Learning a Model-Driven Variational Network for Deformable Image Registration

Jia, Xi; Thorley, Alexander; Chen, Wei; Qiu, Huaqi; Shen, Linlin; Styles, Iain B; Chang, Hyung Jin; Leonardis, Ales; De Marvao, Antonio; O'Regan, Declan P; Rueckert, Daniel; Duan, Jinming

DOI:
10.1109/TMI.2021.3108881

Learning a Model-Driven Variational Network for Deformable Image Registration

Xi Jia, Alexander Thorley, Wei Chen, Huaqi Qiu, Linlin Shen, Iain B. Styles, Hyung Jin Chang, Ales Leonardis, Antonio de Marvao, Declan P. O'Regan, Daniel Rueckert, and Jinming Duan

***Abstract*— Data-driven deep learning approaches to image registration can be less accurate than conventional iterative approaches, especially when training data is limited. To address this issue and meanwhile retain the fast inference speed of deep learning, we propose VR-Net, a novel cascaded variational network for unsupervised deformable image registration. Using a variable splitting optimization scheme, we first convert the image registration problem, established in a generic variational framework, into two sub-problems, one with a point-wise, closed-form solution and the other one being a denoising problem. We then propose two neural layers (i.e. warping layer and intensity consistency layer) to model the analytical solution and a residual U-Net (termed generalized denoising layer) to formulate the denoising problem. Finally, we cascade the three neural layers multiple times to form our VR-Net. Extensive experiments on three (two 2D and one 3D) cardiac magnetic resonance imaging datasets show that VR-Net outperforms state-of-the-art deep learning methods on registration accuracy, whilst maintaining the fast inference speed of deep learning and the data-efficiency of variational models.**

***Index Terms*— Convolutional neural network, image registration, unsupervised learning, variational model, variational neural network.**

I. Introduction

IMAGE registration maps a floating image to a reference image according to their spatial correspondence. The procedure typically involves two operations: 1) estimating the spatial transformation between the image pair; 2) deforming the floating image with the estimated transformation. In medical image analysis, registration is critical for many automatic analysis tasks such as multi-modality fusion, population modeling, and statistical atlas learning [1], [2].

Image registration approaches can be broadly categorized into two major branches: intensity-based and landmark-based approaches. The intensity-based approaches can be either mono-modal or multi-modal. In mono-modal registration, a variational framework is often used in which the problem is framed as an optimization of the form:

$$\min_{\boldsymbol{u}} \frac{1}{2} \int_{\Omega} |I_1(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - I_0(\boldsymbol{x})|^2 \mathrm{d}\boldsymbol{x} + \lambda \mathcal{R}\left(\boldsymbol{u}(\boldsymbol{x})\right), \quad (1)$$

where I_0 and I_1: $\left(\Omega \subseteq \mathbb{R}^d\right) \rightarrow \mathbb{R}$ represent the reference image and the floating image, respectively. $\boldsymbol{u}(\boldsymbol{x}) = (u_x(\boldsymbol{x}), u_y(\boldsymbol{x}))^{\mathrm{T}} : \Omega \rightarrow \mathbb{R}^d$ denotes the deformation. In this paper, we study $d = 2$ and $d = 3$ which correspond to two-dimensional (2D) and three-dimensional (3D) cases. The first term (i.e., data term) is the sum of squared differences (SSD), which is a *similarity measure*. Minimization of the data term alone is typically an ill-posed problem with many possible solutions. Hence, the second term (i.e., regularization term) is needed, which is normally chosen to control the smoothness of the deformation.

The variational model is among the most successful and accurate approaches to calculate a deformation between two images [3]. Given a specific regularization term, such a model has a clear mathematical structure and it is also well understood which mathematical space the solution lies in, e.g., Hilbert space [4]–[6], bounded variation [7], [8], etc. However, the variational model has limitations: (1) For each image pair, the hyper-parameter λ needs to be tuned carefully to deliver a precise deformation. While too small a λ leads to an irregular, non-smooth deformation, setting it too high reduces the deformation magnitude and therefore loses the ability to model large deformations. (2) The hand-crafted regularization term itself is another hyper-parameter, which is usually selected based on assumptions about the deformation. However, existing assumptions may be too simple to capture complex changes of image content associated with biological tissues. (3) The variational model is nonlinear and therefore needs to be optimized iteratively, which is very time-consuming especially for high-dimensional data inputs.

Many deep learning approaches have been proposed for

X. Jia, A. Thorley, W. Chen, I. B. Styles, H. Chang, A. Leonardis, and J. Duan are with the School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK. I. B. Styles, A. Leonardis, and J. Duan are Fellows of the Alan Turing Institute, London NW1 2DB, UK. L. Shen is with the School of Computer Science and Software Engineering and the AI Research Center for Medical Image Analysis and Diagnosis, Shenzhen University, Shenzhen, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China. H. Qiu and D. Rueckert are with the Department of Computing, Imperial College London, London SW7 2AZ, UK. D. Rueckert is also with the Klinikum rechts der Isar, Technical University of Munich, Munich, Germany. A. de Marvao and D. P. O'Regan are with the MRC London Institute of Medical Sciences, Imperial College London, London W12 0NN, UK. The corresponding author is Jinming Duan (j.duan@cs.bham.ac.uk).

This work is supported by the British Heart Foundation Accelerator Award (AA/18/2/34218); the SmartHeart EPSRC Programme Grant (EP/P001009/1); the UK Medical Research Council (MC-A658-5QEB0); the British Heart Foundation (NH/17/1/32725, RG/19/6/34387, RE/18/4/34215); and the National Natural Science Foundation of China (91959108). This research uses the UK Biobank Resource under the application number 40119. X. Jia is partially funded by the China Scholarship Council.

unsupervised deformable medical image registration [9]–[14]. In order to learn a deformation, almost all of these learning-based approaches follow the formulation of $\boldsymbol{u} = f(I_0, I_1|\mathbf{W})$, where f is a convolutional neural network (CNN) and $\mathbf{W}$ denotes the weights of the CNN. These approaches are purely data-driven and differ from iterative variational approaches in two main aspects. (1) Data-driven approaches take images as input and directly output the estimated deformations under a loss criterion, while traditional iterative approaches take an initial deformation as input, and output a final refined deformation which is built upon the previous deformations in the iterative optimization. Whilst data-driven approaches often require substantial quantities of training data to reach an adequate level of performance, the iterative methods can work well in low data regimes. Additionally, the heavy data dependence of deep learning can result in a network that overfits the training data, and therefore lacks generalization abilities. (2) Classical iterative methods explicitly use prior and domain knowledge to construct a mathematical formulation. In contrast, data-driven methods implicitly learn prior and domain knowledge through the optimization of respective loss functions rather than explicitly building this knowledge into the network architecture itself. Some researches [15]–[18] in image reconstruction have shown that integrating such knowledge into the network enables it to learn better. Within image registration, data-driven approaches have not yet exceeded the accuracy of iterative approaches in some tasks according to [1], [19], [20], however, they have the advantage of significantly faster inference than their iterative optimization based counterparts.

In order to take advantage of both methods, in this paper, we unify a data-driven and an iterative approach into one framework and propose a model-driven variational registration network, which we term VR-Net as shown in Fig. 1. By unifying these two approaches, the VR-Net can leverage information from the entire training set to help registration in every single case, thereby having the potential to outperform iterative approaches. Note that we use the term iterative methods to denote traditional registration approaches such as TV-L_1 and FFD, which have a data term, a regularization term, and an optimization scheme to minimize the terms. We use the term data-driven to denote the recent deep learning methods that require large quantities of training data, following [20]. A model-driven approach is a learning-based approach that combines data-driven and iterative methods.

Specifically, with the help of a *variable splitting* scheme for optimization, we decompose the original iterative variational problem into two sub-problems. One has a point-wise, closed-form solution and the other can be formulated as a denoising problem. Next, we formulate the point-wise, closed-form solution with a warping layer and an intensity consistency layer. We then propose a residual U-Net for the denoising sub-problem which can be regarded as a learnable regularization term embedded in the VR-Net, replacing the hand-crafted hyper-parameter seen in iterative variational methods. Finally, within the VR-Net we cascade the warping layer, the intensity consistency layer, and the generalized denoising layer to mimic the iterative process of solving a variational model.

To evaluate the proposed VR-Net, we use two 2D publicly available cardiac MRI datasets, i.e., the UK Biobank dataset [21], Automatic Cardiac Diagnosis Challenge (ACDC) dataset [22], and one 3D cardiac MRI dataset (3D CMR) [23]. Extensive experiments on the datasets show that our VR-Net outperforms data-driven approaches with respect to registration accuracy while retains the fast inference speed of deep networks. However, we note that our VR-Net is based on an intensity constancy assumption which may not work for medical images with contrast variances or from different modalities. Collectively, our main contributions are:

- We propose VR-Net for image registration. To our knowledge, this is the first model-driven deep learning approach tailored for this task. Such a network remedies the aforementioned limitations in both iterative methods and data-driven approaches and therefore paves a new way to solve the challenging task of image registration.
- VR-Net embeds the mathematical structure from the minimization of a generic variational model into a neural network. The network mapping function f therefore inherits prior knowledge from the variational model, whilst maintaining the data efficiency of iterative methods and retaining the fast inference speed of data-driven registration methods. As such, it has the advantages from both communities and is shown to exceed state-of-the-art data-driven registration methods in terms of Dice score and Hausdorf distance on both 2D and 3D datasets.
- For iterative variational approaches to perform well for individual image pairs, one often needs to define a suitable regularization and then tune the corresponding regularization parameter λ. Instead, our VR-Net is trained with a global regularization parameter. This enables the model to effectively learn the regularization term and the values of λ from data, which removes the need to tune on individual image pairs and thus results in a more generalizable model compared with its iterative counterparts.

II. Related Works

Iterative Approaches: Image registration using iterative approaches is performed for each image pair via iterative optimization of the transformation model parameters under both image intensity and regularization constraints. Affine transformations are typically first applied to handle global transformations such as rotation, translation, shearing and scaling. This is followed by a deformable transformation which has more degrees of freedom as well as higher capability to describe local deformations. There is a wide range of classical variational methods to account for local deformations such as diffusion models [4], total variation models [7], [8], fluid models [5], [6], elastic models [24]–[26], biharmonic (linear curvature) models [27], [28], mean curvature models [29], [30], optical flow models [3], [31], [32], fractional-order variation models [33], [34], non-local graph models [35]–[37], etc. The free-form deformation (FFD) methods based on B-splines model [38], [39] are able to accurately model global and local deformations with fewer degrees of freedom parameterized by control points.

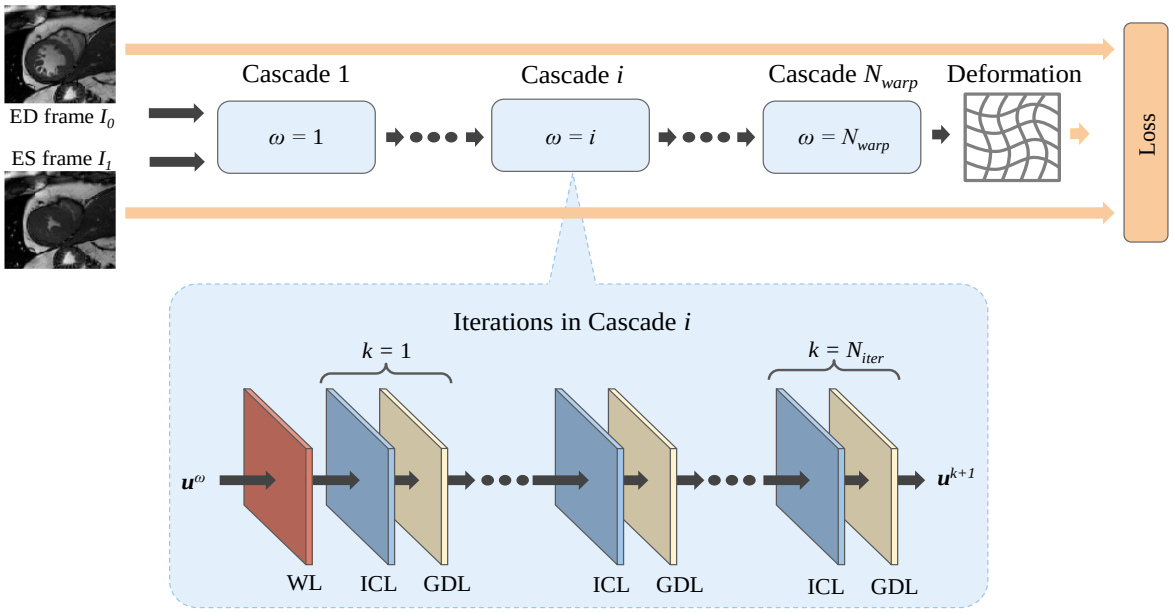


Fig. 1: VR-Net architecture. WL, ICL, and GDL denote the warping layer, intensity consistency layer, and generalized denoising layer, respectively. These layers (detailed in Fig. 2) are designed as per the minimization of a generic variational model for image registration. The cascade number is controlled by $N_{warp} \times N_{iter}$, which mimics the iterative process of minimization.

