VoxelMorph: A Learning Framework for Deformable Medical Image Registration

Guha Balakrishnan, Amy Zhao, Mert R. Sabuncu, John Guttag, and Adrian V. Dalca

***Abstract*—We present VoxelMorph, a fast learning-based framework for deformable, pairwise medical image registration. Traditional registration methods optimize an objective function for each pair of images, which can be time-consuming for large datasets or rich deformation models. In contrast to this approach, and building on recent learning-based methods, we formulate registration as a function that maps an input image pair to a deformation field that aligns these images. We parameterize the function via a convolutional neural network (CNN), and optimize the parameters of the neural network on a set of images. Given a new pair of scans, VoxelMorph rapidly computes a deformation field by directly evaluating the function. In this work, we explore two different training strategies. In the first (unsupervised) setting, we train the model to maximize standard image matching objective functions that are based on the image intensities. In the second setting, we leverage auxiliary segmentations available in the training data. We demonstrate that the unsupervised model's accuracy is comparable to state-of-the-art methods, while operating orders of magnitude faster. We also show that VoxelMorph trained with auxiliary data improves registration accuracy at test time, and evaluate the effect of training set size on registration. Our method promises to speed up medical image analysis and processing pipelines, while facilitating novel directions in learning-based registration and its applications. Our code is freely available at http://voxelmorph.csail.mit.edu.**

***Index Terms*—registration, machine learning, convolutional neural networks**

I. Introduction

DEFORMABLE registration is a fundamental task in a variety of medical imaging studies, and has been a topic of active research for decades. In deformable registration, a dense, non-linear correspondence is established between a pair of images, such as 3D magnetic resonance (MR) brain scans. Traditional registration methods solve an optimization problem for each volume pair by aligning voxels with similar appearance while enforcing constraints on the registration mapping. Unfortunately, solving a pairwise optimization can be computationally intensive, and therefore slow in practice. For example, state-of-the-art algorithms running on the CPU can require tens of minutes to hours to register a pair of scans with high accuracy [1]–[3]. Recent GPU implementations have reduced this runtime to just minutes, but require a GPU for each registration [4].

We present a novel registration method that learns a parametrized registration *function* from a collection of volumes. We implement the function using a convolutional neural network (CNN), that takes two n-D input volumes and outputs a mapping of all voxels of one volume to another volume. The parameters of the network, i.e. the convolutional kernel weights, can be optimized using only a training set of volumes from the dataset of interest. The procedure learns a common representation that enables alignment of a new pair of volumes from the same distribution. In essence, we replace a costly optimization solved for each test image pair with one global function optimization during a training phase. Registration of a *new* test scan pair is achieved by simply evaluating the learned function on the given volumes, resulting in rapid registration, even on a CPU. We implement our method as a general purpose framework, VoxelMorph, available at http://voxelmorph.csail.mit.edu[1].

In the learning-based framework of VoxelMorph, we are free to adopt any differentiable objective function, and in this paper we present two possible choices. The first approach, which we refer to as unsupervised[2], uses only the input volume pair and the registration field computed by the model. Similar to traditional image registration algorithms, this loss function quantifies the dissimilarity between the intensities of the two images and the spatial regularity of the deformation. The second approach also leverages anatomical segmentations available at training time for a subset of the data, to learn network parameters.

Throughout this study, we use the example of registering 3D MR brain scans. However, our method is broadly applicable to other registration tasks, both within and beyond the medical imaging domain. We evaluate our work on a multi-study dataset of over 3,500 scans containing images of healthy and diseased brains from a variety of age groups. Our unsupervised model achieves comparable accuracy to state-of-the-art registration, while taking orders-of-magnitude less time. Registration with VoxelMorph requires less than a minute using a CPU and under a second on a GPU, in contrast to the state-of-the-art baselines which take tens of minutes to over two hours on a CPU.

This paper extends a preliminary version of the work presented at the 2018 International Conference on Computer Vision and Pattern Recognition [6]. We build on that work

Guha Balakrishnan, Amy Zhao and John Guttag are with the Computer Science and Artificial Intelligence Lab, MIT

Mert Sabuncu is with the the School of Electrical and Computer Engineering, and Meinig School of Biomedical Engineering, Cornell University.

