Networks for Joint Affine and Non-parametric Image Registration

Zhengyang Shen
UNC Chapel Hill
zyshen@cs.unc.edu

Xu Han
UNC Chapel Hill
xhs400@cs.unc.edu

Zhenlin Xu
UNC Chapel Hill
zhenlinx@cs.unc.edu

Marc Niethammer
UNC Chapel Hill
mn@cs.unc.edu

Abstract

We introduce an end-to-end deep-learning framework for 3D medical image registration. In contrast to existing approaches, our framework combines two registration methods: an affine registration and a vector momentum-parameterized stationary velocity field (vSVF) model. Specifically, it consists of three stages. In the first stage, a multi-step affine network predicts affine transform parameters. In the second stage, we use a Unet-like network to generate a momentum, from which a velocity field can be computed via smoothing. Finally, in the third stage, we employ a self-iterable map-based vSVF component to provide a non-parametric refinement based on the current estimate of the transformation map. Once the model is trained, a registration is completed in one forward pass. To evaluate the performance, we conducted longitudinal and cross-subject experiments on 3D magnetic resonance images (MRI) of the knee of the Osteoarthritis Initiative (OAI) dataset. Results show that our framework achieves comparable performance to state-of-the-art medical image registration approaches, but it is much faster, with a better control of transformation regularity including the ability to produce approximately symmetric transformations, and combining affine and non-parametric registration.

1. Introduction

Registration is a fundamental task in medical image analysis to establish spatial correspondences between different images. To allow, for example, localized spatial analyses of cartilage changes over time or across subject populations, images are first registered to a common anatomical space.

Traditional image registration algorithms, such as elastic [3, 25], fluid [5, 12, 29, 8, 31] or B-spline models [24], are based on the iterative numerical solution of an optimization problem. The objective of the optimization is to minimize image mismatch and transformation irregularity. The sought-for solution is then a spatial transformation which aligns a source image well to a target image while assuring that the transformation is sufficiently regular. To this end, a variety of different similarity measures to assess image mismatch have been proposed. For image pairs with a similar intensity distribution, Mean Square Error (MSE) on intensity differences is widely used. For multi-modal registration, however, Normalized Cross Correlation (NCC) and Mutual Information (MI) usually perform better. Besides, smooth transformation maps are typically desirable. Methods encouraging or enforcing smoothness use, for example, rigidity penalties [26] or penalties that encourage volume preservation [27, 22] to avoid folds in the transformation. Diffeomorphic transformations can also be achieved by optimizing over sufficiently smooth velocity fields from which the spatial transformation can be recovered via integration. Such methods include Large Displacement Diffeomorphic Metric Mapping (LDDMM) [5, 12] and Diffeomorphic Demons [29]. As optimizations are typically over very high-dimensional parameter spaces, they are computationally expensive.

Recently, taking advantage of deep learning, research has focused on replacing costly numerical optimization with a learned deep regression model. These methods are extremely fast as only the evaluation of the regression model is required at test time. They imitate the behavior of conventional, numerical optimization-based registration algorithms as they predict the same types of registration parameters: displacement fields, velocity fields or momentum fields. Depending on the predicted parameters, theoretical properties of the original registration model can be retained. For example, In Quicksilver [33], a network is learned to predict the initial momentum of LDDMM, which can then be used to find a diffeomorphic spatial transformation via LDDMM's shooting equations. While earlier work has focused on training models based on previously obtained registration parameters via costly numerical optimization [6, 32], recent work has shifted to end-to-end formulations[1] [10, 14, 4, 9]. These end-to-end approaches integrate image resampling into their network and were inspired

[1]For these end-to-end approaches, the sought-for registration parameterization is either the final output of the network (for the prediction of displacement fields) or an intermediate output (for the prediction of velocity fields) from which the transformation map can be recovered. The rest of the formulation stays the same.

by the spatial-transformer work of Jaderberg et al. [13]. Non end-to-end approaches require the sought-for registration parameters at training time. To obtain such data via numerical optimization for large numbers of image pairs can be computationally expensive, whereas end-to-end approaches effectively combine the training of the network with the implicit optimization over the registration parameters (as part of the network architecture).

Existing deep learning approaches to image registration exhibit multiple limitations. First, they assume that images have already been pre-aligned, *e.g.*, by rigid or affine registration. These pre-alignment steps can either be done via a specifically trained network [7] or via standard numerical optimization. In the former case the overall registration approach is no longer end-to-end, while in the latter the pre-registration becomes the computational bottleneck. Second, many approaches are limited by computational memory and hence either only work in 2D or resort to small patches in 3D. Though some work explores end-to-end formulations for entire 3D volumes [4, 9], these approaches perform computations based on the full resolution transformation map, in which case a very simple network can easily exhaust the memory and thus limit extensions of the model. Third, they do not explore iterative refinement.

Our proposed approach addresses these shortcomings. Specifically, our contributions are:

- *A novel vector momentum-parameterized stationary velocity field registration model (vSVF).* The vector momentum field allows decoupling transformation smoothness and the prediction of the transformation parameters. Hence, sufficient smoothness of the resulting velocity field can be guaranteed and diffeomorphisms can be obtained even for large displacements.
- *An end-to-end registration method, merging affine and vSVF registration into a single framework.* This framework achieves comparable performance to the corresponding optimization-based method and state-of-the-art registration approaches while dramatically reducing the computational cost.
- *A multi-step approach* for the affine and the vSVF registration components in our model, which allows refining registration results.
- *An entire registration model via map compositions* to avoid unnecessary image interpolations.
- *An inverse consistency loss both for the affine and the vSVF registration components* thereby encouraging the regression model to learn a mapping which is less dependent on image ordering. *I.e.*, registering image A to B will result in similar spatial correspondences as registering B to A.

Our approach facilitates image registration including affine pre-registration within one unified regression model.

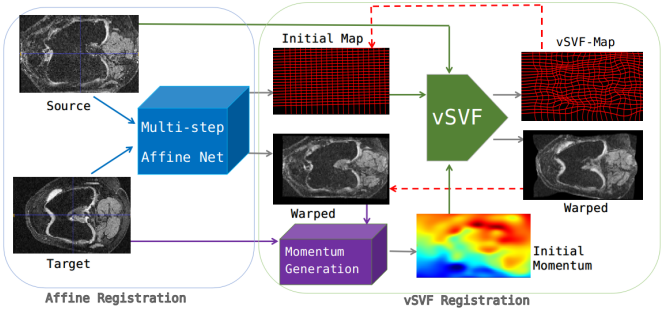

Figure 1. Our framework consists of affine (left) and vSVF (right) registration components. The affine part outputs the affine map and the affinely warped source image. The affine map initializes the map of the vSVF registration. The affinely warped image and the target image are input into the momentum generation network to predict the momentum of the vSVF registration model. The outputs of the vSVF component are the composed transformation map and the warped source image, which can be either taken as the final registration result or fed back (indicated by the dashed line) into the vSVF component to refine the registration solution.

In what follows, we refer to our approach as AVSM (Affine-vSVF-Mapping). Fig. 1 shows an overview of the AVSM framework illustrating the combination of the affine and the vSVF registration components. The affine and the vSVF components are designed independently, but easy to combine. In the affine stage, a multi-step affine network predicts affine parameters for an image pair. In the vSVF stage, a Unet-like network generates a momentum, from which a velocity field can be computed via smoothing. The initial map and the momentum are then fed into the vSVF component to output the sought-for transformation map. A specified number of iterations can also be used to refine the results. The entire registration framework operates on maps and uses map compositions. In this way, the source image is only interpolated once thereby avoiding image blurring. Furthermore, as the transformation map is assumed to be smooth, interpolations to up-sample the map are accurate. Therefore, we can obtain good registration results by predicting a down-sampled transformation. However, the similarity measure is evaluated at full resolution during training. Computing at low resolution greatly reduces the computational cost and allows us to compute on larger image volumes given a particular memory budget. *E.g.*, a map with 1/2 the size only requires 1/8 of the computations and 1/8 of the memory in 3D.

We compare AVSM to publicly available optimization-based methods [20, 17, 24, 19, 2] on longitudinal and cross-subject registrations of 3D image pairs of the OAI dataset.

The manuscript is organized as follows: Sec. 2 describes our ASVM approach; Sec. 3 shows experimental results; Sec. 4 presents conclusions and avenues for future work.