Data-Driven Approaches: Recently, researchers have started to shift their interests to unsupervised data-driven methods for medical image registration. These learning-based methods are normally trained with a large amount of paired images. By extending the spatial transformer network [40], Balakrishnan et al. [9], [41] proposed the VoxelMorph and evaluated the method on brain MRI image registration. Qin et al. [12] proposed a framework for joint registration and segmentation on cardiac MRI sequences, with their registration branch based on a Siamese-style, recurrent multi-scale network. De Vos et al. [1] proposed a multi-stage, coarse-to-fine network (termed DLIR) for parametric registration. DLIR has two types of CNNs that account for global and local transformations, respectively. The global network estimates the affine transformation and the local networks predict the displacements parameterized by the B-spline control points. The work [42] proposed by Guo et al. is also a coarse-to-fine, multi-stage registration framework. However, this method estimates only rigid transformations while our method predicts dense displacements and performs nonrigid registration. Zhao et al. [11] proposed a deep recursive cascade architecture, termed RC-Net. By cascading several base-nets, RC-Net achieved significant gains over VoxelMorph [9] on both liver and brain registration tasks. Similar to RC-Net, the proposed VR-Net also uses a cascaded, end-to-end trainable network architecture. Within each cascade of VR-Net, however, we solve a point-wise, closed-form optimization problem induced by minimizing a generic variational model, which is a major difference from other recursive [11] or multi-stage [1], [42] networks. **Model-Driven Approaches:** The authors in [16], [43], [44] studied trainable variational networks to address supervised, linear image restoration and reconstruction problems, while we tackle an unsupervised, nonlinear image registration problem. Their methods are based on proximal gradient descent, and they learn the regularization term based on the Field of Experts (FoE) [45]. The nonlinearity (derivative of the potential function in the regularization) in their method is imposed by the radial basis kernels. The optimization of their method is done through the specialized inertial incremental proximal method (IIPG), which is not implemented in a standard deep learning framework (e.g. Pytorch) and therefore may be difficult to generalize to other problems. In contrast, our network is based on a linearized variable splitting method, one advantage of which is that we can impose an exact data term (due to its closed-form solution) in each cascade which cannot be done by gradient-based methods. Our regularization is formulated as CNNs, where the nonlinearity is imposed by the activation functions (such as ReLU) and the parameters are optimized by Adam in a standard deep learning framework. There also exist works [15], [17], [18], [46], [47] that have explored variational formulations in the deep learning framework. However, instead of image registration, they were used either for image restoration and reconstruction or for video understanding. Recently, Blendowski et al. [48] proposed a supervised iterative descent algorithm (SUITS) for multi-modal image registration, which has similar ingredients to our method. SUITS uses a CNN to extract image features, which are then plugged into the Horn and Schunck (HS) model [49] to compute displacements. In other words, they need to solve an iterative model within the network each time when new displacements are required. This method can be expensive because (1) the HS model needs to have many data terms (12 in their paper) in order to align all extracted features; (2) solving the HS model itself is costly and requires iterations; and (3) they need to solve the HS model many times within the network. In contrast, we do not need to iteratively solve any optimization model within our network. Instead, we use the iterative process for optimization only to guide the design

of network architecture. Moreover, unlike [48] which uses an algebraic multigrid solver (AMG) to solve the linear system of equations, all subproblems (network layers) in our method have closed-form, point-wise solutions.

III. Generic Variational Method

In this section, we study a more general variational model [3], [49] for image registration, which is given by

$$\min_{\boldsymbol{u}} \frac{1}{s}\int_{\Omega} |I_1(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - I_0(\boldsymbol{x})|^s \mathrm{d}\boldsymbol{x} + \lambda \mathcal{R}\left(\boldsymbol{u}(\boldsymbol{x})\right), \quad (2)$$

where the variables in this formulation have the same meaning as in Eq. (1). The objective is to find the optimal deformation $\boldsymbol{u}^*(\boldsymbol{x}) : (\Omega \subseteq \mathbb{R}^d) \rightarrow \mathbb{R}^d$, that minimizes the formulation. Within the data term, $s = 1$ corresponds to L_1 estimation that is robust to outliers, while $s = 2$ gives the estimation based on the sum of squared difference. The second term is a generic regularization term, which imposes a smoothness constraint on the deformation. The hyper-parameter λ controls the smoothness of the solution. However, it is non-trivial to select both regularization term and λ optimally.

In the data term, we notice that the non-linearity in the function $I_1(\boldsymbol{x}+\boldsymbol{u})$ with respect to $\boldsymbol{u}$ poses a challenge to optimize Eq. (2). To benefit from closed-form solutions, we use the Gauss–Newton algorithm [3], [50] to handle Eq. (2). By employing the first-order Taylor expansion at $\boldsymbol{u}^\omega$, we end up with solving the following alternative problem:

$$I_1\left(\boldsymbol{x}+\boldsymbol{u}\right) = I_1\left(\boldsymbol{x}+\boldsymbol{u}^\omega\right) + \langle \nabla I_1(\boldsymbol{x}+\boldsymbol{u}^\omega), \boldsymbol{u}-\boldsymbol{u}^\omega \rangle \quad (3a)$$

$$\boldsymbol{u}^{\omega+1} = \arg\min_{\boldsymbol{u}} \frac{1}{s}\int_{\Omega} |\rho(\boldsymbol{u})|^s d\boldsymbol{x} + \lambda \mathcal{R}\left(\boldsymbol{u}\right), \quad (3b)$$

where

$$\rho(\boldsymbol{u}) = I_1\left(\boldsymbol{x}+\boldsymbol{u}^\omega\right) + \langle \nabla I_1(\boldsymbol{x}+\boldsymbol{u}^\omega), \boldsymbol{u}-\boldsymbol{u}^\omega \rangle - I_0(\boldsymbol{x}). \quad (4)$$

In Eq. (3a), ∇ is the gradient operator, ∇I_1 represents partial derivatives of I_1, $\langle \cdot, \cdot \rangle$ denotes the inner product and ω denotes the ω^{th} iteration. The linearized version of Eq. (2), seen in Eq. (3b), must to be solved iteratively. As the data term in Eq. (3b) is in a linear, convex form, one can derive a closed-form solution. Of note, to solve Eq. (2) approximately, one needs to iterate between Eq. (3a) and Eq. (3b), meaning that there exist two loops in the resulting numerical implementation.

The regularization $\mathcal{R}\left(\boldsymbol{u}\right)$ has many choices depending on what the final deformation $\boldsymbol{u}^*$ looks like, such as piecewise smooth, piecewise constant, etc. A widely used choice is the Total Variation (TV) [3], [8], [51], which is a powerful regularization that allows discontinuities in the resulting deformation. However, a major issue for those hand-crafted regularization is that they may not be optimal for more complex, task-specific applications. To circumvent these, we propose an end-to-end trainable VR-Net detailed in Section IV-A.

A. Variable Splitting

To design an appropriate VR-Net, we first adopt a variable splitting scheme [3], [17], [51] to minimize the linearized variational model Eq. (3b). Specifically, we introduce an auxiliary splitting variable $\boldsymbol{v} : (\Omega \subseteq \mathbb{R}^d) \rightarrow \mathbb{R}^d$, converting Eq. (3b) into the equivalent constrained minimization problem

$$\min_{\boldsymbol{u},\boldsymbol{v}} \frac{1}{s}\int_{\Omega} |\rho(\boldsymbol{u})|^s d\boldsymbol{x} + \lambda \mathcal{R}(\boldsymbol{v}) \quad s.t. \quad \boldsymbol{u} = \boldsymbol{v}.$$

The introduction of the constraint $\boldsymbol{u} = \boldsymbol{v}$ above decouples $\boldsymbol{u}$ in the regularization term from the data term, therefore a multi-channel denoising problem can be explicitly constructed and a closed-form, point-wise solution can be derived. Using the penalty function method, we then add the constraint back into the model and minimize the single problem

$$\min_{\boldsymbol{u},\boldsymbol{v}} \frac{1}{s}\int_{\Omega} |\rho(\boldsymbol{u})|^s d\boldsymbol{x} + \lambda \mathcal{R}(\boldsymbol{v}) + \frac{\theta}{2}\int_{\Omega} |\boldsymbol{v}-\boldsymbol{u}|^2 d\boldsymbol{x},$$

where θ is the introduced penalty weight. To solve the multi-variable minimization problem, one needs to minimize it with respect to $\boldsymbol{u}$ and $\boldsymbol{v}$ separately.

1) ***u-subproblem*** is a linear problem and handled by considering the following minimization problem

$$\boldsymbol{u}^{k+1} = \operatorname*{argmin}_{\boldsymbol{u}} \frac{1}{s}\int_{\Omega} |\rho(\boldsymbol{u})|^s d\boldsymbol{x} + \frac{\theta}{2}\int_{\Omega} |\boldsymbol{v}^k-\boldsymbol{u}|^2 d\boldsymbol{x},$$

the solution of which depends on the order of s. In the case of $s = 1$, the solution is given by the following thresholding equation

$$\boldsymbol{u}^{k+1} = \boldsymbol{v}^k - \frac{\hat{z}}{\max\left(|\hat{z}|, 1\right)}\frac{\nabla I_1}{\theta}, \quad (5)$$

where $\hat{z} = \theta\rho(\boldsymbol{v}^k)/(|\nabla I_1|^2 + \epsilon)$ and ϵ is a small positive value added to avoid division by zero to prevent vanishing gradients in the image. In Appendix 1 we develop a novel primal-dual method to derive this solution (5). Our new derivation allows the proposed method to easily adapt to vector images which usually appear in data terms that use image patch or (higher-order) gradient information [52], [53].

In the case of $s = 2$, the respective problem is differentiable and we can derive the following Sherman–Morrison formula [54], [55] by differentiating this subproblem with respect to $\boldsymbol{u}$

$$(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \theta\mathbb{1})(\boldsymbol{u}-\boldsymbol{u}^\omega) = \theta(\boldsymbol{v}^k-\boldsymbol{u}^\omega) - \mathbf{J}(I_1 - I_0), \quad (6)$$

where $\mathbf{J}\mathbf{J}^{\mathrm{T}}$ (where $\mathbf{J} = \nabla I_1$) is the rank-1 outer product and $\mathbb{1}$ is an identity matrix. Due to the identity matrix, the Sherman–Morrison formula will lead to a close-form, point-wise solution to $\boldsymbol{u}^{k+1}$. In Appendix 2, we present detailed derivations in both 2D and 3D.