Adrian V. Dalca is with the Computer Science and Artificial Intelligence Lab, MIT and also Martinos Center for Biomedical Imaging, MGH, HMS.

[1]We implement VoxelMorph as a flexible framework that includes the methods proposed in this manuscript, as well as extensions that are beyond the scope of this work [5]

[2]We use the term *unsupervised* to underscore the fact that VoxelMorph is a learning method (with images as input and deformations as output) that requires no deformation fields during training. Alternatively, such methods have also been termed *self-supervised*, to highlight the lack of supervision, or *end-to-end*, to highlight that no external computation is necessary as part of a pipeline (such as computing 'true' deformation fields).

by expanding analyses, and introducing an auxiliary learning model that can use anatomical segmentations during training to improve registration on new test image pairs for which segmentation maps are not available. We focus on providing a thorough analysis of the behavior of the VoxelMorph algorithm using two loss functions and a variety of settings, as follows. We test the unsupervised approach on more datasets and both atlas-based and subject-to-subject registration. We then explore cases where different types and numbers of anatomical region segmentations are available during training as auxiliary information, and evaluate the effect on registration of test data where segmentations are not available. We present an empirical analysis quantifying the effect of training set size on accuracy, and show how instance-specific optimization can improve results. Finally, we perform sensitivity analyses with respect to the hyperparameter choices, and discuss an interpretation of our model as amortized optimization.

The paper is organized as follows. Section 2 introduces medical image registration and Section 3 describes related work. Section 4 presents our methods. Section 5 presents experimental results on MRI data. We discuss insights of the results and conclude in Section 6.

II. Background

In the traditional volume registration formulation, one (moving or source) volume is warped to align with a second (fixed or target) volume. Fig. 1 shows sample 2D coronal slices taken from 3D MRI volumes, with boundaries of several anatomical structures outlined. There is significant variability across subjects, caused by natural anatomical brain variations and differences in health state. Deformable registration enables comparison of structures between scans. Such analyses are useful for understanding variability across populations or the evolution of brain anatomy over time for individuals with disease. Deformable registration strategies often involve two steps: an initial affine transformation for global alignment, followed by a much slower deformable transformation with more degrees of freedom. We concentrate on the latter step, in which we compute a dense, nonlinear correspondence for all voxels.

Most existing deformable registration algorithms iteratively optimize a transformation based on an energy function [7]. Let f and m denote the fixed and moving images, respectively, and let $\boldsymbol{\phi}$ be the registration field that maps coordinates of f to coordinates of m. The optimization problem can be written as:

$$\widehat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} \mathcal{L}(f, m, \boldsymbol{\phi}) \tag{1}$$

$$= \arg\min_{\boldsymbol{\phi}} \mathcal{L}_{sim}(f, m \circ \boldsymbol{\phi}) + \lambda \mathcal{L}_{smooth}(\boldsymbol{\phi}), \tag{2}$$

where $m \circ \boldsymbol{\phi}$ represents m warped by $\boldsymbol{\phi}$, function $\mathcal{L}_{sim}(\cdot, \cdot)$ measures image similarity between its two inputs, $\mathcal{L}_{smooth}(\cdot)$ imposes regularization, and λ is the regularization trade-off parameter.

There are several common formulations for $\boldsymbol{\phi}$, $\mathcal{L}_{sim}$ and $\mathcal{L}_{smooth}$. Often, $\boldsymbol{\phi}$ is characterized by a displacement vector field $\mathbf{u}$ specifying the vector offset from f to m for each voxel: $\boldsymbol{\phi} = Id + \mathbf{u}$, where Id is the identity transform [8].