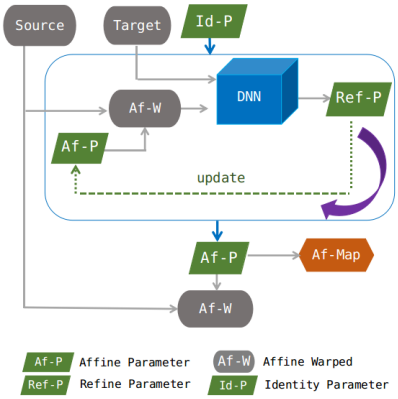

Figure 2. Multi-step affine network structure. As in a recurrent network, the parameters of the affine network are shared by all steps. At each step, the network outputs the parameters to refine the previously predicted affine transformation. *I.e.*, the current estimate is obtained by composition (indicated by dashed line). The overall affine transformation is obtained at the last step.

2. Methods

This section explains our overall approach. It is divided into two parts. The first part explains the affine registration component which makes use of a multi-step network to refine predictions of the affine transformation parameters. The second part explains the vector momentum-parameterized stationary velocity field (vSVF) which accounts for local deformations. Here, a momentum generation network first predicts the momentum parameterizing the vSVF model and therefore the transformation map. The vSVF component can also be applied in a multi-step way thereby further improving registration results.

2.1. Multi-step Affine Network

Most existing non-parametric registration approaches are not invariant to affine transformations as they are penalized by the regularizers. Hence, non-parametric registration approaches typically start from pre-registered image pairs, most typically based on affine registration, to account for large, global displacements or rotations. Therefore, in the first part of our framework, we use a multi-step affine network directly predicting the affine registration parameters and the corresponding transformation map.

The network needs to be flexible enough to adapt to both small and large affine deformations. Although deep convolutional networks can have large receptive fields, our experiments show that training a single affine network does not perform well in practice. Instead, we compose the affine transformation from several steps. This strategy results in significant improvements in accuracy and stability.

Network: Our multi-step affine network is a recurrent network, which progressively refines the predicted affine transformation. Fig. 2 shows the network architecture. To avoid numerical instabilities and numerical dissipation due to successive trilinear interpolations, we directly update the affine registration parameters rather than resampling images in intermediate steps. Specifically, at each step we take the target image and the warped source image (obtained via interpolation from the source image using the previous affine parameters) as inputs and then output the new affine parameters for the transformation refinement. Let the affine parameters be $\Gamma = \begin{pmatrix} A & b \end{pmatrix}$, where $A \in \mathbb{R}^{d\times d}$ represents the linear transformation matrix; $b \in \mathbb{R}^d$ denotes the translation and d is the image dimension. The update rule is as follows:

$$\begin{aligned} &A_{(t)} = \tilde{A}_{(t)}A_{(t-1)},\ b_{(t)} = \tilde{A}_{(t)}b_{(t-1)} + \tilde{b}_{(t)}, \\ &s.t. \quad A_{(0)} = I,\ b_{(0)} = 0. \end{aligned} \tag{1}$$

Here, $\tilde{A}_{(t)}$, $A_{(t)}$ represent the linear transformation matrix output and the composition result at the t-th step, respectively. Similarly, $\tilde{b}_{(t)}$ denotes the affine translation parameter output at the t-th step and $b_{(t)}$ the composition result. Finally, if we consider the registration from the source image to the target image in the space of the target image, the affine map is obtained by $\Phi_a^{-1}(x,\Gamma) = A_{(t_{last})}x + b_{(t_{last})}$.

Loss: The loss of the multi-step affine network consists of three parts: an image similarity loss $L_{a\text{-}sim}$, a regularization loss $L_{a\text{-}reg}$ and a loss encouraging transformation symmetry $L_{a\text{-}sym}$. Let us denote I_0 as the source image and I_1 as the target image. The superscripts st and ^{ts} denote registrations from I_0 to I_1 and I_1 to I_0, respectively[2].

The **image similarity loss** $L_{a\text{-}sim}(I_0, I_1, \Phi_a^{-1})$ can be any standard similarity measure, *e.g.*, Normalized Cross Correlation (NCC), Localized NCC (LNCC), or Mean Square Error (MSE). Here we generalize LNCC to a multi-kernel LNCC formulation (mk-LNCC). Standard LNCC is computed by averaging NCC scores of overlapping sliding windows centered at *sampled* voxels. Let V be the volume of the image; x_i, y_i refer to the i^{th} ($i \in \{1,..,|V|\}$) voxel in the warped source and target volumes, respectively. N_s refers to the number of sliding windows with cubic size $s\times s\times s$. Let ζ_j^s refer to the window centered at the j^{th} voxel and $\bar{x}_j, \bar{y}_j$ to the average image intensity values over ζ_j^s in the warped source and target image, respectively. LNCC with window size s, denoted as κ_s, is defined by

$$\kappa_s(x,y) = \frac{1}{N_s}\sum_j \frac{\sum\limits_{i\in\zeta_j^s}(x_i-\bar{x}_j)(y_i-\bar{y}_j)}{\sqrt{\sum\limits_{i\in\zeta_j^s}(x_i-\bar{x}_j)^2 \sum\limits_{i\in\zeta_j^s}(y_i-\bar{y}_j)^2}}. \tag{2}$$

We define mk-LNCC as a weighted sum of LNCCs with different window sizes. For computational efficiency LNCC can be evaluated over windows centered over a subset of

[2]To simplify the notation, we omit st (source to target registration) in what follows and only emphasize ^{ts} (target to source registration).

voxels of V. The image similarity loss is then

$$L_{a\text{-}sim}(I_0, I_1, \Gamma) = \sum_i \omega_i \kappa_{s_i}(I_0 \circ \Phi_a^{-1}, I_1),$$
$$\text{s.t.} \quad \Phi_a^{-1}(x, \Gamma) = Ax + b \quad \text{and} \quad \sum_i \omega_i = 1, w_i \geq 0. \tag{3}$$

The **regularization loss** $L_{a\text{-}reg}(\Gamma)$ penalizes deviations of the composed affine transform from the identity:

$$L_{a\text{-}reg}(\Gamma) = \lambda_{ar}(||A - I||_F^2 + ||b||_2^2), \tag{4}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\lambda_{ar} \geq 0$ is an epoch-dependent weight factor designed to be large at the beginning of the training to constrain large deformations and then gradually decaying to zero. See Eq. 14 for details.

The **symmetry loss** $L_{a\text{-}sym}(\Gamma, \Gamma^{ts})$ encourages the registration to be inverse consistent. *I.e.*,, we want to encourage that the transformation computed from source to target image is the inverse of the transformation computed from the target to the source image (*i.e.*, $A^{ts}(Ax + b) + b^{ts} = x$):

$$L_{a\text{-}sym}(\Gamma, \Gamma^{ts}) = \lambda_{as}(||A^{ts}A - I||_F^2 + ||A^{ts}b + b^{ts}||_2^2), \tag{5}$$

where $\lambda_{as} \geq 0$ is a chosen constant.

The **complete loss** $\mathcal{L}_a(I_0, I_1, \Gamma, \Gamma^{ts})$ is then:

$$\begin{aligned}\mathcal{L}_a(I_0, I_1, \Gamma, \Gamma^{ts}) =& \ell_a(I_0, I_1, \Gamma) + \ell_a(I_1, I_0, \Gamma^{ts}) \\ &+ L_{a\text{-}sym}(\Gamma, \Gamma^{ts}),\end{aligned} \tag{6}$$

where $\ell_a(I_0, I_1, \Gamma) = L_{a\text{-}sim}(I_0, I_1, \Gamma) + L_{a\text{-}reg}(\Gamma)$.