2) ***v-subproblem*** is handled by considering the following minimization problem

$$\boldsymbol{v}^{k+1} = \operatorname*{argmin}_{\boldsymbol{v}} \lambda\mathcal{R}(\boldsymbol{v}) + \frac{\theta}{2}\int_{\Omega} |\boldsymbol{v}-\boldsymbol{u}^{k+1}|^2 d\boldsymbol{x}. \quad (7)$$

Given a known $\boldsymbol{u}^{k+1}$, this problem essentially is a denoising problem with the generic regularization $\mathcal{R}(\boldsymbol{v})$. Note that we assume the noise here is additive and follows a Gaussian distribution. On the other hand, if the regularization $\mathcal{R}(\boldsymbol{v})$ is TV, then it is a TV denoising problem, as in Zach's paper [3].

Putting these derivations together, we have Algorithm 1 to minimize Eq. (2) using variable splitting. Since Taylor expansion is used to linearize the non-linear function, Eq. (3a) holds only if the resulting deformation $\boldsymbol{u}^*$ is small. As such,

we adopt an extra $warping$ operation (via $\boldsymbol{u}^\omega$) in Algorithm 1, i.e., $I_1^\omega = I_1(\boldsymbol{x} + \boldsymbol{u}^\omega)$. With $warping$, we can break down a large deformation into N_{warp} small ones, each of which can be solved iteratively and optimally. The total iterations for the algorithm is $N_{warp} \times N_{iter}$.

Algorithm 1 VS for generic variational registration model

Inputs : I_0, I_1 and $(\theta, \lambda, N_{warp}, N_{iter})$.
Initialize : $\boldsymbol{u}^1$ and $\boldsymbol{v}^1$.
for $\omega = 1 : N_{warp}$ **do** ▷ # Taylor expansions
  I_1^ω = $warping(I_1, \boldsymbol{u}^\omega)$
  while $k < N_{iter}$ **do** ▷ # iterations
    update $\boldsymbol{u}^{k+1}$ via (5), (16) or (17) with $I_1 = I_1^\omega$
    $\boldsymbol{v}^{k+1} = denoiser(\boldsymbol{u}^{k+1})$
  end while
  $\boldsymbol{u}^\omega$ = $\boldsymbol{u}^{k+1}$
end for
return $\boldsymbol{u}^* = \boldsymbol{u}^\omega$ ▷ # return final solution

IV. Learning a Variational Registration Network

So far, we have shown how the variable splitting scheme can be derived to tackle the generic variational registration model. We first handle the original problem Eq. (2) with the Gauss-Newton method. For the resulting linearized minimization problem Eq. (3b) we have two sub-problems, one with a closed-form, point-wise solution for either choice of the s and one a denoising problem with $\mathcal{R}(\boldsymbol{u})$. As of yet, we have not defined the exact form of $denoiser$ in Algorithm 1. In the following section, we will detail the full VR-Net architecture, and show how a residual CNN is used as our $denoiser$ to solve the second denoising sub-problem.

A. Network Architecture

We construct the proposed VR-Net by unrolling the iterative procedure in Algorithm 1. Fig. 1 depicts the resulting network architecture. There are two types of cascade in the architecture to learn a large displacement: (1) $cascade\text{-}iter$ indicated by $k \in \{1, ..., N_{iter}\}$, stands for the inner loop in Algorithm 1; (2) $cascade\text{-}warp$ indicated by $\omega \in \{1, ..., N_{warp}\}$, corresponds to the outer loop in Algorithm 1. Note that $cascade\text{-}warp$ contains multiple nested $cascade\text{-}iter$s. In Fig. 2, we show the three computational layers contained in the network, which are the *warping layer* (WL), the *intensity consistency layer* (ICL) and the *generalized denoising layer* (GDL). They respectively correspond to Step 4, 6 and 7 in Algorithm 1.

Warping layer is achieved by using a bilinear interpolation for 2D images, following the spatial transformer networks [40]. Recall that the $warping$ operation is defined in Algorithm 1 by $I_1^\omega = I_1(\boldsymbol{x} + \boldsymbol{u}^\omega)$, where $\boldsymbol{u}^\omega$ is the estimated displacement. The bilinear interpolation is continuous and piecewise smooth, and the partial gradients with respect to $\boldsymbol{u}^\omega$ can be derived as in [40]. The 2D warping layer can be easily extended to transform 3D volumes, as in [41]. In Fig. 2, we show the computational graph of this layer, which takes $\boldsymbol{u}^\omega$ and I_1 as the inputs and outputs the warped image I_1^ω.

Intensity consistency layer is crucial as it effectively imposes intensity consistency between the warped image (I_1^w) and the target image (I_0) such that the data term in Eq. (2) can be minimized. Fig. 2 presents the computational graph of this layer. Specifically, the input I_1^w from the upstream warping layer, concurrently with I_0, $\boldsymbol{v}^k$, $\boldsymbol{u}^\omega$ and θ, are passed through Eq. (5), (16) or (17) to produce $\boldsymbol{u}^{k+1}$, which then feeds the downstream generalized denoising layer. Note that the calculations in this layer are both computationally efficient and numerically accurate thanks to the existence of point-wise, analytical solutions from Eq. (5), (16) or (17). The penalty weight θ is often manually selected in iterative methods, however in this paper we instead make it a learnable parameter.

Generalized denoising layer is a residual U-Net that explicitly defines $denoiser$ in Algorithm 1. As illustrated in Fig. 2, we intend to denoise a two-channel displacement $\boldsymbol{u}^{k+1}$ with the residual U-Net and produce its denoised version $\boldsymbol{v}^{k+1}$ for ICL in next iteration. Since the input and output of ICL and GDL are both deformations, it is natural that we can adopt a residual connection between two adjacent cascades. As the generalized denoising layer represents the denoising subproblem Eq. (7), it implicitly absorbs the hyper-parameters λ and θ and thus there is no need to tune them manually. Note that while we use a residual U-Net as the backbone here, our setup is generic and therefore allows for the incorporation of more advanced denoising CNN architectures.

The function in Eq. (5) needs special attention when implemented as a neural layer. Although it is a continuous and piecewise smooth function, it is non-differentiable. As such, the concept of sub-gradients must be used during network back-propagation. As a result, this gives us a sub-differentiable mechanism with respect to network parameters, which allows loss gradients to flow back not only to the GDL and WL but also to the ICL.

B. Network Loss and Parameterizations

Network loss: While the design of VR-Net architecture follows the philosophy of conventional optimization for iterative methods, training the network parameters is another optimization process, for which a loss function must be explicitly formulated. Due to the absence of ground truth transformations in medical imaging, we adopt an unsupervised loss function, using the floating image I_1, the reference image I_0 and the predicted deformation $\boldsymbol{u}$. The loss $\mathcal{L}(\boldsymbol{\Theta})$ is

$$\min_{\boldsymbol{\Theta}} \frac{1}{N} \sum_{i=1}^{N} \|I_1^i(\boldsymbol{x} + \boldsymbol{u}_i(\boldsymbol{\Theta})) - I_0^i(\boldsymbol{x})\|_1 + \frac{\alpha}{N} \sum_{i=1}^{N} \|\nabla \boldsymbol{u}_i(\boldsymbol{\Theta})\|_2^2, \tag{8}$$

where N is the number of training image pairs, $\boldsymbol{\Theta}$ are the network parameters to be learned and α is a hyper-parameter balancing the two losses. Note that the first loss defines the sum of absolute differences (SAD) between the warped images and the reference images and the second loss defines the smoothness on the resulting displacements. The graph representation of the two loss functions is detailed in Fig. 2.

Despite the model-driven components of our VR-Net, the method is essentially a deep learning approach so it also

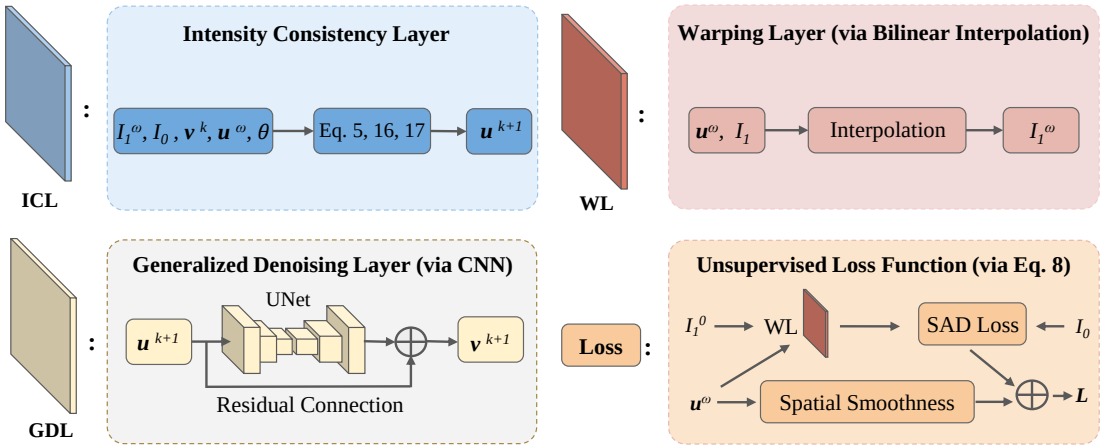


Fig. 2: Computational graph of each layer in VR-Net. ICL and GDL are designed based on solutions of sub-problems resulting from applying variable splitting to the original image registration model (2).

requires a smoothness parameter α that regularizes the learned displacements for the whole dataset. In contrast to the manual tuning of θ in Eq. (5) and (6) which is required for each test pair image in traditional iterative methods, this smoothness parameter α is only tuned in the training set and used for inference without further optimization. Note that α is not necessary if we instead use the SSD loss between predicted and ground-truth deformations as our loss function. As shown in the Fig. 1, we do not evaluate the loss function (8) at every cascade and only use it once at the very end of the VR-Net.

Parameterizations: The network learnable parameters $\boldsymbol{\Theta}$ include both the residual U-Net parameters $\mathbf{W}$ in GDL layers and the penalty weights θ in the ICLs. Recall that in VR-Net (see Fig. 1) we have $cascade\text{-}iter$ and $cascade\text{-}warp$, and therefore each GDL and ICL layer has a set of parameters $\mathbf{W}$ and θ, respectively. We experimented with two parameterization settings: $\boldsymbol{\Theta^1} = \{\mathbf{W}, \theta\}$ and $\boldsymbol{\Theta^2} = \{\{\mathbf{W}_{k,\omega}, \theta_{k,\omega}\}_{k=1}^{N_{iter}}\}_{\omega=1}^{N_{warp}}$. For $\boldsymbol{\Theta^1}$, we let the parameters $\mathbf{W}$ and θ respectively be shared by the DLs and the ICLs across $cascade\text{-}warp$ and $cascade\text{-}iter$. In $\boldsymbol{\Theta^2}$, the parameters are not shared in either $cascade\text{-}iter$ or $cascade\text{-}warp$, meaning that each layer (GDL or ICL) has its own learnable parameter. For both parameterizations we experimented with, backpropagation is employed to minimize the loss with respect to the network parameters $\boldsymbol{\Theta}$ in an end-to-end fashion.

C. Initialization

While data-driven methods take image pairs as input and directly output the estimated deformations, we need an initial displacement as input of VR-Net as stated in Step 2 of Algorithm 1. The initial displacement is then refined by the iterative process. In this paper, we proposed 3 different initialization strategies. The first strategy is to initialize the $\boldsymbol{u}^1$ and $\boldsymbol{v}^1$ with zeros, which is used in the original TV-L_1 paper [3]. The second strategy is using the Gaussian noise as initialization. However, initializing $\boldsymbol{u}^1$ and $\boldsymbol{v}^1$ with zeros or noise is not necessarily the optimal choice. Inspired by [46], we propose to learn the initialization from the data by concatenating a U-Net prior to the first WL. Note that the additional concatenated U-Net is not pre-trained. It is a part of the VR-Net and its weights are updated along with the whole VR-Net during the training process.

We evaluate the three different initialization strategies in V-D and show that the registration performance benefits from making the initialization learnable.

V. Experimental Results

In this section, we introduce the datasets and quantitative metrics used for experiments. Then we describe the implementation details of the proposed method as well as ablation studies using different configurations. Finally, we compare the proposed VR-Net with state-of-the-art methods, including both iterative methods and data-driven approaches.