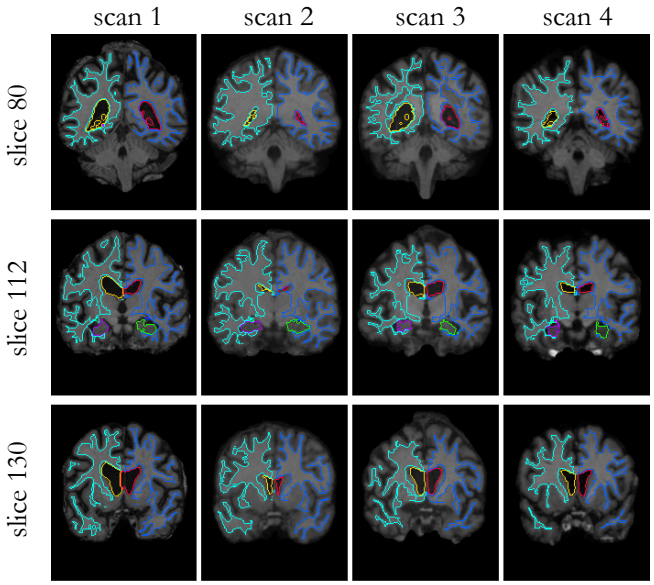

Fig. 1: Example coronal slices from the MRI brain dataset, after affine alignment. Each column is a different scan (subject) and each row is a different coronal slice. Some anatomical regions are outlined using different colors: L/R white matter in light/dark blue, L/R ventricles in yellow/red, and L/R hippocampi in purple/green. There are significant structural differences across scans, necessitating a deformable registration step to analyze inter-scan variations.

Diffeomorphic transforms model $\boldsymbol{\phi}$ through the integral of a velocity vector field, preserving topology and maintaining invertibility on the transformation [9]. Common metrics used for $\mathcal{L}_{sim}$ include intensity mean squared error, mutual information [10], and cross-correlation [11]. The latter two are particularly useful when volumes have varying intensity distributions and contrasts. $\mathcal{L}_{smooth}$ enforces a spatially smooth deformation, often modeled as a function of the spatial gradients of $\mathbf{u}$.

Traditional algorithms optimize (1) for each volume pair. This is expensive when registering many volumes, for example as part of population-wide analyses. In contrast, we assume that a field can be computed by a parameterized function of the data. We optimize the function parameters by minimizing the expected energy of the form of (1) over a dataset of volume pairs. Essentially, we replace pair-specific optimization of the deformation field by global optimization of the shared parameters, which in other domains has been referred to as amortization [12]–[15]. Once the global function is estimated, a field can be produced by evaluating the function on a given volume pair. In this paper, we use a displacement-based vector field representation, and focus on various aspects of the learning framework and its advantages. However, we recently demonstrated that velocity-based representations are also possible in a VoxelMorph-like framework, also included in our codebase [5].

III. Related Work

A. Medical Image Registration (Non-learning-based)

There is extensive work in 3D medical image registration [8], [9], [11], [16]–[21]. Several studies optimize within the space of displacement vector fields. These include elastic-type models [8], [22], [23], statistical parametric mapping [24], free-form deformations with b-splines [25], discrete methods [17], [18] and Demons [19], [26]. Diffeomorphic transforms, which are topology-preserving, have shown remarkable success in various computational anatomy studies. Popular formulations include Large Diffeomorphic Distance Metric Mapping (LDDMM) [9], [21], [27]–[32], DARTEL [16], diffeomorphic demons [33], and standard symmetric normalization (SyN) [11]. All of these non-learning-based approaches optimize an energy function for each image pair, resulting in slow registration. Recent GPU-based algorithms build on these concepts to reduce algorithm runtime to several minutes, but require a GPU to be available for each registration [4], [34].

B. Medical Image Registration (Learning-based)

There are several recent papers proposing neural networks to learn a function for medical image registration. Most of these rely on ground truth warp fields [35]–[39], which are either obtained by simulating deformations and deformed images, or running classical registration methods on pairs of scans. Some also use image similarity to help guide the registration [35]. While supervised methods present a promising direction, ground truth warp fields derived via conventional registration tools as ground truth can be cumbersome to acquire and can restrict the type of deformations that are learned. In contrast, VoxelMorph is unsupervised, and is also capable of leveraging auxiliary information such as segmentations during training if those are available.