2.2. Vector Momentum-parameterized SVF

This section presents the momentum based stationary velocity field method followed by the network to predict the momentum. For simplicity, we describe the one step vSVF here, which forms the basis of the multi-step approach.

vSVF Method: To capture large deformations and to guarantee diffeomorphic transformations, registration algorithms motivated by fluid mechanics are frequently employed. Here, the transformation map Φ [3] in source image space is obtained via time-integration of a velocity field $v(x, t)$, which needs to be estimated. The governing differential equation is: $\Phi_t(x, t) = v(\Phi(x, t), t)$, $\Phi(x, 0) = \Phi_{(0)}(x)$, where $\Phi_{(0)}$ is the initial map. For a sufficiently smooth velocity field v one obtains a diffeomorphic transformation. Sufficient smoothness is achieved by penalizing non-smoothness of v. Specifically, the optimization problem is

$$v^* = \underset{v}{\text{argmin}}\, \lambda_{vr} \int_0^1 \|v\|_L^2 \, \mathrm{d}t + \text{Sim}[I_0 \circ \Phi^{-1}(1), I_1],$$
$$\text{s.t.} \quad \Phi_t^{-1} + D\Phi^{-1}v = 0 \quad \text{and} \quad \Phi^{-1}(0) = \Phi_{(0)}^{-1}, \tag{7}$$

where D denotes the Jacobian and $\|v\|_L^2 = \langle L^\dagger Lv, v\rangle$ is a spatial norm defined by specifying the differential operator L and its adjoint $L^\dagger$. As the vector-valued momentum m is equivalent to $m = L^\dagger Lv$, one can express the norm as $\|v\|_L^2 = \langle m, v\rangle$. In the LDDMM approach [5], time-dependent vector fields $v(x, t)$ are estimated. A slightly simpler approach is to use a *stationary velocity field* (SVF) $v(x)$ [18]. The rest of the formulation remains the same. While the SVF registration algorithms optimize directly over the velocity field v, we propose a *vector momentum SVF (vSVF)* formulation which is computed as

$$m^* = \underset{m_0}{\text{argmin}}\, \lambda_{vr}\langle m_0, v_0\rangle + \text{Sim}[I_0 \circ \Phi^{-1}(1), I_1], \text{ s.t.}$$
$$\Phi_t^{-1} + D\Phi^{-1}v = 0, \Phi^{-1}(0) = \Phi_{(0)}^{-1}, v_0 = (L^\dagger L)^{-1}m_0, \tag{8}$$

where m_0 denotes the vector momentum and $\lambda_{vr} > 0$ is a constant. This formulation can be considered a simplified version of the vector momentum-parameterized LDDMM formulation [30]. The benefit of such a formulation is that it allows us to explicitly control spatial smoothness as the deep network predicts the momentum which gets subsequently smoothed to obtain the velocity field, instead of predicting the velocity field v directly which would then require the network to learn to predict a smooth vector field.

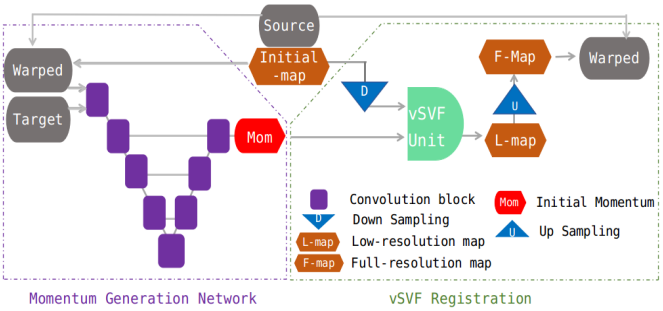

Figure 3. vSVF registration framework illustration (one step), including the momentum generation network and the vSVF registration. The network outputs a low-resolution momentum. The momentum and the down-sampled initial map are input to the vSVF unit outputting a low-resolution transformation map, which is then up-sampled to full resolution before warping the source image.

Fig. 3 illustrates the framework of the vector momentum-parameterized stationary velocity field (vSVF) registration. We compute using a low-resolution velocity field, which greatly reduces memory consumption. The framework consists of two parts: 1) a momentum generation network taking as the input the warped source image, together with the target image, outputting the low-resolution momentum; 2) the vSVF registration part. Specifically, the predicted momentum and the down-sampled initial map are input into the vSVF unit, the output of which is finally up-sampled to obtain the full resolution transformation map. Inside the vSVF unit, a velocity field is obtained by smoothing the momentum and then used to solve the advection equation, $\Phi_{(\tau)t}^{-1} + D\Phi_{(\tau)}^{-1}v = 0$, for unit time (using several discrete time points). This then results in the sought-for transformation map. The initial map mentioned here can be the affine map or the map obtained from a previous vSVF step,

[3] The subscript $_v$ of Φ_v is omitted, where v refers to vSVF method.

namely for the τ-th step, set $\Phi^{-1}_{(\tau)}(x,0) = \Phi^{-1}_{(\tau-1)}(x,1)$.

Momentum Generation Network: We implement a deep neural network to generate the vector momentum. As our work does not focus on the network architecture, we simply implement a four-level U-net with residual links [23, 16]. Implementation details can be found in the supplementary material. During training, the gradient is first backpropagated through the integrator for the advection equation followed by the momentum generation network. This can require a lot of memory. Therefore, to reduce memory requirements, the network outputs a low-resolution momentum. In practice, we remove the last decoder level of the U-net. In this case, the remaining vSVF component also operates on the low-resolution map.

Loss: Similar to the loss in the affine network, the loss for the vSVF part of the network also consists of three terms: a similarity loss $L_{v\text{-}sim}$, a regularization loss $L_{v\text{-}reg}$ and a symmetry loss $L_{v\text{-}sym}$.

The **similarity loss** $L_{v\text{-}sim}(I_0, I_1, \Phi^{-1})$ is the same as for the affine network. *I.e.*, we also use mk-LNCC.

The **regularization loss** $L_{v\text{-}reg}(m_0)$ penalizes the velocity field. Thus, we have

$$L_{v\text{-}reg}(m_0) = \lambda_{vr}\|v\|^2_L = \lambda_{vr}\langle m_0, v_0\rangle, \tag{9}$$

where $v_0 = (L^\dagger L)^{-1} m_0$. We implement $(L^\dagger L)^{-1}$ as a convolution with a multi-Gaussian kernel [21].

The **symmetric loss** is defined as

$$L_{v\text{-}sym}(\Phi^{-1}, (\Phi^{ts})^{-1}) = \lambda_{vs}\|\Phi^{-1}\circ(\Phi^{ts})^{-1} - id\|^2_2, \tag{10}$$

where id denotes the identity map, $\lambda_{vs} \geq 0$ refers to the symmetry weight factor, $(\Phi^{ts})^{-1}$ denotes the map obtained from registering the target to the source image in the space of the source image and Φ^{-1} denotes the map obtained from registering the source image to the target image in the space of the target image. Consequentially, the composition also lives in the target image space.

The **complete loss** $\mathcal{L}_v(I_0, I_1, \Phi^{-1}, (\Phi^{ts})^{-1}, m_0, m_0^{ts})$ for vSVF registration with one step is as follows:

$$\begin{aligned}\mathcal{L}_v(I_0, I_1, \Phi^{-1}, (\Phi^{ts})^{-1}, m_0, m_0^{ts}) &= \ell_v(I_0, I_1, \Phi^{-1}, m_0) \\ &+ \ell_v(I_1, I_0, (\Phi^{ts})^{-1}, m_0^{ts}) \\ &+ L_{v\text{-}sym}(\Phi^{-1}, (\Phi^{ts})^{-1}),\end{aligned} \tag{11}$$

where:
$\ell_v(I_0, I_1, \Phi^{-1}, m_0) = L_{v\text{-}sim}(I_0, I_1, \Phi^{-1}) + L_{v\text{-}reg}(m_0)$.

For the vSVF model with T steps, the complete loss is:

$$\begin{aligned}&\sum_{\tau=1}^{T} \mathcal{L}_v(I_0, I_1, \Phi^{-1}_{(\tau)}, {\Phi^{ts}_{(\tau)}}^{-1}, m_{0(\tau)}, m^{ts}_{0(\tau)}) \quad \text{s.t.} \\ &\Phi^{-1}_{(\tau)}(x,0) = \Phi^{-1}_{(\tau-1)}(x,1), \\ &(\Phi^{ts}_{(\tau)})^{-1}(x,0) = (\Phi^{ts}_{(\tau-1)})^{-1}(x,1).\end{aligned} \tag{12}$$

3. Experiments and Results

Dataset: The Osteoarthritis Initiative (OAI) dataset consists of 176 manually labeled magnetic resonance (MR) images from 88 patients (2 longitudinal scans per patient) and 22,950 unlabeled MR images from 2,444 patients. Labels are available for femoral and tibial cartilage. All images are of size $384 \times 384 \times 160$, where each voxel is of size $0.36 \times 0.36 \times 0.7mm^3$. We normalize the intensities of each image such that the 0.1th percentile and the 99.9th percentile are mapped to $0, 1$ and clamp values that are smaller to 0 and larger to 1 to avoid outliers. All images are downsampled to size $192 \times 192 \times 80$.

Evaluation: We evaluate on both longitudinal and cross-subject registrations. We divide the unlabeled patients into a training and a validation group, with a ratio of 7:3. For the *longitudinal registrations*, 4,200 pairs from the training group (obtained by swapping the source and the target from 2,100 pairs of images) are randomly selected for training, and 50 pairs selected from the validation group are used for validation. All 176 longitudinal pairs with labels are used as our test set. For the *cross-subject registrations*, we randomly pick 2,800 (from 1,400 pairs) cross-subject training pairs and 50 validation pairs; 300 pairs (from 150 pairs) are randomly selected as the test set. We use the average Dice score [11] over all testing pairs as the evaluation metric.