A. Datasets and Quantitative Metrics

2D Datasets: We evaluate the proposed VR-Net on the UK Biobank dataset [21] and the ACDC dataset [22]. The UK Biobank [21] is a large scale cardiac MRI image dataset designed for cohort studies on 100,000 subjects. MRI scans in this dataset were acquired from healthy volunteers by using the same equipment and protocols, and the in-plane and through-plane resolutions are $1.8mm$ and $10mm$, respectively. We randomly select 220 subjects and split them into 100, 20, and 100 for training, validation, and testing, respectively. The ACDC dataset [22] was created from real clinical exams. Acquisitions were obtained over a 6 year period with two MRI scanners of different magnetic strengths. The dataset is composed of 150 patients evenly divided into 5 types of pathology. We select the 100 subjects that have ground truth segmentation masks for experiments. We split these subjects into 40, 10, and 50 for training, validation, and testing, respectively. Since the in-plane resolution varies from 1.34 to $1.68mm$, we resample all the images to $1.8mm$ before experiments. For both datasets, we perform experiments on only basal, mid-ventricular, and apical image slices.

3D Dataset: The 3D CMR dataset [23] used in our experiments consists of 220 pairs of 3D high-resolution (HR) cardiac MRI images corresponding to the end diastolic (ED)

and end systolic (ES) frames of the cardiac cycle. HR imaging requires only one single breath-hold and therefore introduces no inter-slice shift artifacts. All images are resampled to $1.2\times1.2\times1.2mm^3$ resolution and cropped or padded to matrix size $128\times128\times96$. To train comparative deep learning methods and tune hyperparameters in different methods, the dataset is split into 100/20/100 corresponding to training, validation, and test sets. We report final quantitative results on the test set only.

Due to the absence of ground truth deformations for these datasets, we evaluate the performance of different methods using the segmentation masks of left ventricle cavity (LV), left ventricle myocardium (Myo), and right ventricle cavity (RV). Specifically, we calculate the deformation between ES and ED frames and then warp the ES segmentation using the deformation. Based on the warped ES segmentation and the ground truth ED segmentation, we compute Dice score and Hausdorff distance (HD) score [56]. The Dice score varies from 0 to 1, with higher values corresponding to a better match. The HD is measured on the outer contour of each anatomical structure: LV, Myo, and RV. It is on an open-ended scale, with smaller values implying a better result.

B. Implementation Details

We implement the proposed 2D VR-Net with U-Net [57] as the backbone for all generalized denoising layers. We used the original U-Net architecture in [57] and no further optimization of the architecture is performed. As the input and output of such layers are displacements, we also apply a residual connection to the U-Net. To numerically discretize the partial derivatives ∇I_1 in Eq. (5) and Eq. (6) and $\nabla \boldsymbol{u}$ in the loss Eq. (8), the central finite difference method is adopted. To train the 2D VR-Net, the batch size is set to 10 pairs of images. α in Eq. (8) is selected using the grid-search strategy on the validation set and is set to 0.1 for UK Biobank and 0.05 for ACDC. For training, we use the basal, mid-ventricular, and apical image slices in all frames from all subjects in the training set. During inference, we evaluate the 2D VR-Net and other comparative approaches using the three slices at the ED and ES phases from all subjects in the test set. This is because we only have manual segmentation masks at the two phases. Extending the 2D VR-Net to 3D is straightforward. The major difference between the 2D and 3D VR-Net is the generalized denoising layer, for 3D, we adopt a lighter 5-level hierarchical U-shape network from [58] as the backbone. The training batch size of 3D VR-Net is set to 2. α in Eq. (8) is selected to be 0.0001 using the validation set for the 3D CMR dataset.

Both 2D and 3D VR-Nets are implemented with Pytorch [59] and trained using a GeForce 1080 Ti GPU with 11GB RAM. An Adam optimizer [60] with two beta values of 0.9 and 0.999 is used and the initial learning rate is set 0.0001. Note that we train our VR-Net using each dataset separately. For UK Biobank, the maximum iterations are 50,000 and the learning rate is gradually reduced after 25,000 iterations. For ACDC, the maximum iterations are 20,000 and the learning rate is gradually reduced after 10,000 iterations. For the 3D CMR dataset, the maximum iterations are 30,000 and the learning rate is fixed during training. Due to the limitation of GPU memory, the maximum cascade number we could afford is 6 and 2 for 2D and 3D VR-Net, respectively. VR-Net is memory intensive as it has multiple cascaded GDL, however, it is very efficient in terms of speed during inference, as listed in Section V-E. Note the memory dependencies can be reduced by using lighter CNN architectures in GDL.

C. Ablation Studies

In this section, we test different configurations for VR-Net. Specifically, we explore the impact of using different data terms, denoising networks, parameterizations and varying numbers of cascades. For simplicity, we use shorthand notations to represent different configurations. For example, R-L_1-3×2 indicates that we use the U-Net with residual connection, Eq. (5) (L_1 data term), $N_{warp}=3$ and $N_{iter}=2$ in VR-Net. U-L_2-6×1 indicates that we use the U-Net without residual connection, Eq. (6) (L_2 data term), $N_{warp}=6$ and $N_{iter}=1$.

We first compare the results obtained by using different cascades in VR-Net. From Table I, we observe that the best results almost all come from using 6 cascades (either 3×2 or 6×1), indicating that increasing cascade number improves the performance. On the UK Biobank, the best result is achieved by R-L_2-6×1 (0.804 Dice and 10.26 HD), while on ACDC the best result is achieved by R-L_1-3×2 (0.873 Dice and 6.33 HD). When comparing the best performance among different data terms, L_1 performs worse than L_2 on UK Biobank, while on ACDC L_1 is better. This suggests that the proposed VR-Net is robust to different data terms. Next, we compare the results obtained by using different denoising networks, and we notice a tiny improvement when a residual connection is applied.

In Fig. 3a we show the performance of VR-Net on UK Biobank with two different parameterizations: $\boldsymbol{\Theta}^1$ and $\boldsymbol{\Theta}^2$. From these boxplots, we see that using $\boldsymbol{\Theta}^2$ performs better on RV and Myo anatomical structures, while on RV using $\boldsymbol{\Theta}^1$ is better. The averaged results (last two columns) on the three anatomical regions indicate a similar performance between the two parameterizations. Note that the number of network parameters in $\boldsymbol{\Theta}^1$ is 1/6 of that in $\boldsymbol{\Theta}^2$.

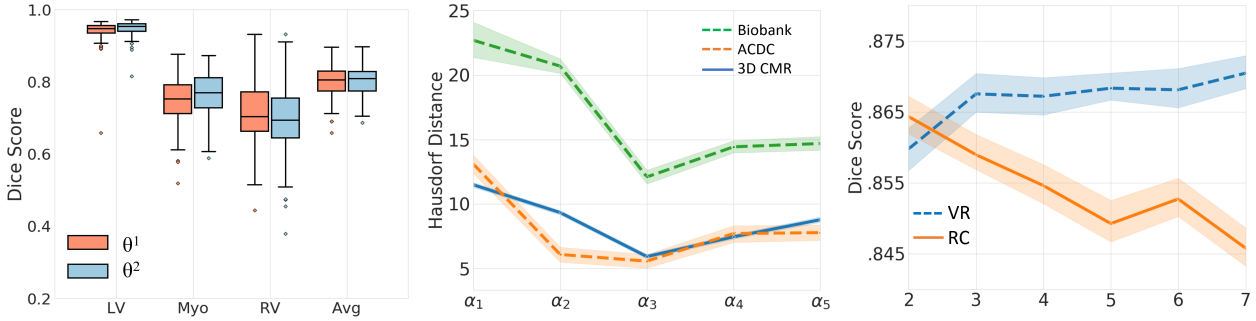
While the original regularization weight λ is absorbed in the $\boldsymbol{v}$-subproblem to avoid manual choice, by using the training loss in Eq. (8) we do introduce another parameter α. However, tuning α is based on the whole dataset and we tune it only during training. We presented a curve plot that illustrates how different α affect the registration accuracy (as shown in Fig. 3b). Specifically, we used five different values of α to train the proposed VR-Net five times on three datasets, i.e., $\alpha_{UKBB}=\{1, 0.5, 0.1, 0.05, 0\}$, $\alpha_{ACDC}=\{0.5, 0.1, 0.05, 0.005, 0\}$, and $\alpha_{3DCMR}=\{0.01, 0.001, 0.0001, 0.00001, 0\}$. We then plot their registration accuracy (in terms of Hausdorff Distance) on each dataset as α varies. As suggested by the curve plot, the optimal values of α for UK Biobank, ACDC, and 3D CMR datasets are 0.1, 0.05, and 0.0001, respectively.

D. Initialization Strategies

In Table II, we explore the performance of VR-Net using different initialization approaches on the UK Biobank dataset.

TABLE I: Comparison of image registration performance on two datasets using different configurations for the proposed VR-Net. Dice (HD) score is computed by averaging that of LV, Myo and RV at the basal, mid-ventricular and apical image slices from all subjects in the test set. Mean and standard deviation (in parenthesis) are reported.

Methods	UK Biobank		ACDC		Methods	UK Biobank		ACDC	
	Dice	HD	Dice	HD		Dice	HD	Dice	HD
R-L_2-1×1	.785(.047)	10.69(3.05)	.860(.058)	6.74(2.40)	U-L_2-1×1	.784(.046)	10.65(3.03)	.860(.062)	6.72(2.53)
R-L_2-2×1	.795(.046)	10.69(3.15)	.850(.063)	6.67(2.19)	U-L_2-2×1	.791(.045)	10.52(3.06)	.861(.058)	6.57(2.43)
R-L_2-2×2	.798(.046)	10.49(3.12)	.867(.054)	6.60(2.38)	U-L_2-2×2	.793(.044)	10.48(3.09)	**.866(.056)**	6.58(2.53)
R-L_2-3×2	.799(.044)	10.63(3.18)	**.872(.052)**	**6.44(2.38)**	U-L_2-3×2	.802(.043)	**10.32(3.09)**	.866(.060)	**6.52(2.43)**
R-L_2-6×1	**.804(.043)**	**10.26(3.07)**	.869(.054)	6.65(2.50)	U-L_2-6×1	**.803(.043)**	10.47(3.14)	.856(.062)	6.76(2.52)
R-L_1-1×1	.779(.048)	10.77(3.03)	.853(.061)	6.75(2.53)	U-L_1-1×1	.781(.047)	10.77(3.07)	.854(.063)	6.88(2.58)
R-L_1-2×1	.789(.047)	10.62(3.13)	.865(.058)	6.51(2.42)	U-L_1-2×1	.783(.048)	10.74(3.04)	.858(.062)	6.77(2.54)
R-L_1-2×2	.794(.045)	10.58(3.00)	.865(.060)	6.48(2.38)	U-L_1-2×2	.793(.046)	10.56(3.01)	.867(.058)	6.55(2.48)
R-L_1-3×2	.796(.046)	10.54(3.16)	**.873(.050)**	**6.33(2.13)**	U-L_1-3×2	**.797(.045)**	**10.53(3.14)**	**.872(.052)**	**6.39(2.50)**
R-L_1-6×1	**.800(.045)**	**10.35(3.10)**	.850(.063)	6.67(2.19)	U-L_1-6×1	.790(.098)	10.61(3.10)	.845(.068)	7.03(2.62)



(a) Two parameterizations (b) Impact of using different α (c) Cascade Numbers

Fig. 3: (a): Dice scores of R-L_2-6×1 using the two parameterizations in Sec. IV-B on the UK Biobank. (b) Impact of using different α in terms of Hausdorff distance on the three datasets. (c): Comparing VR-Net and RC-Net [11] using a different number of cascades on the ACDC dataset.

As is evident in this table, with zeros or noises as the initial displacements, the Dice results of VR-Net dropped by 6.0% and 6.4%, respectively, and the HD results dropped by $1.59mm$ and $1.42mm$, respectively. These results suggest that making the initialization learnable is crucial as (1) registration is nonconvex and its solution depends on initialization, and (2) our network builds on iterative optimization methods and thus also relies on initialization. Furthermore, our VR-Net is derived using the Taylor linearization and as such computes only a small displacement in each iteration. When we initialize the input displacement with noise or zeros, 6 iterations are not sufficient to perform a good registration.

