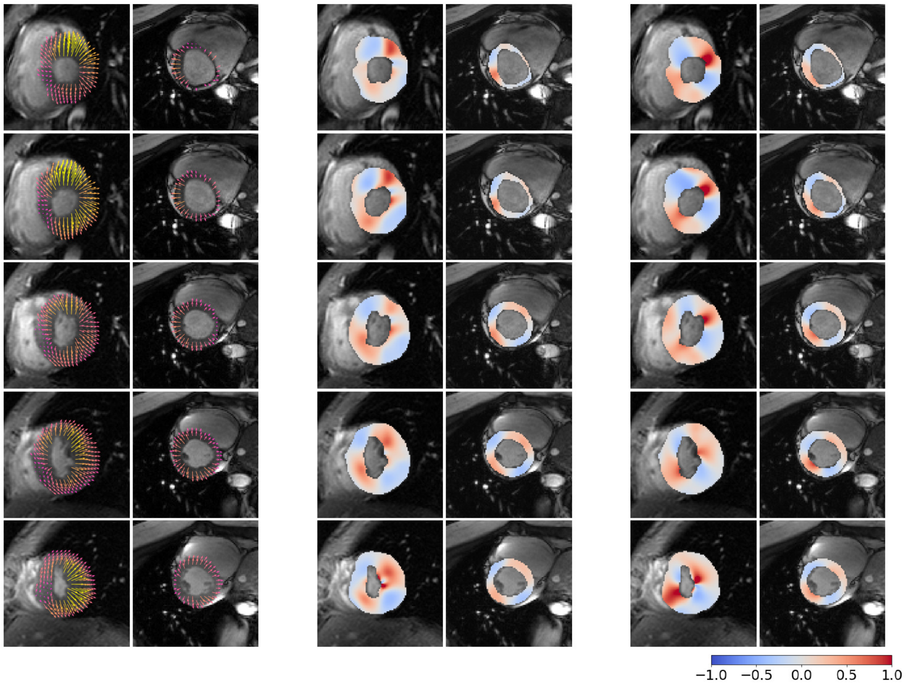


Fig. 11. Motion (left), the circumferential strain (middle), and the radial strain (right) of LV myocardium from the ES phase to the ED phase. The values of strain outside the display range are drawn in the color corresponding to the limit value.

$\min_{p_j \in P-\{p_i\}} \|p_i - p_j\|$, for the local region $\|\boldsymbol{o} - p_i\| < (r/2)$, only the control point $p_i$ is placed in this local region. When $\boldsymbol{o} = p_i$, $\boldsymbol{\psi}^2((\|p_i - \boldsymbol{o}\|)/r)$ is maximum, which implies that the uncertainty of the control point $p_i$ is larger than that of its surrounding points. On the contrary, when $\boldsymbol{o}$ locates in the middle of two adjacent control points $p_j$ and $p_k$, $\sum_{i=1}^{n} \boldsymbol{\psi}^2((\|p_i - \boldsymbol{o}\|)/r) \approx \boldsymbol{\psi}^2((\|p_j - \boldsymbol{o}\|)/r) + \boldsymbol{\psi}^2((\|p_k - \boldsymbol{o}\|)/r)$, and its value is relatively small compared with that of $p_i$ and $p_j$. Correspondingly, the uncertainties of these points are minor compared with that of adjacent control points. The same conclusion also applies to the $y$-direction.

Moreover, Fig. 10 shows that higher uncertainties mainly occurred near the region of the cardiac structure, where large deformations existed. Similarly, Yang *et al.* [30] also found that high uncertainty occurs for areas with large deformation or appearance changes. The higher uncertainty also appeared in the homogeneous areas, such as the inner area of RV and LV, consistent with the conclusion given by Simpson *et al.* [54], who measured the spatial uncertainty of each voxel in brain registration. Our uncertainty measure of networks can be used for uncertainty-based smoothing registration [55], surgical treatment planning, visualization for qualitative analyses, and pathological areas' detection [30].

*7) Motion Estimation and Strain Assessment:* For LV motion estimation, the displacements of the left myocardium are the motion fields, which can be used to calculate myocardium strain. To illustrate the LV motion, Fig. 11 shows the displacements of myocardium from ES to ED for two cases selected, the York dataset and ACDC dataset, respectively. No instances of MICCAI2009 are demonstrated because no annotations of epicardium are provided. Furthermore, the circumferential strain and the radial strain of LV myocardium [22] are also illustrated in Fig. 11.