Two recent papers [40], [41], were the first to present unsupervised learning based image registration methods. Both propose a neural network consisting of a CNN and spatial transformation function [42] that warps images to one another. However, these two initial methods are only demonstrated on limited subsets of volumes, such as 3D subregions [41] or 2D slices [40], and support only small transformations [40].

A recent method has proposed a segmentation driven cost function to be used in registering different imaging modalities – T2w MRI and 3D ultrasound – within the same subject [43], [44]. The authors demonstrate that a loss functions based solely on segmentation maps can lead to an accurate within-subject cross-modality registration network. Parallel to this work, in one of our experiments, we demonstrate the use of segmentation maps during training in subject-to-atlas registration. We provide an analysis of the effect of different anatomical label availability on overall registration quality, and evaluate how a combination of segmentation and image based losses behaves in various scenarios. We find that a segmentation-based loss can be helpful, for example if the input segment labels are the same as those we evaluate on (consistent with [43], and [44]). We also show that the image-based and smoothness losses are still necessary, especially when we evaluate registration accuracy on labels not observed during training, and to encourage deformation regularity.

C. 2D Image Alignment

Optical flow estimation is a related registration problem for 2D images. Optical flow algorithms return a dense displacement vector field depicting small displacements between a pair of 2D images. Traditional optical flow approaches typically solve an optimization problem similar to (1) using variational methods [45]–[47]. Extensions that better handle large displacements or dramatic changes in appearance include feature-based matching [48], [49] and nearest neighbor fields [50].

In recent years, several learning-based approaches to optical flow estimation using neural networks have been proposed [51]–[56]. These algorithms take a pair of images as input, and use a convolutional neural network to learn image features that capture the concept of optical flow from data. Several of these works require supervision in the form of ground truth flow fields [52], [53], [55], [56], while we build on a few that use an unsupervised objective [51], [54]. The spatial transform layer enables neural networks to perform both global parametric 2D image alignment [42] and dense spatial transformations [54], [61], [62] without requiring supervised labels. An alternative approach to dense estimation is to use CNNs to match image patches [57]–[60]. These methods require exhaustive matching of patches, resulting in slow runtime.

We build on these ideas and extend the spatial transformer to achieve n-D volume registration, and further show how leveraging image segmentations during training can improve registration accuracy at test time.

IV. Method

Let f, m be two image volumes defined over an n-D spatial domain $\Omega \subset \mathbb{R}^n$. For the rest of this paper, we focus on the case $n = 3$ but our method and implementation are dimension independent. For simplicity we assume that f and m contain single-channel, grayscale data. We also assume that f and m are affinely aligned as a preprocessing step, so that the only source of misalignment between the volumes is nonlinear. Many packages are available for rapid affine alignment.

We model a function $g_\theta(f, m) = \mathbf{u}$ using a convolutional neural network (CNN), where θ are network parameters, the kernels of the convolutional layers. The displacement field $\mathbf{u}$ between f and m is in practice stored in a $n+1$-dimensional image. That is, for each voxel $\mathbf{p} \in \Omega$, $\mathbf{u}(\mathbf{p})$ is a displacement such that $f(\mathbf{p})$ and $[m \circ \boldsymbol{\phi}](\mathbf{p})$ correspond to similar anatomical locations, where the map $\boldsymbol{\phi} = Id + \mathbf{u}$ is formed using an identity transform and $\mathbf{u}$.

Fig. 2 presents an overview of our method. The network takes f and m as input, and computes $\boldsymbol{\phi}$ using a set of parameters θ. We warp m to $m \circ \boldsymbol{\phi}$ using a spatial transformation function, enabling evaluation of the similarity of $m \circ \boldsymbol{\phi}$ and f. Given unseen images f and m during test time, we obtain a registration field by evaluating $g_\theta(f, m)$.