Training details: The training stage includes two parts:

1) Training multi-step affine net: It is difficult to train the multi-step affine network from scratch. Instead, we train a single-step network first and use its parameters to initialize the multi-step network. For longitudinal registration, we train with a three-step affine network, but use a seven-step network during testing. This results in better testing performance than a three-step network. Similarly, for cross-subject registration we train with a five-step network and test with a seven-step one. The affine symmetry factor λ_{as} is set to 10.

2) Training momentum generation network: During training, the affine part is fixed. For vSVF, we use 10 time-steps and a multi-Gaussian kernel with standard deviations $\{0.05, 0.1, 0.15, 0.2, 0.25\}$ and corresponding weights $\{0.067, 0.133, 0.2, 0.267, 0.333\}$ (spacing is scaled so that the image is in $[0,1]^3$). We train with two steps for both longitudinal and cross-subject registrations. The vSVF regularization factor λ_{vr} is set to 10 and the symmetry factor λ_{vs} is set to 1e-4. For both parts, we use the same training strategy: 1 pair per batch, 400 batches per epoch, 200 epochs per experiment; we set a learning rate of 5e-4 with a decay factor of 0.5 after every 60 epochs. We use mk-LNCC as the similarity measure with $(\omega, s) = \{(0.3, S/4), (0.7, S/2)\}$, where S refers to the smallest image dimension. Besides, in our implementation of mk-LNCC, we set the sliding window stride to $S/4$ and kernel dilation to 2.

Additionally, the affine regularization factor λ_{ar} is *epoch-dependent* during training and defined as:

$$\lambda_{ar} := \frac{C_{ar} K_{ar}}{K_{ar} + e^{n/K_{ar}}}, \tag{13}$$

where C_{ar} is a constant, K_{ar} controls the decay rate, and n is the epoch count. In both longitudinal and cross-subject experiments, K_{ar} is set to 4 and C_{ar} is set to 10.

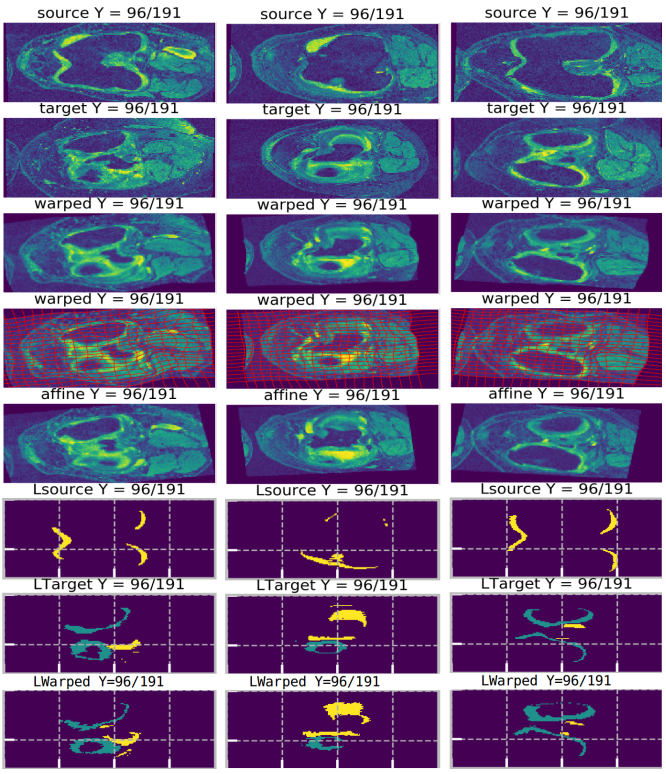

Figure 4. Illustration of registration results achieved by AVSM, each column refers to an example. The first five rows refer to source, target, warped image by AVSM, warped image with deformation grid (visualizing Φ^{-1}), warped image by multi-step affine respectively, followed by source label, target label and warped label by AVSM separately. There is high similarity between the warped and the target labels and the deformations are smooth.

Baseline methods: We implement the corresponding numerically optimized versions (*e.g.*, directly optimizing the momentum) of affine (affine-opt) and vSVF (vSVF-opt) registrations. We compare with three widely-used public registration methods: *SyN* [2, 1], *Demons* [29, 28] and *NiftyReg* [20, 17, 24, 19]. Besides, we also compare the most recent VoxelMorph variant [9]. We report their performance after an in-depth search for good parameters. For Demons, SyN and NiftyReg, we use isotropic voxel spacing $1 \times 1 \times 1 mm^3$ as this gives improved results compared to using physical spacing. This implies anisotropic regularization in physical space. For our approaches, isotropic or anisotropic regularization in physical space gives similar results. Hence, we choose the more natural isotropic regularization in physical space.

Optimization-based multi-scale affine registration: Instead of optimizing for the affine parameters on a single image scale, we use a multi-scale strategy. Specifically, we start at a low image-resolution, where affine parameters are roughly estimated, and then use them as the initialization for the next higher scale. Stochastic gradient descent is used with a learning rate of 1e-4. Three image scales $\{0.25, 0.5, 1.0\}$ are used, each with $\{200, 200, 50\}$ iterations. We use mk-LNCC as the similarity measure. At each scale k, let image size (smallest length among image dimensions) be S_k, here $k \in \{0.25, 0.5, 1.0\}$. At scale 1.0, parameters are set to $(\omega, s) = \{(0.3, S_k/4), (0.7, S_k/2)\}$, *i.e.*, the same parameters as for the network version; at scales 0.5 and 0.25, $(\omega, s) = \{(1.0, S_k/2)\}$.

Optimization-based multi-scale vSVF registration: We take the affine map (resulting from the optimization-based multi-scale affine registration) as the initial map and then numerically optimize the vSVF model. The same multi-scale strategy as for the affine registration is used. The momentum is up-sampled between scales. We use L-BGFS [15] for optimization. In our experiments, we use three scales $\{0.25, 0.5, 1.0\}$ with 60 iterations per scale. The same mk-LNCC similarity measure as for the *optimization-based multi-scale affine registration* is used. The number of time steps for the integration of the advection equation and the settings for the multi-Gaussian kernel are the same as for the proposed deep network model.

NiftyReg: We run two registration phases: affine followed by B-spline registration. Three scales are used in each phase and the interval of the B-spline control points is set to 10 voxels. In addition, we find that using LNCC as the similarity measure, with a standard deviation of 40 for the Gaussian kernel, performs better than the default Normalized Mutual Information, but introduces folds in the transformation. In LNCC experiments, we therefore use a log of the Jacobi determinant penalty of 0.01 to reduce folds.

Demons: We take the affine map obtained from NiftyReg as the initial map and use the *Fast Symmetric Forces Demons Algorithm* [29] via SimpleITK. The Gaussian smoothing standard deviation for the displacement field is set to 1.2. We use MSE as the similarity measure.

SyN: We compare with Symmetric Normalization (SyN), a widely used registration method implemented in the ANTs software package [1]. We take Mattes as the metric for affine registration, and take CC with sampling radius set to 4 for SyN registration. We use multi-resolution optimization with four scales with {2100, 1200, 1200, 20} iterations; the standard deviation for Gaussian smoothing at each level is set to {3, 2, 1, 0}. The flow standard deviation to smooth the gradient field is set to 3.

VoxelMorph: We compute results based on the most re-

Method	Longitudinal		Cross-subject		
	Dice	Folds	Dice	Folds	Time (s)
affine-NiftyReg	75.07 (6.21)	0	30.43 (12.11)	0	45
affine-opt	**78.61** (4.48)	0	34.49 (18.07)	0	8
affine-net (7-step)	77.75 (4.77)	0	**44.58** (7.74)	0	0.20
Demons	**83.43** (2.64)	10.7 [0.56]	63.47 (9.52)	19.0 [0.56]	114
SyN	83.13 (2.67)	0	65.71 (15.01)	0	1330
NiftyReg-NMI	83.17 (2.76)	0	59.65 (7.62)	0	143
NiftyReg-LNCC	83.35 (2.70)	0	67.92 (5.24)	203.3 [35.19]	270
vSVF-opt	82.99 (2.68)	0	67.35 (9.73)	0	79
VoxelMorph(w/o aff)	71.25 (9.54)	2.72 [1.57]	46.06 (14.94)	83.0 [18.13]	0.12
VoxelMorph(with aff)	82.54 (2.78)	5.85 [0.59]	66.08 (5.13)	39.0 [3.31]	0.31
AVSM (2-step)	82.60 (2.73)	0	67.59 (4.47)	5.5 [0.39]	0.62
AVSM (3-step)	82.67 (2.74)	3.4 [0.12]	**68.40** (4.35)	14.3 [1.07]	0.83

Table 1. Dice scores (standard deviation) of different registration methods for longitudinal and cross-subject registrations on the OAI dataset. *Affine-opt* and *vSVF-opt* refer to optimization-based multi-scale affine and vSVF registrations. *AVSM* (n-step) refers to a seven-step affine network and an n-step vSVF model. *Folds* ($|\{x : J_\phi(x) < 0\}|$) refers to the average number of folds and corresponding absolute Jacobi determinant value in square brackets; *Time* refers to the average time per image registration.

cent VoxelMorph variant [9], which is also a deep-learning based. As VoxelMorph assumes that images are pre-aligned, for a fair comparison, we therefore initialized it via our proposed multi-step affine network. Best parameters are determined via grid search.