TABLE II: Performance of VR-Net on Biobank using different initialization. Note the U-Net is not pretrained, it is also a learnable layer in the whole VR-Net.

Initialization	Dice	HD
U-Net	.804(.043)	10.26(3.07)
Noise	.740(.048)	11.68(3.05)
Zeros	.744(.051)	11.85(3.18)

E. Comparison with State-of-the-Art

In this section, we compare our VR-Net with iterative methods (i.e. FFD [38] and TV-L_1 [3]) and data-driven deep learning methods (i.e. VoxelMorph [9], [41], Siamese network [12], [19] and RC-Net [11]) on the UK Biobank, ACDC and 3D CMR dataset. An overview of the Dice and HD scores of different methods can be found in the boxplots in Fig. 5.

2D Methods: For FFD, we use the implementation in MIRTK [38], where we chose the SSD similarity with bending energy regularisation. We use a 3-level multi-resolution scheme and set the spacing of B-spline control points on the highest resolution to $8mm$. For TV-L_1, which uses the L_1 data term and the total variation regularization, we implement its ADMM solver, in which we use a similar three-level multi-scale strategy for the minimization. We implement TV-L_1 using the same variable splitting and therefore its overall iterative structure is very similar to our VR-Net. However, because TV-L_1 is cheap to iterate, we can set sufficient numbers of inner iterations (associated with variable splitting) and outer iterations (associated with Taylor expansions) to compute the final deformation. In other words, we tune TV-L_1 to its maximum capability to compete with our method. The regularization weights in the two methods are tuned to maximize the accuracy performance on validation sets. For the data-driven methods, we first compare our VR-Net with VoxelMorph [41] which we re-implement for 2D registration. We also compare VR-Net with the Siamese network regularized by the approximated Huber loss [12], [19]. Lastly, for the recursive cascade network (RC-Net) [11], which used a 3D U-Net-like architecture in a cascade fashion, we re-implement a 2D version. Overall, the backbone of both VoxelMorph and RC-Net is a U-Net and the loss functions (without segmentation loss) are similar to ours. Note that all the compared data-driven methods (including Siamese, VoxelMorph, and RC-

TABLE III: Comparison of image registration performance using different methods on UK Biobank. 'Avg' means that Dice (HD) is computed by averaging that of LV, Myo and RV of all subjects in the test set. Here mean and standard deviation (in parenthesis) are reported. 'Unreg' stands for unregistered and #×RC-Net the number of cascades used in RC-Net.

Methods	Dice				HD				$J_{<0}\%$	$\|\nabla J\|$
	LV	Myo	RV	Avg	LV	Myo	RV	Avg		
Unreg	.634(.072)	.344(.086)	.551(.080)	.510(.055)	11.99(1.64)	10.08(2.91)	24.52(6.24)	15.53(2.40)	–	–
FFD	.934(.025)	.711(.081)	.672(.110)	.772(.051)	4.87(2.15)	7.86(4.03)	21.18(7.69)	11.30(3.26)	0.23(0.29)	.019(.011)
TV-L_1	.937(.036)	.717(.076)	.701(.105)	.785(.047)	4.75(1.67)	7.12(3.16)	19.73(7.21)	10.53(2.86)	0.65(0.30)	.051(.017)
Siamese	.932(.022)	.706(.069)	.695(.099)	.778(.046)	4.75(1.65)	6.52(3.23)	20.69(7.02)	10.65(3.01)	0.42(0.21)	.065(.016)
VoxelMorph	.931(.029)	.717(.072)	.685(.102)	.778(.047)	4.57(1.47)	6.71(3.43)	21.78(7.27)	10.69(3.06)	0.07(0.10)	.027(.008)
2×RC-Net	.942(.022)	.737(.066)	.703(.099)	.794(.044)	4.43(1.57)	6.65(3.35)	20.29(7.15)	10.46(3.01)	0.22(0.17)	.041(.010)
3×RC-Net	.944(.036)	.736(.068)	**.705(.105)**	.795(.048)	4.28(1.81)	7.39(3.32)	**19.96(6.53)**	10.55(2.80)	0.70(0.36)	.069(.019)
4×RC-Net	.945(.022)	.736(.065)	.701(.120)	.794(.045)	4.36(1.54)	7.23(3.39)	20.54(7.32)	10.71(2.98)	0.49(0.24)	.056(.015)
5×RC-Net	.944(.033)	.723(.065)	.703(.109)	.790(.045)	4.24(1.60)	8.09(3.51)	20.16(6.83)	10.83(2.77)	1.18(0.53)	.094(.023)
6×RC-Net	.941(.024)	.714(.068)	.696(.114)	.784(.048)	4.60(1.53)	7.66(3.23)	20.51(7.17)	10.92(2.98)	1.29(0.55)	.096(.023)
7×RC-Net	.943(.025)	.721(.066)	.695(.113)	.786(.047)	4.52(1.56)	7.66(3.20)	20.42(7.06)	10.87(2.89)	1.00(0.44)	.084(.022)
R-L_2-6×1	**.948(.021)**	**.764(.060)**	.700(.105)	**.804(.043)**	**3.90(1.41)**	**6.49(3.79)**	20.38(7.21)	**10.26(3.07)**	0.38(0.18)	.039(.012)

TABLE IV: Comparison of image registration performance using different methods on the ACDC dataset.

Methods	Dice				HD				$J_{<0}\%$	$\|\nabla J\|$
	LV	Myo	RV	Avg	LV	Myo	RV	Avg		
Unreg	.666(.178)	.540(.143)	.672(.145)	.626(.108)	12.21(4.34)	7.65(2.67)	12.74(4.56)	10.87(3.13)	–	–
FFD	.920(.063)	.792(.067)	.803(.126)	.838(.059)	5.16(2.14)	5.87(2.18)	9.60(4.56)	6.88(2.40)	0.32(0.42)	.031(.037)
TV-L_1	.902(.106)	.793(.086)	.835(.117)	.843(.075)	5.97(3.25)	6.11(2.85)	9.51(4.49)	7.20(2.81)	0.52(0.37)	.053(.020)
Siamese	.872(.106)	.723(.113)	.778(.132)	.791(.081)	7.34(3.49)	6.05(1.84)	10.55(4.30)	7.98(2.68)	0.15(0.17)	.052(.011)
VoxelMorph	.924(.066)	.789(.096)	.837(.104)	.850(.062)	5.51(2.81)	5.83(2.26)	9.33(4.02)	6.89(2.50)	0.38(0.35)	.066(.018)
2×RC-Net	.931(.053)	.798(.083)	.864(.082)	.864(.050)	5.46(2.49)	6.22(2.56)	8.63(3.97)	6.77(2.46)	0.54(0.37)	.097(.023)
3×RC-Net	.931(.048)	.794(.076)	.852(.095)	.859(.049)	6.08(2.61)	7.02(2.71)	9.15(3.95)	7.42(2.40)	1.02(0.57)	.131(.029)
4×RC-Net	.926(.056)	.789(.075)	.849(.086)	.855(.050)	6.01(2.78)	6.62(2.70)	9.32(3.76)	7.32(2.53)	0.95(0.55)	.123(.028)
5×RC-Net	.919(.072)	.780(.075)	.849(.091)	.849(.055)	6.38(2.80)	7.29(2.96)	9.37(3.78)	7.68(2.40)	1.24(0.68)	.140(.031)
6×RC-Net	.926(.050)	.779(.088)	.853(.090)	.853(.051)	6.67(2.73)	8.02(3.69)	9.19(3.89)	7.96(2.63)	1.94(0.91)	.180(.040)
7×RC-Net	.927(.048)	.769(.085)	.842(.104)	.846(.053)	6.72(2.57)	8.44(3.35)	9.44(3.86)	8.20(2.50)	2.17(1.00)	.194(.042)
R-L_1-3×2	**.934(.052)**	**.815(.078)**	**.869(.082)**	**.873(.050)**	**5.09(2.20)**	**5.48(2.19)**	**8.43(3.72)**	**6.33(2.13)**	0.32(0.25)	.078(.024)

Net) are only trained with the training data and no test-time (instance) optimization is adopted. The hyper-parameters of all data-driven methods are tuned individually according to the validation set for a fair comparison.

3D Methods: We again use a three-level pyramid scheme with SSD similarity and bending energy regularisation for FFD, tuning control point spacing on the validation set. Next, we compare our VR-Net with the diffeomorphic Demons [55] implemented in SimpleITK [61]. For Demons, we use a three-level pyramid scheme, and optimize the number of iterations and smoothing parameter on the validation set. Finally, we compare with the official ANTs SyN implementation [62] with SSD similarity and a four-level pyramid scheme. Hyper-parameters in ANTs SyN such as similarity, number of pyramid levels, and number of iterations in each level are tuned on the whole validation set.

In Table III and IV, we show the quantitative results obtained by using different methods on UK Biobank and ACDC. In the tables, one can see that VR-Net outperforms iterative methods and data-driven methods on both datasets for almost all anatomical structures. On UK Biobank, RC-Net achieves the best results on RV in terms of both Dice and HD, which are 0.005 and 0.42mm higher than those obtained by our best configuration (R-L_2-6×1). However, in terms of Dice, VR-Net achieves 0.948 on LV and 0.764 on Myo, outperforming 3×RC-Net by 0.004 and by 0.028, respectively. In terms of HD for LV and Myo, our VR-Net improves 3×RC-Net from 4.28mm to 3.90mm and from 7.39mm to 6.49mm, respectively. On average, the proposed VR-Net achieves a better Dice and HD score than 3×RC-Net, making our VR-Net the best method on this dataset.

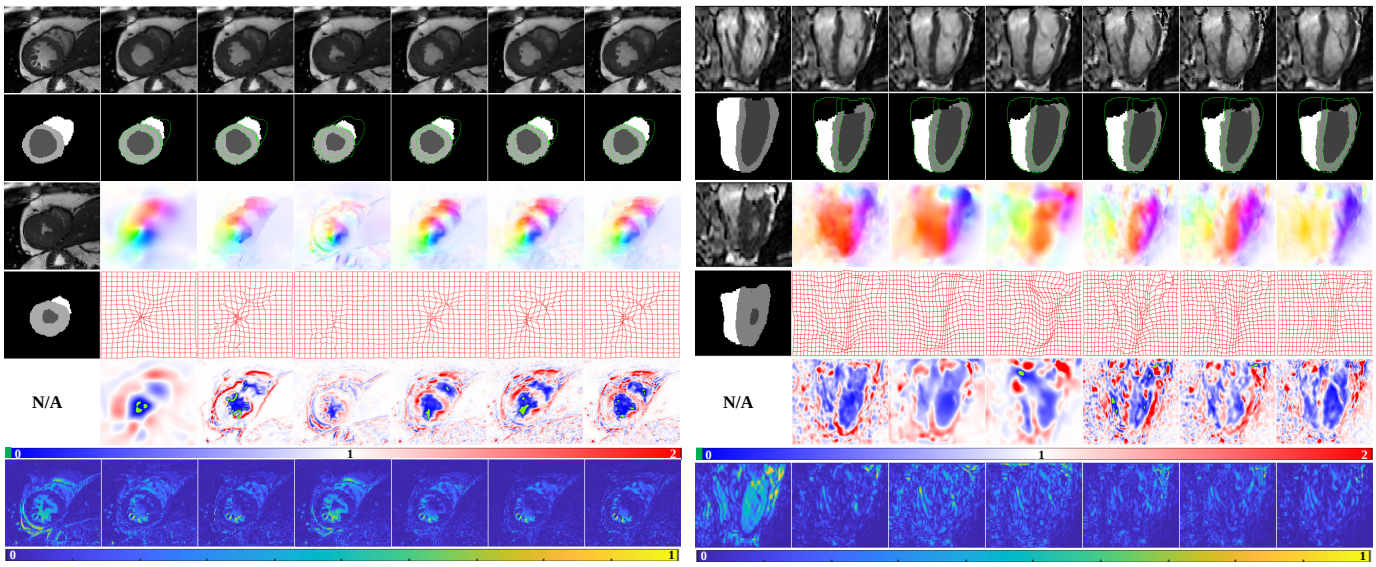
On ACDC, the proposed VR-Net with the configuration of R-L_1-3×2 outperforms all other methods across all anatomical structures. While 2×RC-Net also obtains comparable results, one can notice that its performance drops rapidly with more cascades. To visualize this, we plotted the average Dice scores of both RC-Net and VR-Net versus the number of cascades in Fig. 3c on this dataset. As is evident from this figure, there is a sharp decrease in the performance of RC-Net, which is due to RC-Net overfits the small training set of 40 subjects. In contrast, VR-Net performs constantly well using an increasing number of cascades, demonstrating its data-efficiency. This is attributable to the integration of the iterative variational model (prior knowledge) into the VR-Net.