## V. Conclusion

In this article, we design two unsupervised learning networks based on CSRBF-based transformations to estimate LV motion. Our proposed networks learn the probabilistic coefficients of control points. The network NetDC can simultaneously estimate the drifting control points and corresponding coefficients, and the NetGI can estimate transformation coefficients for given control points. In the evaluation, we also design two networks, NetGE-64 and NetGE-256, for registration with 64 and 256 evenly spaced control points. Experiments show that our networks outperform state-of-the-art networks and generate smooth DVFs. NetDC can estimate the proper locations of control points, and these control points may be used as feature points in some computer vision tasks. One potential drawback of NetDC is that the number of control points is fixed. In the future, we will work on the adaptive number of control points and explore the potential use of NetDC.

## References

[1] P. Yan, A. Sinusas, and J. S. Duncan, "Boundary element method-based regularization for recovering of LV deformation," *Med. Image Anal.*, vol. 11, no. 6, pp. 540–554, 2007.

[2] J. Wu, Z. Gan, W. Guo, X. Yang, and A. Lin, "A fully convolutional network feature descriptor: Application to left ventricle motion estimation based on graph matching in short-axis MRI," *Neurocomputing*, vol. 392, pp. 196–208, Jun. 2020.

[3] K. McLeod, M. Sermesant, P. Beerbaum, and X. Pennec, "Spatio-temporal tensor decomposition of a polyaffine motion model for a better analysis of pathological left ventricular dynamics," *IEEE Trans. Med. Imag.*, vol. 34, no. 7, pp. 1562–1575, Jul. 2015.

[4] N. S. Phatak, S. A. Maas, A. I. Veress, N. A. Pack, E. V. R. Di Bella, and J. A. Weiss, "Strain measurement in the left ventricle during systole with deformable image registration," *Med. Image Anal.*, vol. 13, no. 2, pp. 354–361, Sep. 2009.

[5] H. Gao, A. Allan, C. McComb, X. Luo, and C. Berry, "Left ventricular strain and its pattern estimated from cine CMR and validation with DENSE," *Phys. Med. Biol.*, vol. 59, no. 13, pp. 3637–3656, 2014.

[6] J. Krebs, H. Delingette, B. Mailhé, N. Ayache, and T. Mansi, "Learning a probabilistic model for diffeomorphic registration," *IEEE Trans. Med. Imag.*, vol. 38, no. 9, pp. 2165–2176, Sep. 2019.

[7] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning for fast probabilistic diffeomorphic registration," in *Medical Image Computing and Computer Assisted Intervention—MICCAI*, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, Eds. Cham, Switzerland: Springer, 2018, pp. 729–738.

[8] Z. Gan, J. Tang, and X. Yang, "Left ventricle motion estimation based on unsupervised recurrent neural network," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2019, pp. 2342–2349.

[9] K. Anjyo and J. P. Lewis, "RBF interpolation and Gaussian process regression through an RKHS formulation," *J. Math-Ind.*, vol. 3, no. 6, pp. 63–71, 2011.

[10] J. Krebs *et al.*, "Robust non-rigid registration through agent-based action learning," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 10433. Cham, Switzerland: Springer, 2017, pp. 344–352.

[11] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum, "End-to-end unsupervised deformable image registration with a convolutional neural network," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2017, pp. 204–212.

[12] B. D. de Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring, and I. Išgum, "A deep learning framework for unsupervised affine and deformable image registration," *Med. Image Anal.*, vol. 52, pp. 128–143, Feb. 2018.

[13] G. Wu, M. Kim, Q. Wang, B. C. Munsell, and D. Shen, "Scalable high-performance image registration framework by unsupervised deep feature representations learning," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 7, pp. 1505–1516, Jul. 2016.

[14] X. Cao, J. Yang, J. Zhang, D. Nie, M. Kim, Q. Wang, and D. Shen, "Deformable image registration based on similarity-steered CNN regression," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, vol. 10433, 2017, pp. 300–308.

[15] X. Cao, J. Yang, J. Zhang, Q. Wang, P.-T. Yap, and D. Shen, "Deformable image registration using a cue-aware deep regression network," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 9, pp. 1900–1911, Sep. 2018.

[16] K. A. J. Eppenhof, M. W. Lafarge, P. Moeskops, M. Veta, and J. P. W. Pluim, "Deformable image registration using convolutional neural networks," *Proc. SPIE*, vol. 10574, Mar. 2018, Art. no. 105740S.