NiftyReg, *Demons* and *SyN* are run on a server with i9-7900X (10 cores @ 3.30GHz) , while all other methods run on a single NVIDIA GTX 1080Ti.

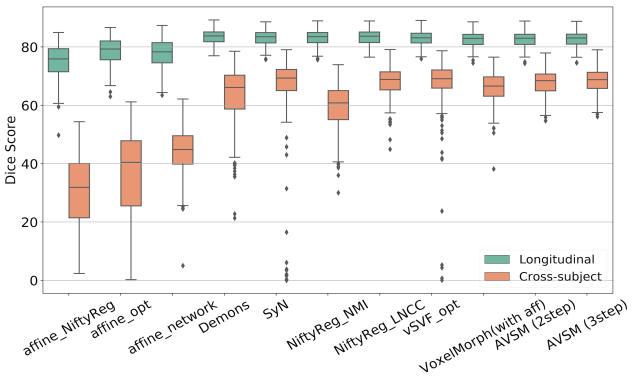

Figure 5. Box-plots of the performance of the different registration methods for longitudinal registration (green) and cross-subject registration (orange). Both AVSM and NiftyReg (LNCC) show high performance and small variance.

Tab.1 compares the performance of our framework with its corresponding optimization version and public registration tools. Overall, our AVSM framework performs best in cross-subject registration and achieves slightly better performance than optimization-based methods, both for affine and non-parametric registrations. NiftyReg with LNCC shows similar performance. For longitudinal registration, AVSM shows good performance, but slightly lower than the optimization-based methods, including vSVF-opt which AVSM is based on. A possible explanation is that for longitudinal registrations deformations are subtle and source/target image pairs are very similar in appearance. Hence, numerical optimization can very accurately align such image-pairs at convergence. VoxelMorph runs fastest among all the methods. Without initial affine registration, it unsurprisingly performs poorly. Once the input pair is well pre-aligend, VoxelMorph shows competitive results for longitudinal registrations, but is outperformed by our approach for the more challenging cross-subject registration. To evaluate the smoothness of the transformation map, we compute the determinant of the Jacobian of the estimated map, $J_\phi(x) := |D\phi^{-1}(x)|$, and count folds defined by $|\{x : J_\phi(x) < 0\}|$ in each image ($192 \times 192 \times 80$ voxels in total). We also report the absolute value of the determinant of the Jacobian in these cases indicating the severity of the fold. Even though the regularization is used, numerical optimization (vSVF-opt) always results in diffeomorphisms, but very few folds remain in AVSM for cross-subject registration. This may be caused by numerical discretization artifacts, by very large predicted momenta, or by inaccuracies of the predictions with respect to the numerical optimization results. Fig. 5 shows the corresponding boxplot results. AVSM achieves small variance and high performance in both registration tasks and exhibits less registration failures (outliers). As AVSM only requires one forward pass to complete both the affine and the vSVF registration, it is much faster than using iterative numerical optimization.

Tab. 2 shows results for an ablation study on AVSM. For the affine part, it is difficult to train the single-step affine network without the regularization term. Hence, registrations fail. Introducing multi-step and inverse consistency boosts the affine performance. Compared with using NCC as similarity measure, our implementation of mk-LNCC improves results greatly. In the following vSVF part, we observe a large difference between methods IV and VI, illustrating that vSVF registration results in large improvements. Adding mk-LNCC and multi-step training in methods VII and VIII further improves performance. The exception is the vSVF symmetry loss which slightly worsens the performance for both longitudinal and cross-subject registration, but results in good symmetry measures (see Fig. 6).

We still retain the symmetric loss as it helps the network to converge to solutions with smoother maps as shown in Fig. 6. Instead of using larger Gaussian kernels, which can remove local displacements, penalizing asymmetry helps regularize the deformation without smoothing the map too much and without sacrificing too much performance. To numerically evaluate the symmetry, we compute $\ln(\frac{1}{|V|}\|\Phi^{-1} \circ (\Phi^{ts})^{-1} - id\|_2^2)$ for all registration methods, where V refers to the volume size and Φ the map obtained via composition of the affine and the deformable transforms. Since differ-

Method	Af-Reg	Af-Multi	Af-Sym	Af-MK	vSVF	vSVF-MK	vSVF-Multi	vSVF-Sym	Longitudinal	Better?	Cross-subject	Better?
I									-		-	
II	✓								55.41	✓	28.68	✓
III	✓	✓							64.78	✓	36.31	✓
IV	✓	✓	✓						68.87	✓	37.54	✓
V	✓	✓	✓	✓					77.75	✓	44.58	✓
VI	✓	✓	✓		✓				80.71	✓	59.21	✓
VII	✓	✓	✓	✓	✓	✓			81.64	✓	64.56	✓
VIII	✓	✓	✓	✓	✓	✓	✓		82.81	✓	69.08	✓
IV	✓	✓	✓	✓	✓	✓	✓	✓	82.67	✗	68.40	✗

Table 2. Ablation study of AVSM using different combinations of methods. **Af-** and **vSVF-** separately refer to the affine and to the vSVF related methods; *Reg* refers to adding epoch-dependent regularization; *Multi* refers to multi-step training and testing; *Sym* refers to adding the symmetric loss; *MK* refers to using mk-LNCC as similarity measure (default NCC). Except for the last approach which uses *vSVF-Sym* (last row) and encourages symmetric vSVF solutions, all other approaches result in performance improvements.

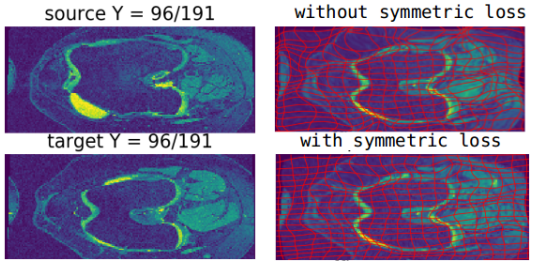

Figure 6. Illustration of symmetric loss for AVSM. The left column shows the source and target images. The right column shows the warped image from a network trained with and without symmetric loss. The deformation with symmetric loss is smoother.

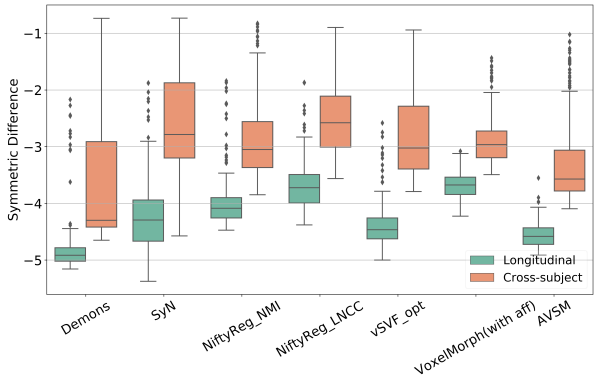

Figure 7. Box-plots of the symmetry evaluation (the lower the better) of different registration methods for longitudinal registration (green) and cross-subject registration (orange). AVSM (tested with two-step vSVF) shows good results.

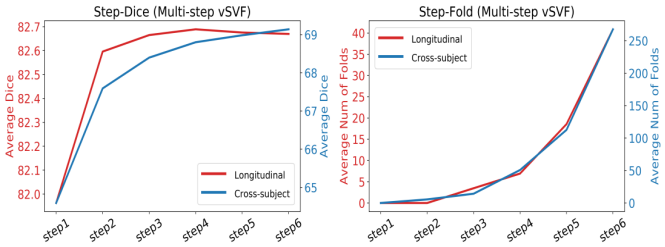

Figure 8. Multi-step vSVF registration results for two-step vSVF training. Performance increases with steps (left), but the number of folds also increases (right).

ent methods treat boundaries differently, we only evaluate this measure in the interior of the image volume (10 voxels away from the boundary). Fig. 7 shows the results. AVSM obtains low values for both registration tasks, confirming its good symmetry properties. Both the Demons and SyN also encourage symmetry, but only AVSM shows a nice compromise between accuracy and symmetry.