We use (single-element) stochastic gradient descent to find optimal parameters $\hat{\theta}$ by minimizing an expected loss function using a training dataset. We propose two unsupervised loss

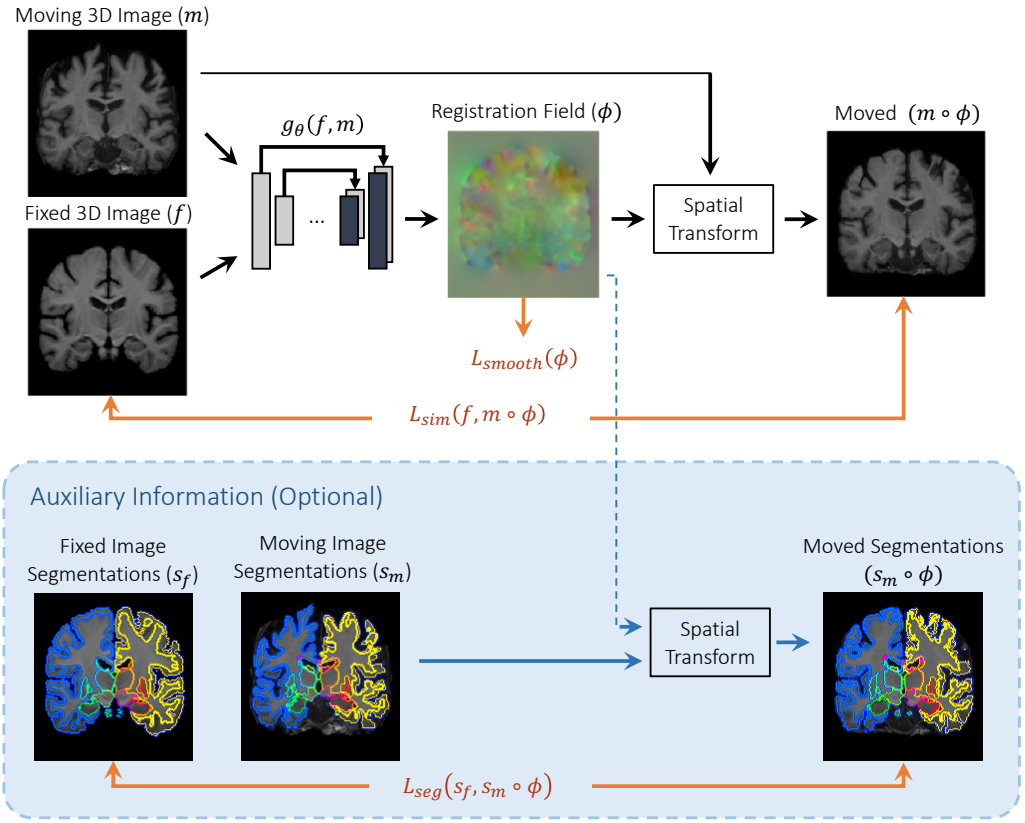

Fig. 2: Overview of the method. We learn parameters θ for a function $g_\theta(\cdot,\cdot)$, and register 3D volume m to a second, fixed volume f. During training, we warp m with ϕ using a spatial transformer function. Optionally, auxiliary information such as anatomical segmentations $\mathbf{s_f}, \mathbf{s_m}$ can be leveraged during training (blue box).

functions in this work. The first captures image similarity and field smoothness, while the second also leverages anatomical segmentations. We describe our CNN architecture and the two loss functions in detail in the next sections.

A. VoxelMorph CNN Architecture

In this section we describe the particular architecture used in our experiments, but emphasize that a wide range of architectures may work similarly well and that the exact architecture is not our focus. The parametrization of $g_\theta(\cdot,\cdot)$ is based on a convolutional neural network architecture similar to UNet [63], [64], which consists of encoder and decoder sections with skip connections.

Fig. 3 depicts the network used in VoxelMorph, which takes a single input formed by concatenating m and f into a 2-channel 3D image. In our experiments, the input is of size $160 \times 192 \times 224 \times 2$, but the framework is not limited by a particular size. We apply 3D convolutions in both the encoder and decoder stages using a kernel size of 3, and a stride of 2. Each convolution is followed by a LeakyReLU layer with parameter 0.2. The convolutional layers capture hierarchical features of the input image pair, used to estimate ϕ. In the encoder, we use strided convolutions to reduce the spatial dimensions in half at each layer. Successive layers of the encoder therefore operate over coarser representations of the input, similar to the image pyramid used in traditional image registration work.