On the 3D CMR dataset, as listed in Table V, FFD outperforms all compared methods on the Myo and RV, and achieves the highest average Dice score, i.e. 0.739. Although the average Dice of our VR-Net (U-L_1-2×1) is lower than that of FFD with 0.11 margin, the average HD score is higher than that of FFD. Furthermore, the proposed VR-Net achieves both the highest Dice and HD score among the compared data-driven methods.

We also listed the percentage of negative Jacobian determinant values as well as the gradient magnitude of the Jacobian determinant of all compared methods on both the UK Biobank and ACDC datasets. From Table III and IV, we can see that although VR-Net generates foldings in deformation, it produces fewer than RC-Net with the same number of cascades, i.e. 0.38% of R-L_2-6×1 and 1.00% of 7×RC-Net on the UK Biobank, and 0.32% of R-L_1-3×2 and 2.17% of

TABLE V: Comparison of image registration performance using different methods on the 3D CMR dataset.

Methods	Dice				HD				$J_{<0}\%$	$\|\nabla J\|$
	LV	Myo	RV	Avg	LV	Myo	RV	Avg		
Unreg	.516(.039)	.384(.084)	.579(.044)	.493(.043)	9.99(1.15)	7.22(.10)	8.00(1.11)	8.40(.89)	–	–
Demons	.812(.049)	.710(.052)	.659(.047)	.727(.040)	5.64(1.28)	6.46(1.28)	8.12(1.21)	6.74(1.00)	0.00(0.00)	.063(.008)
FFD	.808(.055)	**.726(.059)**	**.684(.061)**	**.739(.047)**	5.89(1.52)	7.26(1.78)	**7.98(1.38)**	7.04(1.20)	0.77(0.30)	.034(.008)
SyN	.803(.060)	.710(.061)	.649(.064)	.721(.051)	5.70(1.62)	7.26(1.84)	**7.98(1.38)**	6.98(1.23)	0.01(0.01)	.027(.004)
VoxelMorph	.817(.028)	.676(.051)	.634(.046)	.709(.032)	5.94(1.35)	7.22(1.26)	9.13(1.31)	7.43(1.05)	0.75(0.22)	.089(.014)
2×RC-Net	.820(.030)	.701(.051)	.657(.047)	.726(.034)	5.60(1.25)	6.50(1.22)	8.17(1.21)	6.76(0.99)	0.11(0.05)	.057(.008)
3×RC-Net	.824(.027)	.692(.050)	.647(.048)	.721(.033)	5.67(1.10)	6.87(1.24)	8.61(1.23)	7.05(0.96)	0.49(0.13)	.085(.013)
R-L_2-2×1	**.825(.026)**	.695(.050)	.649(.047)	.723(.032)	5.56(1.30)	6.80(1.23)	8.61(1.25)	6.99(1.04)	1.07(0.31)	.110(.018)
R-L_1-2×1	.821(.026)	.706(.046)	.650(.046)	.726(.031)	5.42(1.25)	6.57(1.26)	8.29(1.26)	6.76(1.03)	0.84(0.25)	.120(.017)
U-L_2-2×1	.822(.027)	.674(.051)	.645(.046)	.714(.033)	**5.37(1.16)**	6.76(1.07)	8.52(1.29)	6.89(0.98)	0.58(0.25)	.069(.011)
U-L_1-2×1	.821(.028)	.706(.045)	.657(.047)	.728(.031)	5.38(1.28)	**6.23(1.14)**	8.03(1.21)	**6.55(1.01)**	0.24(0.12)	.081(.011)



(a) 2D ACDC (b) 3D CMR

Fig. 4: Comparing visual results obtained by different registration methods on the ACDC and 3D CMR datasets. The 1st column includes ED image, ED mask, ES image, ES mask, and absolute difference between ES image and the ED image. Excluding the 1st column, for (a) ACDC, from left to right: FFD, TV-L_1, Siamese Net, VoxelMorph, RC-Net, and VR-Net results, respectively, for (b) 3D CMR, from left to right: Demons, ANTs SyN, FFD, VoxelMorph, RC-Net, and VR-Net results, respectively. From top to bottom: warped ES images, warped ES masks (with ground truth mask shown in green contours), estimated deformations (shown in HSV and grid), the Jacobian map, and absolute differences between warped ES images and the ground truth ED image, respectively.

7×RC-Net on ACDC. On the 3D CMR dataset, as shown in Table V, VR-Net (0.24%) again outperforms the 3×RC-Net (0.49%) as well as VoxelMorph (0.75%), however, it is lower than the 2×RC-Net (0.11%). Overall, VR-Net cannot guarantee zero foldings in estimated deformations, it produces deformations comparable with VoxelMorph and RC-Net.

In Table VI, we list the runtime of different methods. Although we adopt the mathematical structure of a variational model, our VR-Net is very close to the purely data-driven deep learning methods as the solutions are point-wise closed-form, and it is much faster than traditional iterative methods. The runtime is measured and averaged over 100 test subjects.

Lastly, in Fig. 4, we compare the visual results of different methods by showing two image registration examples from the ACDC and 3D CMR datasets. On the ACDC, as can be seen, FFD (2nd column), which used L_2 regularization, over-smooths the displacement, the warped ES images are also over-smoothed around the Myo/LV area resulting in the high absolute differences. In contrast, TV-L_1 (3rd column), which used L_1 regularization, preserves edges in the resulting displacements. However, the shape of RV warped by TV-L_1 is not very smooth. This side effect also can be seen in the Siamese network result, and the Siamese network also produces a very high difference map. The displacement results of VoxelMorph, RC-Net, and VR-Net are smooth and look more natural. But the absolute difference map of VoxelMorph shows the less accurate registration than VR-Net. Additionally, the Jacobian map of RC-Net has more foldings than VR-Net(highlighted in green). The warped Myo of VoxelMorph from ACDC has unsmooth shape. The unsmooth shape can also be found in the warped masks of RC-Net. In terms of similarity, the result of VR-Net is the closest one to the ground truth, visually illustrating that the method is more accurate for image warping. On the 3D CMR, Demons and ANTs SyN do not have any negative Jacobians (i.e. no green area in Jacobian maps), due to diffeomorphsims. However, SyN produces a

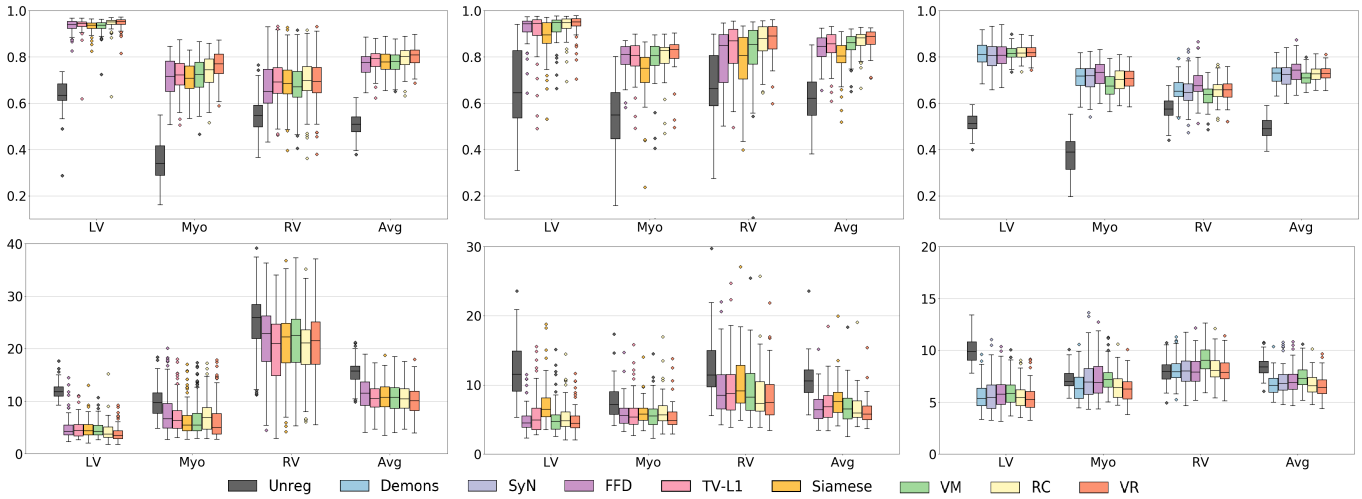


Fig. 5: Boxplot illustration of Dice (top row) and HD (bottom row) results obtained by different registration methods on the UK Biobank (left), ACDC (middle), and the 3D CMR (right) datasets. The proposed VR-Net outperforms all compared methods on the UK Biobank and ACDC datasets. Although the Dice of VR-Net is lower than that of FFD on the 3D CMR dataset, it achieves the best HD score.

very high absolute difference map. Though the warped ES mask of FFD has a very good overlapping with the ground truth ED mask, its displacement has many foldings (shown in the red grid). The foldings of displacements can also be seen in the VoxelMorph, RC-Net, and VR-Net, however, the warped ES image of VR-Net is closer to the ground truth ED image. The warped ES images of both VoxelMorph and RC-Net have distorted regions on the upper right, resulting in high difference maps on this area.

TABLE VI: Runtimes of different methods. The runtimes are measured and averaged over 100 test subjects.

Methods	2D		3D	
	CPU	GPU	CPU	GPU
TV-L_1	10.01	–	–	–
Demons	–	–	13.01	–
SyN	–	–	77.39	–
FFD	5.15	–	141.38	–
Siamese	0.07	0.01	–	–
VoxelMorph	0.07	0.01	5.97	0.10
2×RC-Net	0.13	0.01	11.95	0.21
3×RC-Net	0.19	0.02	17.85	0.34
7×RC-Net	0.44	0.03	–	–
R,U-$L_{1,2}$-1×1	0.13	0.01	11.95	0.22
R,U-$L_{1,2}$-2×1	0.19	0.02	18.25	0.33
R,U-$L_{1,2}$-6×1	0.42	0.03	–	–

F. Discussion

1) Relationship with VoxelMorph and RC-Net: In the proposed VR-Net, we use an additional U-Net to learn initial displacements. We emphasize here that this U-Net is not pre-trained and instead it is part of the VR-Net, which is trained end-to-end. In this case, without any DL, WL, or ICL layers this initial U-Net alone is essentially VoxelMorph, the performance of which is inferior to our VR-Net by a large margin as shown in Tables III, IV and V. If we recursively use the U-Net for multiple times without using other subsequent layers (such as ICL/GDL), then the model is equivalent to the RC-Net, the performance of which is worse than our VR-Net as shown in Table III, IV and V.

2) Generalized Denoising Layers: To understand how the GDL layer is functioning within the network, in Fig. 6 we illustrate the output of this layer after each cascade of the VR-Net using two different setups. Specifically, we use R-L_2-6×1 using both the U-Net and random noise initialization from Table II. For the U-Net initialization, we add a Gaussian noise to the input displacement to demonstrate whether this layer can produce any smoothing effect. As shown in the top two rows of Fig. 6, the deformation becomes gradually smooth as cascades proceed. The deformation also gets increasingly smooth for the random noise initialization. This visualization suggests that our GDL has the denoising effect. However, the capability of GDL is beyond denoising alone. As shown in the last two rows in Fig. 6, this layer can turn a pure random noise into deformation, indicating its capability of inducing smoothness whilst going beyond denoising and contributing to the deformation itself.