Fig. 8 shows the average Dice sores over the number of test iteration steps of vSVF. The model is trained using a two-step vSVF. It can be observed that iterating the model for more than two steps can increase performance as these iterations result in registration refinements. However, the average number of folds also increases, mostly at boundary regions and in regions of anatomical inconsistencies. Examples are shown in the supplementary material.

4. Conclusions and Future Work

We introduced an end-to-end 3D image registration approach (AVSM) consisting of a multi-step affine network and a deformable registration network using a momentum-based SVF algorithm. AVSM outputs a transformation map which includes an affine pre-registration *and* a vSVF non-parametric deformation in a single forward pass. Our results on cross-subject and longitudinal registration of knee MR images show that our method achieves comparable and sometimes better performance to popular registration tools with a dramatically reduced computation time and with excellent deformation regularity and symmetry. Future work will focus on also learning regularizers and evaluations on other registration tasks, *e.g.* in the brain and the lung.

Acknowledgements: Research reported in this publication was supported by the National Institutes of Health (NIH) and the National Science Foundation (NSF) under award numbers NSF EECS1711776 and NIH 1R01AR072013. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH or the NSF.

A. Supplementary material

This supplementary material provides additional details illustrating the proposed approach. Specifically, Sec. A.1 describes how the affine training is regularized in an epoch-dependent way. Sec. A.2 shows registration performance for different numbers of steps for the affine registration-part of the network. Sec. A.3 details the structure of the momentum generation network. Lastly, Sec. A.4 shows additional registration examples.

A.1. Affine regularization factor

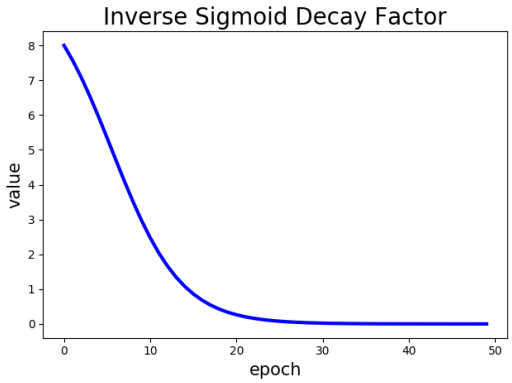

Figure 9. Graph of the affine regularization factor. Its value decays to zero over the epochs.

To help with convergence of the affine registration network, we use an epoch-dependent regularization factor, which discourages large transformations at the start of the training. Specifically, we define this epoch-dependent regularization factor as

$$\lambda_{ar} := \frac{C_{ar} K_{ar}}{K_{ar} + e^{n/K_{ar}}}, \tag{14}$$

where C_{ar} is a constant, K_{ar} controls the decay rate, and n is the epoch count. Fig. 9 shows the value of λ_{ar} plotted over the epochs. As the value decays to zero with the epochs, its influence on the training becomes negligible. For both longitudinal and cross-subject experiments, K_{ar} is set to 4 and C_{ar} is set to 10.

A.2. Dice over steps in Multi-step Affine Network

The main manuscript shows the average Dice scores over the number of test iteration steps for the vSVF registration component. For completeness, Fig. 10 shows the average Dice scores over the number of steps for the affine network. The model is trained using a three-step affine network for longitudinal registrations and using five steps for cross-subject registration. Similar to the vSVF registration, it can be observed that model performance improves with large number of steps and saturates at a high performance level.

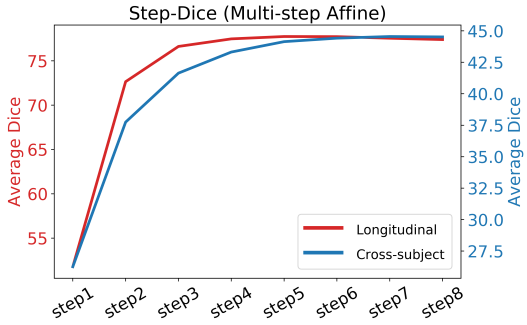

Figure 10. Multi-step Affine registration results over iteration steps. The affine network is trained using three steps for longitudinal registration (red) and five steps for cross-subject registration (blue). Performance increases with steps and finally saturates.

A.3. Structure of Momentum Generation Network

As the network structure itself is not the main contribution of our work, we do not describe it in detail in the main manuscript. For completeness, we describe the architecture here. Fig. 11 shows the structure of the Momentum Generation Network. It takes a pair of images as the input and outputs a low-resolution initial momentum. We use a four-level U-net [23, 16] with residual links, but remove the last decoder level to output the low-resolution momentum. As the momentum can be positive or negative, no activation function (*e.g.* ReLu [**?**] or leakyRelu [**?**]) is used after the last two convolutional layers, which output the momentum.

A.4. Visualization

To provide more insight into the registration behavior of our network, we visualize results illustrating deformation folds, results for different steps in the multi-step approach, and additional examples. Specifically, we show the following:

- *Folds*: To better visualize the folds produced by the multi-step vSVF, we report the registration results, shown in Fig. 12, from the six-step vSVF. These folds mostly occur at regions of anatomical inconsistency or at the image boundary where map interpolation artifacts may influence the solution. In these regions, very large momentum values may be predicted which can result in folds due to discretization artifacts when integrating the advection equation.

- *Multi-step in vSVF registration*: Fig. 13 shows the registration results over the steps of the vSVF. Although folds may result from the multi-step strategy in some very large deformation cases, the transformation maps are largely well regularized. We observe that the registration results improve over the steps.

- *More AVSM examples*: Fig. 14 shows additional

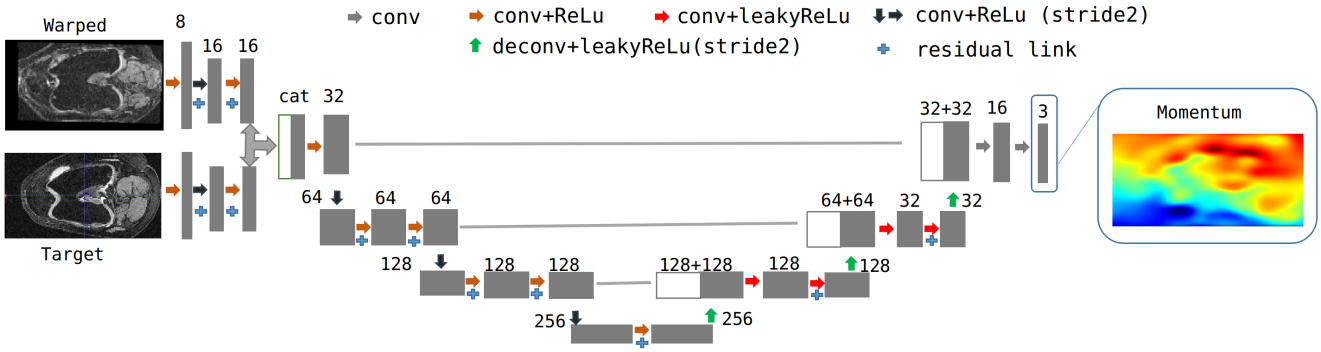

Figure 11. Illustration of the structure of Momentum Generation Network. It follows the structure of the U-net but the last level decoder is removed.

AVSM registration results. It can be observed that AVSM achieves good registration results with smooth transformation maps for cases with large and small deformations.