[17] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[18] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.

[19] J. Krebs, T. Mansi, B. Mailhé, N. Ayache, and H. Delingette, "Unsupervised probabilistic deformation modeling for robust diffeomorphic registration," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2018, pp. 101–109.

[20] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.

[21] X. Yang, Z. Xue, X. Liu, and D. Xiong, "Topology preservation evaluation of compact-support radial basis functions for image registration," *Pattern Recognit. Lett.*, vol. 32, no. 8, pp. 1162–1177, Jun. 2011.

[22] S. Zhang, Y. Ma, G. Wang, and Y. Guo, "Left ventricular strain analysis from cine MRI," in *Proc. 2nd Int. Conf. Biomed. Eng. Informat.*, 2009, pp. 1–5.

[23] N. Lin and J. S. Duncan, "Generalized robust point matching using an extended free-form deformation model: Application to cardiac images," in *Proc. 2nd IEEE Int. Symp. Biomed. Imag., Macro Nano*, Apr. 2004, pp. 320–323.

[24] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE Trans. Med. Imag.*, vol. 32, no. 7, pp. 1153–1190, Jul. 2013.

[25] D. Perperidis, R. H. Mohiaddin, and D. Rueckert, "Spatio-temporal free-form registration of cardiac MR image sequences," *Med. Image Anal.*, vol. 9, no. 5, pp. 441–456, 2005.

[26] A. I. Veress, G. T. Gullberg, and J. A. Weiss, "Measurement of strain in the left ventricle during diastole with cine-MRI and deformable image registration," *J. Biomech. Eng.*, vol. 127, no. 7, pp. 1195–1207, 2005.

[27] M.-M. Rohé, M. Datar, T. Heimann, M. Sermesant, and X. Pennec, "SVF-Net: Learning deformable image registration using shape matching," in *Medical Image Computing and Computer Assisted Intervention*. Cham, Switzerland: Springer, 2017, pp. 266–274.

[28] H. Sokooti, B. de Vos, F. Berendsen, B. P. F. Lelieveldt, I. Išgum, and M. Staring, "Nonrigid image registration using multi-scale 3D convolutional neural networks," in *Medical Image Computing and Computer Assisted Intervention*. Cham, Switzerland: Springer, 2017, pp. 232–239.

[29] K. A. J. Eppenhof and J. P. W. Pluim, "Pulmonary CT registration through supervised learning with convolutional neural networks," *IEEE Trans. Med. Imag.*, vol. 38, no. 5, pp. 1097–1105, May 2019.

[30] X. Yang, R. Kwitt, M. Styner, and M. Niethammer, "Quicksilver: Fast predictive image registration—A deep learning approach," *NeuroImage*, vol. 158, pp. 378–396, Sep. 2017.

[31] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[32] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: Application to breast MR images," *IEEE Trans. Med. Imag.*, vol. 18, no. 8, pp. 712–721, Aug. 1999.

[33] H. Li and Y. Fan, "Non-rigid image registration using self-supervised fully convolutional networks without training data," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 1075–1078.

[34] J. M. Sloan, K. A. Goatman, and J. P. Siebert, "Learning rigid image registration—Utilizing convolutional neural networks for medical image registration," in *Proc. 11th Int. Joint Conf. Biomed. Eng. Syst. Technol. (BIOIMAGING)*. Lisboa, Portugal: INSTICC, 2018, pp. 89–99, doi: 10.5220/0006543700890099.

[35] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "VoxelMorph: A learning framework for deformable medical image registration," *IEEE Trans. Med. Imag.*, vol. 38, no. 8, pp. 1788–1800, Aug. 2019.

[36] G. Balakrishnan, A. Zhao, M. R. Sabuncu, A. V. Dalca, and J. Guttag, "An unsupervised learning model for deformable medical image registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9252–9260.

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[38] C. Qin *et al.*, "Joint learning of motion estimation and segmentation for cardiac MR image sequences," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018, pp. 472–480.

[39] C. Qin, B. Shi, R. Liao, T. Mansi, D. Rueckert, and A. Kamen, "Unsupervised deformable registration for multi-modal images via disentangled representations," in *Information Processing in Medical Imaging*. Cham, Switzerland: Springer, 2019, pp. 249–261.