3) Identifying the Optimal Structure: In Table I, we list 24 configurations of VR-Net, along with proposed two parameterizations of $\boldsymbol{\Theta^1}$ and $\boldsymbol{\Theta^2}$. Empirically searching for the best structure can be computationally expensive. What is the strategy to efficiently determine the combination? We observe that the best results almost all come from VR-Net with 6 cascades (maximum we can afford) in 2D datasets, indicating that increasing the cascade number improves performance. We therefore suggest using more cascades if one can afford them. As for the two parameterizations (Θ^1 and Θ^2), we notice a slight improvement using Θ^2 and therefore use this parameterization for all our comparative experiments. Comparing different data terms, we find L_1 and L_2 are on par with each other, which may be due to their closed-form solutions. We therefore use both L_1 and L_2 data terms for comparative experiments. Next, by comparing the results obtained by using

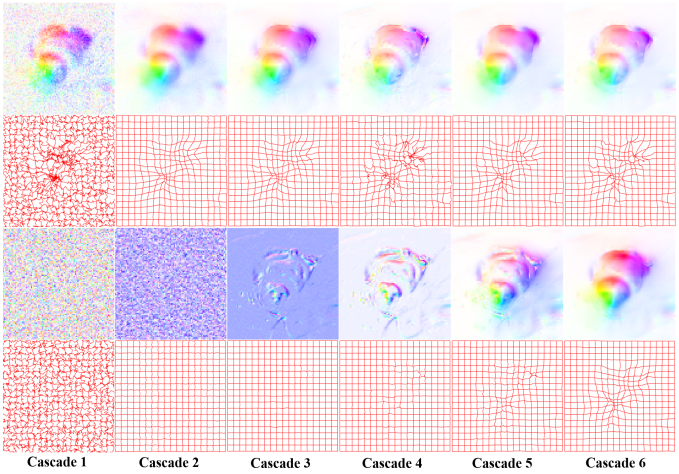


Fig. 6: Visualizing deformation in each cascade (after GDL) using noise corrupted deformation (top) and random noise (bottom) as initialization. Top two rows show a noise corrupted deformation is denoised by GDL as cascades proceed. Bottom two rows show if we input random noise, VR-Net is still capable of producing a smooth deformation.

different denoising networks in Table I, we find residual U-Net performs better and therefore use it for the comparative experiments on UK Biobank and ACDC. However, U-Net is better on the 3D CMR dataset, as shown in Table V.

4) Brightness constancy assumption: The brightness constancy assumption in Eq. (2) is often not suited for medical images with contrast variances and therefore our method will not work well for those images. However, we would like to point out that the proposed framework is not limited to only this assumption and can be extended to other similarity/dissimilarity metrics such as local cross correlation (invariant to multiplicative illumination changes), mutual information (suitable for multi-modality image registration) and others. The idea is to use the second-order Taylor theorem [35], [52] to expand a respective metric and then approximate the Hessian matrix in the Taylor expansion with a positive semi-definite matrix. In this case, the resultant problem is a convex optimization which fits in our proposed framework. On the other hand, it is also possible to consider other L_1 or L_2 based data terms, including contrast invariant descriptors based on image gradients [52], [53] or modality independent image descriptors such as nonlocal MIND [63]. We will investigate these in our future research.

VI. Conclusion

In this paper, we propose a model-driven VR-Net for deformable image registration, which combines the iterative variational method with modern data-driven deep learning methods. By taking advantage of both approaches, our VR-Net outperforms deep data-driven methods as well as classical iterative methods (in terms of Hausdorff distance) on three cardiac MRI datasets. Extensive experimental results show our VR-Net is fast, accurate, and data-efficient. For our future work, we will extend the VR-Net to multi-modality image registration.

VII. Appendix 1

In this section, we propose to derive the solution of $\boldsymbol{u}-$subproblem ($s = 1$) in Section III-A using a primal-dual method, originally proposed in [64] for Total Variation denoising [65]. Here we use all notations in 3D only. First, we rewrite the subproblem into its discrete form

$$\min_{\boldsymbol{u}} \|\rho(\boldsymbol{u})\|^1 + \frac{\theta}{2}\|\boldsymbol{v}^k - \boldsymbol{u}\|^2, \tag{9}$$

where $\rho(\boldsymbol{u}) = \langle \nabla I_1, \boldsymbol{u} - \boldsymbol{u}^\omega \rangle + I_1 - I_0$. This minimization problem (9) can be converted equivalently to a saddle-point problem by writing the first term as a maximization, i.e.

$$\|\rho(\boldsymbol{u})\|^1 = \max_{\|z\|_\infty \leq 1} \langle \rho(\boldsymbol{u}), z \rangle,$$

over the dual variable $z \in \mathbb{R}^{MNH}$ where MNH is the the image size, and $\|z\|_\infty = \max_{i,j,l} |z_{i,j,l}|$ and $\langle \rho(\boldsymbol{u}), z \rangle = \sum_{i,j,l} (\rho(\boldsymbol{u}))_{i,j,l} z_{i,j,l}$ where i, j, l denote image indices.

The minimization problem (9) is equivalent to the following primal-dual (min-max) problem, i.e.

$$\min_{\boldsymbol{u}} \max_{z, \|z\|_\infty \leq 1} \langle \rho(\boldsymbol{u}), z \rangle + \frac{\theta}{2}\|\boldsymbol{v}^k - \boldsymbol{u}\|^2, \tag{10}$$

over the primal variable $\boldsymbol{u} \in \mathbb{R}^{MNH}$ and the dual variable z, respectively.

First, we differentiate (10) with respect to $\boldsymbol{u}$ and derive its first-order optimality condition, resulting in the following closed-form solution for $\boldsymbol{u}$

$$\boldsymbol{u} = \boldsymbol{v}^k - z\frac{\nabla I_1}{\theta}. \tag{11}$$

We then plug the solution (11) into (10), converting the primal-dual problem into the following dual problem only

$$\max_{z, \|z\|_\infty \leq 1} \langle \rho(\boldsymbol{v}^k - z\frac{\nabla I_1}{\theta}), z \rangle + \frac{1}{2\theta}\|z\nabla I_1\|_2^2. \tag{12}$$

If we differentiate (12) with respect to z and derive its first-order optimality condition, we have the following formulation

$$\hat{z} = \frac{\theta\rho(\boldsymbol{v}^k)}{|\nabla I_1|^2},$$

which needs to be projected to the convex set $Z = \left\{ z \in \mathbb{R}^{MNH} : \|z\|_\infty \leq 1 \right\}$ to satisfy the constraint $\|z\|_\infty \leq 1$. This results in

$$z = \frac{\hat{z}_{i,j,l}}{\max\left(|\hat{z}_{i,j,l}|, 1\right)}. \tag{13}$$

Note that, although the KKT condition is not considered when we handle the inequality constraint $\|z\|_\infty \leq 1$, the derivation of z above still makes sense as it is equivalent to a one-step proximal gradient descent with the optimal step size.

Finally, we plug (13) into (11) which leads to the solution for $\boldsymbol{u}$ without involving the dual variable z

$$\boldsymbol{u} = \boldsymbol{v}^k - \frac{\hat{z}_{i,j,l}}{\max\left(|\hat{z}_{i,j,l}|, 1\right)}\frac{\nabla I_1}{\theta}, \tag{14}$$

which is a point-wise, closed-form solution, the same as (5) of the $\boldsymbol{u}-$subproblem in Section III-A. We highlight that our derivation presented here can be easily applied to vector images, which usually appear in data terms that use image patch or gradient information.

VIII. Appendix 2

In this section, we derive the solution of the Sherman Morrison formula (6) in 2D and 3D. For both cases, we need to invert the left-hand side matrix in Eq. (6). As per [54], we have

$$(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \theta\mathbb{1})^{-1} = \theta^{-1}\mathbb{1} - \frac{\mathbf{J}\mathbf{J}^{\mathrm{T}}}{\theta^2 + \theta\mathbf{J}^{\mathrm{T}}\mathbf{J}}.$$

In 2D, this matrix is a 2×2 symmetric matrix for which each entry is of the 2D image size (MN). In 3D, it becomes a 3×3 symmetric matrix for which each entry is of the 3D image size (MNH). The solution $\boldsymbol{u}^{k+1}$ is therefore given by

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^{\omega} + \left[\mathbb{1} - \frac{\mathbf{J}\mathbf{J}^{\mathrm{T}}}{\theta + \mathbf{J}^{\mathrm{T}}\mathbf{J}}\right]\left[\boldsymbol{v}^k - \boldsymbol{u}^{\omega} - \theta^{-1}\mathbf{J}(I_1 - I_0)\right], \tag{15}$$

which is the form in terms of matrix and vector multiplication. With Eq. (15), it is now trivial to derive the final point-wise, closed-form solutions in both 2D and 3D.

First, in 2D where $I_1 \in \mathbb{R}^{MN}$, we have

$$\mathbf{J}\mathbf{J}^{\mathrm{T}} = \begin{bmatrix} I_1^x I_1^x & I_1^x I_1^y \\ I_1^y I_1^x & I_1^y I_1^y \end{bmatrix} \in \left(\mathbb{R}^{MN}\right)^4$$

and $\mathbf{J}^{\mathrm{T}}\mathbf{J} = |\nabla I_1|^2 = I_1^x I_1^x + I_1^y I_1^y$, where $I_1^x \in \mathbb{R}^{MN}$ and $I_1^y \in \mathbb{R}^{MN}$ are respectively the horizontal and vertical derivatives of the source image I_1. With $\boldsymbol{u} = (u_1, u_2)^{\mathrm{T}} \in \left(\mathbb{R}^{MN}\right)^2$ and $\boldsymbol{v} = (v_1, v_2)^{\mathrm{T}} \in \left(\mathbb{R}^{MN}\right)^2$, we can rewrite Eq. (15) into the following forms in terms of both components of $\boldsymbol{u}$

$$\begin{cases} u_x^{k+1} = u_x^{\omega} + \dfrac{\begin{array}{c}(I_1^y I_1^y + \theta)(v_x^k - u_x^{\omega}) - I_1^x I_1^y (v_y^k - u_y^{\omega}) \\ -I_1^x (I_1 - I_0)\end{array}}{I_1^x I_1^x + I_1^y I_1^y + \theta} \\ u_y^{k+1} = u_y^{\omega} + \dfrac{\begin{array}{c}(I_1^x I_1^x + \theta)(v_y^k - u_y^{\omega}) - I_1^y I_1^x (v_x^k - u_x^{\omega}) \\ -I_1^y (I_1 - I_0)\end{array}}{I_1^x I_1^x + I_1^y I_1^y + \theta} \end{cases}. \tag{16}$$