References

[1] Brian B Avants, Charles L Epstein, Murray Grossman, and James C Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis*, 12(1):26–41, 2008. 6

[2] Brian B Avants, Nick Tustison, and Gang Song. Advanced normalization tools (ANTS). *Insight j*, 2:1–35, 2009. 2, 6

[3] Ruzena Bajcsy and Stane Kovačič. Multiresolution elastic matching. *CVGIP*, 46(1):1–21, 1989. 1

[4] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. An unsupervised learning model for deformable medical image registration. In *CVPR*, pages 9252–9260, 2018. 1, 2

[5] M Faisal Beg, Michael I Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *IJCV*, 61(2):139–157, 2005. 1, 4

[6] Xiaohuan Cao, Jianhua Yang, Jun Zhang, Qian Wang, Pew-Thian Yap, and Dinggang Shen. Deformable image registration using a cue-aware deep regression network. *IEEE Transactions on Biomedical Engineering*, 65(9):1900–1911, 2018. 1

[7] Evelyn Chee and Joe Wu. Airnet: Self-supervised affine registration for 3d medical images using neural networks. *arXiv preprint arXiv:1810.02583*, 2018. 2

[8] Zhuoyuan Chen, Hailin Jin, Zhe Lin, Scott Cohen, and Ying Wu. Large displacement optical flow from nearest neighbor fields. In *CVPR*, pages 2443–2450, 2013. 1

[9] Adrian V Dalca, Guha Balakrishnan, John Guttag, and Mert R Sabuncu. Unsupervised learning for fast probabilistic diffeomorphic registration. *arXiv preprint arXiv:1805.04605*, 2018. 1, 2, 6, 7

[10] Bob D de Vos, Floris F Berendsen, Max A Viergever, Marius Staring, and Ivana Išgum. End-to-end unsupervised deformable image registration with a convolutional neural network. In *MLCDS*, pages 204–212. Springer, 2017. 1

[11] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. 5

[12] Gabriel L Hart, Christopher Zach, and Marc Niethammer. An optimal control approach for deformable registration. In *CVPR*, pages 9–16. IEEE, 2009. 1

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 2

[14] Hongming Li and Yong Fan. Non-rigid image registration using fully convolutional networks with deep self-supervision. *arXiv preprint arXiv:1709.00799*, 2017. 1

[15] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989. 6

[16] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016. 5, 9

[17] Marc Modat, David M Cash, Pankaj Daga, Gavin P Winston, John S Duncan, and Sébastien Ourselin. Global image registration using a symmetric block-matching approach. *Journal of Medical Imaging*, 1(2):024003, 2014. 2, 6

[18] Marc Modat, Pankaj Daga, M Jorge Cardoso, Sebastien Ourselin, Gerard R Ridgway, and John Ashburner. Parametric non-rigid registration using a stationary velocity field. In *2012 IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA)*, pages 145–150. IEEE, 2012. 4

[19] Marc Modat, Gerard R Ridgway, Zeike A Taylor, Manja Lehmann, Josephine Barnes, David J Hawkes, Nick C Fox, and Sébastien Ourselin. Fast free-form deformation using graphics processing units. *Computer methods and programs in biomedicine*, 98(3):278–284, 2010. 2, 6

[20] Sébastien Ourselin, Alexis Roche, Gérard Subsol, Xavier Pennec, and Nicholas Ayache. Reconstructing a 3D structure from serial histological sections. *Image and vision computing*, 19(1-2):25–31, 2001. 2, 6

[21] Laurent Risser, François-Xavier Vialard, Robin Wolz, Daryl D Holm, and Daniel Rueckert. Simultaneous fine and

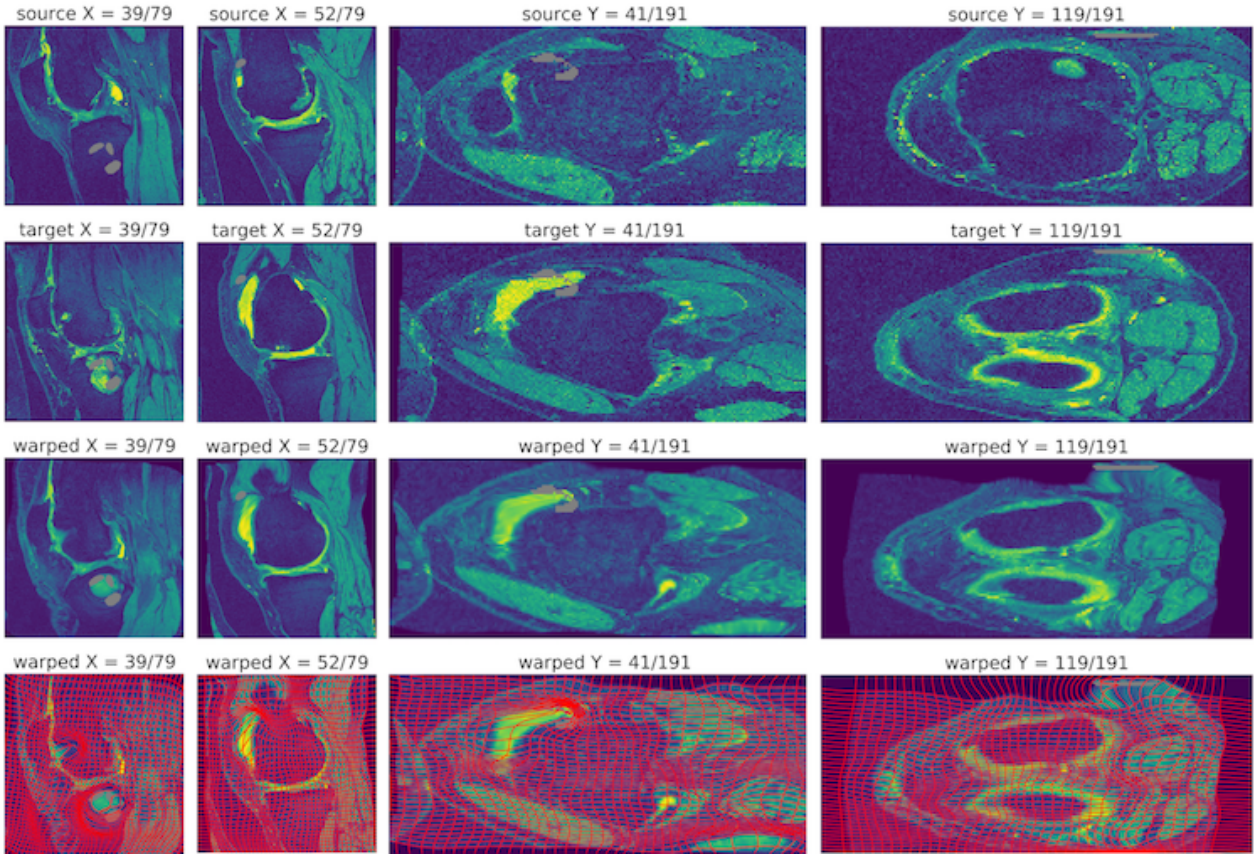

Figure 12. Examples of folds produced by a six-step vSVF (trained using a two-step vSVF). Each column refers to an example registration case. From top to bottom source, target, warped image by AVSM and warped image with deformation grid (visualizing Φ^{-1}) are shown. Folds are shown in *gray*. From left to right, the first three columns refer to cases with anatomical inconsistency and the last column refers to a case where the folds occur at the boundary.

coarse diffeomorphic registration: application to atrophy measurement in Alzheimers disease. In *MICCAI*, pages 610–617. Springer, 2010. 5

[22] Torsten Rohlfing, Calvin R Maurer, David A Bluemke, and Michael A Jacobs. Volume-preserving nonrigid registration of MR breast images using free-form deformation with an incompressibility constraint. *TMI*, 22(6):730–741, 2003. 1

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 5, 9

[24] Daniel Rueckert, Luke I Sonoda, Carmel Hayes, Derek LG Hill, Martin O Leach, and David J Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *TMI*, 18(8):712–721, 1999. 1, 2, 6

[25] Dinggang Shen and Christos Davatzikos. HAMMER: hierarchical attribute matching mechanism for elastic registration. *TMI*, 21(11):1421–1439, 2002. 1

[26] Marius Staring, Stefan Klein, and Josien PW Pluim. A rigidity penalty term for nonrigid registration. *Medical physics*, 34(11):4098–4108, 2007. 1

[27] Christine Tanner, Julia A Schnabel, Daniel Chung, Matthew J Clarkson, Daniel Rueckert, Derek LG Hill, and David J Hawkes. Volume and shape preservation of enhancing lesions when applying non-rigid registration to a time series of contrast enhancing MR breast images. In *MICCAI*, pages 327–337. Springer, 2000. 1

[28] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Symmetric log-domain diffeomorphic registration: A demons-based approach. In *MICCAI*, pages 754–761. Springer, 2008. 6

[29] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage*, 45(1):S61–S72, 2009. 1, 6

[30] F-X. Vialard, L. Risser, D. Rueckert, and C.J. Cotter. Diffeomorphic 3D image registration via geodesic shooting using an efficient adjoint calculation. *IJCV*, 97(2):229–241, 2012. 4

[31] Jonas Wulff and Michael J Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *CVPR*, pages 120–130, 2015. 1

[32] Xiao Yang, Roland Kwitt, and Marc Niethammer. Fast predictive image registration. In *DLMIA*, pages 48–57. Springer, 2016. 1

[33] Xiao Yang, Roland Kwitt, Martin Styner, and Marc Niethammer. Quicksilver: Fast predictive image registration–a deep