[40] J. Fan, X. Cao, Z. Xue, P.-T. Yap, and D. Shen, "Adversarial similarity network for evaluating image alignment in deep learning based registration," in *Medical Image Computing and Computer Assisted Intervention*. Cham, Switzerland: Springer, 2018, pp. 739–746.

[41] A. Sheikhjafari, M. Noga, K. Punithakumar, and N. Ray, "Unsupervised deformable image registration with fully connected generative neural network," in *Proc. Med. Imag. With Deep Learn. (MIDL)*, Amsterdam, The Netherlands, 2018.

[42] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, "Image matching from handcrafted to deep features: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 23–79, 2021.

[43] X. Cao, J. Fan, P. Dong, S. Ahmad, P.-T. Yap, and D. Shen, "Image registration using machine and deep learning," in *Handbook of Medical Image Computing and Computer Assisted Intervention* (The Elsevier and MICCAI Society Book Series). New York, NY, USA: Academic, 2020, ch. 14, pp. 319–342.

[44] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.

[45] P. Radau, Y. Lu, K. Connelly, G. Paul, A. Dick, and G. Wright, "Evaluation framework for algorithms segmenting short axis cardiac MRI," *MIDAS J.-Cardiac MR Left Ventricle Segmentation Challenge*, vol. 49, Jul. 2009.

[46] A. Andreopoulos and J. K. Tsotsos, "Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI," *Med. Image Anal.*, vol. 12, no. 3, pp. 335–357, 2008.

[47] O. Bernard *et al.*, "Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: Is the problem solved?" *IEEE Trans. Med. Imag.*, vol. 37, no. 11, pp. 2514–2525, Nov. 2018.

[48] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. NIPS*, 2017, pp. 1–4.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[50] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[51] J. Ashburner, "A fast diffeomorphic image registration algorithm," *NeuroImage*, vol. 38, no. 1, pp. 95–113, 2007.

[52] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. ECCV*, 2012, pp. 611–625.

[53] M. Hub, M. L. Kessler, and C. P. Karger, "A stochastic approach to estimate the uncertainty involved in B-spline image registration," *IEEE Trans. Med. Imag.*, vol. 28, no. 11, pp. 1708–1716, Nov. 2009.

[54] I. J. A. Simpson, J. A. Schnabel, A. R. Groves, J. L. R. Andersson, and M. W. Woolrich, "Probabilistic inference of regularisation in non-rigid registration," *NeuroImage*, vol. 59, no. 3, pp. 2438–2451, Feb. 2012.

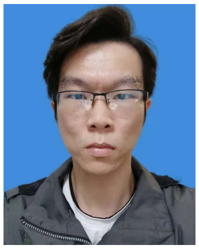
[55] I. J. A. Simpson, M. W. Woolrich, A. R. Groves, and J. A. Schnabel, "Longitudinal brain MRI analysis with uncertain registration," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*, G. Fichtinger, A. Martel, and T. Peters, Eds. Berlin, Germany: Springer, 2011, pp. 647–654.

**Ziyu Gan** received the bachelor's degree from the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China, in 2019, where he is currently pursuing the master's degree.

His research interests include medical image registration and deep learning.

**Wei Sun** received the B.S. degree from Xidian University, Xi'an, China, in 2006, the M.S. degree from Shenzhen University, Shenzhen, China, in 2011, and the Ph.D. degree from the Biomedical Imaging Group Rotterdam, Erasmus MC, Rotterdam, The Netherlands, in 2015, where he developed novel methods for stochastic image registration.

In 2016, he joined the Stevens Neuroimaging and Informatics Institute, University of Southern California, Los Angeles, CA, USA, to continue his post-doctoral research on vision-related brain connectome. In 2018, he joined the Neurology Department, Radboud University Medical Center (MC), Nijmegen, The Netherlands. His research interests include medical image analysis, deep learning, and brain connectome.



**Kaimin Liao** received the bachelor' degree from the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China, in 2019, where he is currently pursuing the master's degree.

His research interests include medical image segmentation and deep learning.

**Xuan Yang** received the Ph.D. degree in communication and information systems from Xi'an Jiaotong University, Xi'an, China, in 1998.

She has worked with the Postdoctoral Mobile Station of Electronic Science and Technology, Xi'an University, Xi'an, from 1999 to 2001. She has been working in teaching and scientific research with Shenzhen University, Shenzhen, China, since December 2001. Her current research interests include image processing, medical image processing, and deep learning.