Then, in 3D where $I_1 \in \mathbb{R}^{MNH}$, we have

$$\mathbf{J}\mathbf{J}^{\mathrm{T}} = \begin{bmatrix} I_1^x I_1^x & I_1^x I_1^y & I_1^x I_1^z \\ I_1^y I_1^x & I_1^y I_1^y & I_1^y I_1^z \\ I_1^z I_1^x & I_1^z I_1^y & I_1^z I_1^z \end{bmatrix} \in \left(\mathbb{R}^{MNH}\right)^9$$

and $\mathbf{J}^{\mathrm{T}}\mathbf{J} = |\nabla I_1|^2 = I_1^x I_1^x + I_1^y I_1^y + I_1^z I_1^z$, where $I_1^x \in \mathbb{R}^{MNH}$, $I_1^y \in \mathbb{R}^{MNH}$ and $I_1^z \in \mathbb{R}^{MNH}$ are the derivatives of I_1 along x, y and z directions, respectively. With $\boldsymbol{u} = (u_1, u_2, u_3)^{\mathrm{T}} \in \left(\mathbb{R}^{MNH}\right)^3$ and $\boldsymbol{v} = (v_1, v_2, v_3)^{\mathrm{T}} \in \left(\mathbb{R}^{MNH}\right)^3$, we can rewrite Eq. (15) into the following forms in terms of each component of $\boldsymbol{u}$:

$$\begin{cases} u_x^{k+1} = u_x^{\omega} + \dfrac{\begin{array}{c}(I_1^y I_1^y + I_1^z I_1^z + \theta)(v_x^k - u_x^{\omega}) - I_1^x I_1^y (v_y^k - u_y^{\omega}) \\ -I_1^x I_1^z (v_z^k - u_z^{\omega}) - I_1^x (I_1 - I_0)\end{array}}{I_1^x I_1^x + I_1^y I_1^y + I_1^z I_1^z + \theta} \\ u_y^{k+1} = u_y^{\omega} + \dfrac{\begin{array}{c}(I_1^x I_1^x + I_1^z I_1^z + \theta)(v_y^k - u_y^{\omega}) - I_1^y I_1^x (v_x^k - u_x^{\omega}) \\ -I_1^y I_1^z (v_z^k - u_z^{\omega}) - I_1^y (I_1 - I_0)\end{array}}{I_1^x I_1^x + I_1^y I_1^y + I_1^z I_1^z + \theta} \\ u_z^{k+1} = u_z^{\omega} + \dfrac{\begin{array}{c}(I_1^x I_1^x + I_1^y I_1^y + \theta)(v_y^k - u_y^{\omega}) - I_1^z I_1^x (v_x^k - u_x^{\omega}) \\ -I_1^z I_1^y (v_z^k - u_z^{\omega}) - I_1^z (I_1 - I_0)\end{array}}{I_1^x I_1^x + I_1^y I_1^y + I_1^z I_1^z + \theta} \end{cases}. \tag{17}$$

We note that both 2D and 3D solutions, i.e., Eqs. (16) and (17), are closed-form and point-wise and therefore can be computed very efficiently.

References

[1] 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, 2019.

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

[3] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L_1 optical flow," in *Joint Pattern Recognit. Symp.* Springer, 2007, pp. 214–223.

[4] B. Fischer and J. Modersitzki, "Fast diffusion registration," *Contemp Math*, vol. 313, pp. 117–128, 2002.

[5] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," *Int J Comput Vision*, vol. 61, no. 2, pp. 139–157, 2005.

[6] C. Chen, B. Gris, and O. Oktem, "A new variational model for joint image reconstruction and motion estimation in spatiotemporal imaging," *SIAM J. Imaging Sci.*, vol. 12, no. 4, pp. 1686–1719, 2019.

[7] C. Frohn-Schauf, S. Henn, and K. Witsch, "Multigrid based total variation image registration," *Comput. Vis. Sci.*, vol. 11, no. 2, pp. 101–113, 2008.

[8] V. Vishnevskiy, T. Gass, G. Szekely, C. Tanner, and O. Goksel, "Isotropic total variation regularization of displacements in parametric image registration," *IEEE Trans. Med. Imag.*, vol. 36, no. 2, pp. 385–395, 2016.

[9] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 9252–9260.

[10] J. Zhang, "Inverse-consistent deep networks for unsupervised deformable image registration," *arXiv preprint arXiv:1809.03443*, 2018.

[11] S. Zhao, Y. Dong, E. I.-C. Chang, and Y. Xu, "Recursive cascaded networks for unsupervised medical image registration," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[12] C. Qin, W. Bai, J. Schlemper, S. E. Petersen, S. K. Piechnik, S. Neubauer, and D. Rueckert, "Joint learning of motion estimation and segmentation for cardiac MR image sequences," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 472–480.

[13] A. Hering, B. van Ginneken, and S. Heldmann, "mlvirnet: Multilevel variational image registration network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 257–265.

[14] 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, 2019.

[15] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, "A deep cascade of convolutional neural networks for dynamic MR image reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 2, pp. 491–503, 2017.

[16] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, "Learning a variational network for reconstruction of accelerated MRI data," *Magn Reson Med*, vol. 79, no. 6, pp. 3055–3071, 2018.

[17] J. Duan, J. Schlemper, C. Qin, C. Ouyang, W. Bai, C. Biffi, G. Bello, B. Statton, D. P. O'Regan, and D. Rueckert, "VS-Net: Variable splitting network for accelerated parallel MRI reconstruction," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 713–722.

[18] H. K. Aggarwal, M. P. Mani, and M. Jacob, "Modl: Model-based deep learning architecture for inverse problems," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 394–405, 2018.

[19] H. Qiu, C. Qin, L. Le Folgoc, B. Hou, J. Schlemper, and D. Rueckert, "Deep learning for cardiac motion estimation: supervised vs. unsupervised training," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2019, pp. 186–194.

[20] D. Rueckert and J. A. Schnabel, "Model-based and data-driven strategies in medical image computing," *Proc. IEEE*, vol. 108, no. 1, pp. 110–124, Jan 2020.

[21] S. E. Petersen, P. M. Matthews, F. Bamberg, D. A. Bluemke, J. M. Francis, M. G. Friedrich, P. Leeson, E. Nagel, S. Plein, F. E. Rademakers *et al.*, "Imaging in population science: cardiovascular magnetic resonance in 100,000 participants of UK Biobank-rationale, challenges and approaches," *J Cardiov Magn Reson*, vol. 15, no. 1, p. 46, 2013.

[22] O. Bernard, A. Lalande, C. Zotti, F. Cervenansky, X. Yang, P.-A. Heng, I. Cetin, K. Lekadir, O. Camara, M. A. G. Ballester *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, 2018.

[23] J. Duan, G. Bello, J. Schlemper, W. Bai, T. J. Dawes, C. Biffi, A. de Marvao, G. Doumoud, D. P. O'Regan, and D. Rueckert, "Automatic 3D bi-ventricular segmentation of cardiac images by a shape-refined multi-task deep learning approach," *IEEE Trans. Med. Imag.*, vol. 38, no. 9, pp. 2151–2164, 2019.

[24] C. Broit, *Optimal registration of deformed images*. University of Pennsylvania, 1981.

[25] T. Lin, C. Le Guyader, I. Dinov, P. Thompson, A. Toga, and L. Vese, "Gene expression data to mouse atlas registration using a nonlinear elasticity smoother and landmark points constraints," *J. Sci. Comput.*, vol. 50, no. 3, pp. 586–609, 2012.

[26] L. A. Vese and C. Le Guyader, *Variational methods in image processing*. CRC Press Boca Raton, FL, 2016.

[27] B. Fischer and J. Modersitzki, "Curvature based image registration," *J Math Imaging Vis*, vol. 18, no. 1, pp. 81–85, 2003.

[28] J. Modersitzki, *Numerical methods for image registration*. Oxford University Press on Demand, 2004.

[29] N. Chumchob, K. Chen, and C. Brito-Loeza, "A fourth-order variational image registration model and its fast multigrid algorithm," *Multiscale Modeling & Simulation*, vol. 9, no. 1, pp. 89–128, 2011.

[30] N. Chumchob and K. Chen, "Improved variational image registration model and a fast algorithm for its numerical approximation," *Numer Meth Part D E*, vol. 28, no. 6, pp. 1966–1995, 2012.

[31] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision (ECCV)*. Springer, 2004, pp. 25–36.

[32] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for TV-L_1 optical flow," in *Statistical and Geometrical Approaches to Visual Motion Analysis*. Springer, 2009, pp. 23–45.

[33] J. Zhang and K. Chen, "Variational image registration by a total fractional-order variation model," *J. Comput. Phys.*, vol. 293, pp. 442–461, 2015.

[34] C. Xu, Y. Wen, and B. He, "A novel fractional order derivate based log-demons with driving force for high accurate image registration," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 1997–2001.

[35] M. Werlberger, T. Pock, and H. Bischof, "Motion estimation with non-local total variation regularization," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2464–2471.

[36] R. Ranftl, K. Bredies, and T. Pock, "Non-local total generalized variation for optical flow estimation," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 439–454.

[37] B. W. Papież, A. Szmul, V. Grau, J. M. Brady, and J. A. Schnabel, "Non-local graph-based regularization for deformable image registration," in *Medical Computer Vision and Bayesian and Graphical Models for Biomedical Imaging*. Springer, 2016, pp. 199–207.

[38] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. 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, 1999.

[39] W. Lu, M.-L. Chen, G. H. Olivera, K. J. Ruchala, and T. R. Mackie, "Fast free-form deformable registration via calculus of variations," *Physics in Medicine & Biology*, vol. 49, no. 14, p. 3067, 2004.

[40] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[41] 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.

[42] H. Guo, M. Kruger, S. Xu, B. J. Wood, and P. Yan, "Deep adaptive registration of multi-modal prostate images," *Comput Med Imag Grap*, vol. 84, p. 101769, 2020.

[43] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, 2017.

[44] E. Kobler, T. Klatzer, K. Hammernik, and T. Pock, "Variational networks: Connecting variational methods and deep learning," in *Pattern Recognition*, ser. Lecture Notes in Computer Science. Springer, 2017, pp. 281–293.

[45] S. Roth and M. Black, "Fields of Experts: a framework for learning image priors," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, 2005, pp. 860–867.

[46] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang, "End-to-end learning of motion representation for video understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6016–6025.

[47] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3929–3938.

[48] M. Blendowski, L. Hansen, and M. P. Heinrich, "Weakly-supervised learning of multi-modal features for regularised iterative descent in 3d image registration," *Med Image Anal*, vol. 67, p. 101822, 2021.

[49] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[50] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Int J Comput Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[51] W. Lu, J. Duan, Z. Qiu, Z. Pan, R. W. Liu, and L. Bai, "Implementation of high-order variational models made easy for image processing," *Math. Methods Appl. Sci.*, vol. 39, no. 14, pp. 4208–4233, 2016.

[52] C. Vogel, S. Roth, and K. Schindler, "An evaluation of data costs for optical flow," in *German Conference on Pattern Recognition*. Springer, 2013, pp. 343–353.

[53] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, "Highly accurate optic flow computation with theoretically justified warping," *Int J Comput Vision*, vol. 67, no. 2, pp. 141–158, 2006.

[54] M. S. Bartlett, "An inverse matrix adjustment arising in discriminant analysis," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 107–111, 1951.

[55] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Diffeomorphic Demons: Efficient non-parametric image registration," *NeuroImage*, vol. 45, no. 1, pp. S61–S72, 2009.

[56] W. Bai, M. Sinclair, G. Tarroni, O. Oktay, M. Rajchl, G. Vaillant, A. M. Lee, N. Aung, E. Lukaschuk, M. M. Sanghvi *et al.*, "Automated cardiovascular magnetic resonance image analysis with fully convolutional networks," *J Cardiov Magn Reson*, vol. 20, no. 1, p. 65, 2018.

[57] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[58] T. C. Mok and A. C. Chung, "Fast symmetric diffeomorphic image registration with convolutional neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

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

[61] B. C. Lowekamp, D. T. Chen, L. Ibáñez, and D. Blezek, "The design of SimpleITK," *Frontiers in neuroinformatics*, vol. 7, p. 45, 2013.

[62] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, "A reproducible evaluation of ants similarity metric performance in brain image registration," *Neuroimage*, vol. 54, no. 3, pp. 2033–2044, 2011.

[63] M. P. Heinrich, M. Jenkinson, M. Bhushan, T. Matin, F. V. Gleeson, M. Brady, and J. A. Schnabel, "Mind: Modality independent neighbourhood descriptor for multi-modal deformable registration," *Med Image Anal*, vol. 16, no. 7, pp. 1423–1435, 2012.

[64] A. Chambolle, "An algorithm for total variation minimization and applications," *J Math Imaging Vis*, vol. 20, no. 1, pp. 89–97, 2004.

[65] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D: Nonlinear Phenom.*, vol. 60, no. 1-4, pp. 259–268, 1992.