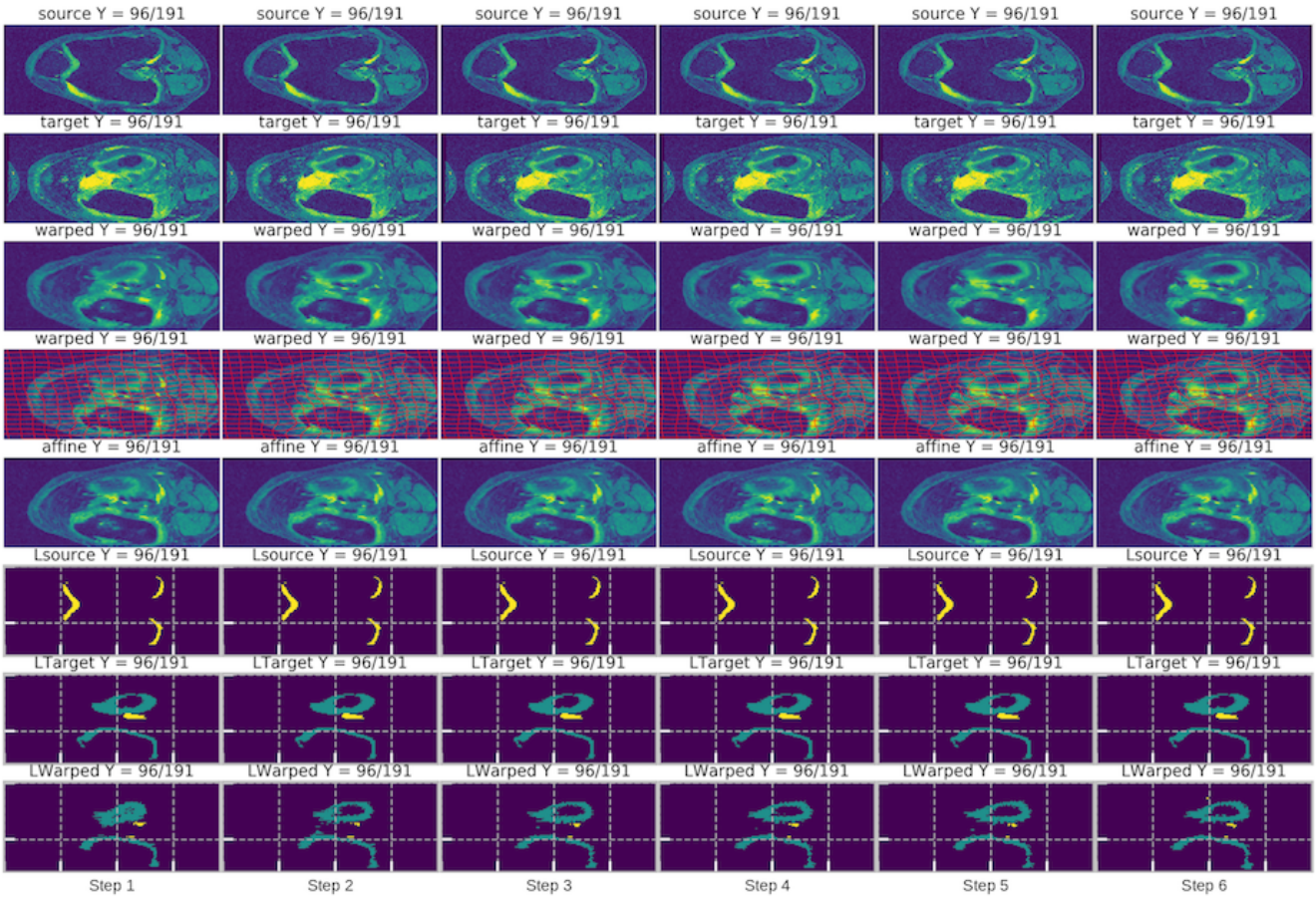

Figure 13. Illustration of the results of *one registration case (with six steps)* by AVSM (trained using a two-step vSVF). From left to right, each column shows results for different steps. The first five rows refer to source, target, warped image by AVSM, warped image with deformation grid (visualizing Φ^{-1}) and warped image by the multi-step affine network respectively. The last three rows show the source label, target label and warped label for the AVSM result. The transformation map gets refined over the six steps and the registration result improves as indicated by a better correspondence between the target label and the warped label images (last two rows).

learning approach. *NeuroImage*, 158:378–396, 2017. 1

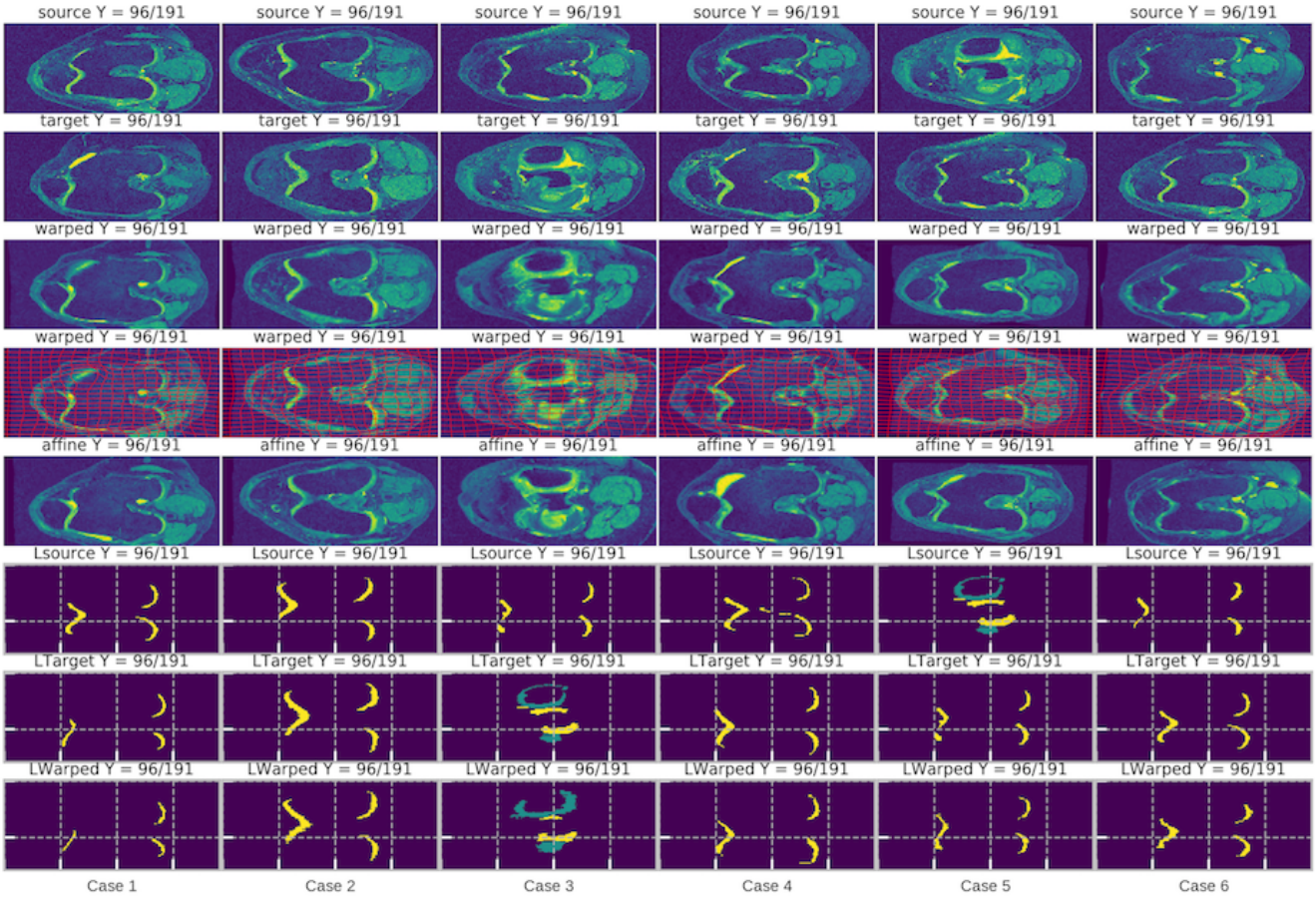

Figure 14. Illustration of results of *six registration cases* by AVSM. Each column refers to an example registration case. The first five rows refer to source, target, warped image by AVSM, warped image with deformation grid (visualizing Φ^{-1}) and warped image by the multi-step affine network respectively. The last three rows show the source label, target label and warped label by AVSM. There is high similarity between the warped and the target images and the deformations are smooth, illustrating the good registration performance of our proposed AVSM approach.