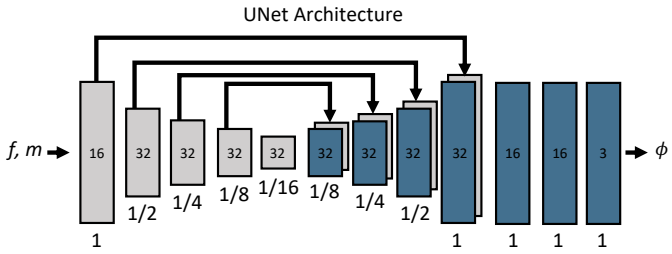

Fig. 3: Convolutional UNet architecture implementing $g_\theta(f,m)$. Each rectangle represents a 3D volume, generated from the preceding volume using a 3D convolutional network layer. The spatial resolution of each volume with respect to the input volume is printed underneath. In the decoder, we use several 32-filter convolutions, each followed by an upsampling layer, to bring the volume back to full resolution. Arrows represent skip connections, which concatenate encoder and decoder features. The full-resolution volume is further refined using several convolutions.

In the decoding stage, we alternate between upsampling, convolutions and concatenating skip connections that propagate features learned during the encoding stages directly to layers generating the registration. Successive layers of the decoder operate on finer spatial scales, enabling precise

anatomical alignment. The receptive fields of the convolutional kernels of the smallest layer should be at least as large as the maximum expected displacement between corresponding voxels in f and m. In our architecture, the smallest layer applies convolutions over a volume $(1/16)^3$ of the size of the input images.

B. Spatial Transformation Function

The proposed method learns optimal parameter values in part by minimizing differences between $m \circ \boldsymbol{\phi}$ and f. In order to use standard gradient-based methods, we construct a differentiable operation based on spatial transformer networks [42] to compute $m \circ \boldsymbol{\phi}$.

For each voxel $\mathbf{p}$, we compute a (subpixel) voxel location $\mathbf{p}' = \mathbf{p} + \mathbf{u}(\mathbf{p})$ in m. Because image values are only defined at integer locations, we linearly interpolate the values at the eight neighboring voxels:

$$m \circ \boldsymbol{\phi}(\mathbf{p}) = \sum_{\mathbf{q} \in \mathcal{Z}(\mathbf{p}')} m(\mathbf{q}) \prod_{d \in \{x,y,z\}} (1 - |\mathbf{p}'_d - \mathbf{q}_d|), \tag{3}$$

where $\mathcal{Z}(\mathbf{p}')$ are the voxel neighbors of $\mathbf{p}'$, and d iterates over dimensions of Ω. Because we can compute gradients or subgradients,[3] we can backpropagate errors during optimization.

C. Loss Functions

In this section, we propose two loss functions: an unsupervised loss $\mathcal{L}_{us}$ that evaluates the model using only the input volumes and generated registration field, and an auxiliary loss $\mathcal{L}_a$ that also leverages anatomical segmentations at training time.

1) Unsupervised Loss Function: The unsupervised loss $\mathcal{L}_{us}(\cdot,\cdot,\cdot)$ consists of two components: $\mathcal{L}_{sim}$ that penalizes differences in appearance, and $\mathcal{L}_{smooth}$ that penalizes local spatial variations in $\boldsymbol{\phi}$:

$$\mathcal{L}_{us}(f, m, \boldsymbol{\phi}) = \mathcal{L}_{sim}(f, m \circ \boldsymbol{\phi}) + \lambda \mathcal{L}_{smooth}(\boldsymbol{\phi}), \tag{4}$$

where λ is a regularization parameter. We experimented with two often-used functions for $\mathcal{L}_{sim}$. The first is the mean squared voxelwise difference, applicable when f and m have similar image intensity distributions and local contrast:

$$MSE(f, m \circ \boldsymbol{\phi}) = \frac{1}{|\Omega|} \sum_{p \in \Omega} \left[f(\mathbf{p}) - [m \circ \boldsymbol{\phi}](\mathbf{p}) \right]^2. \tag{5}$$

The second is the local cross-correlation of f and $m \circ \boldsymbol{\phi}$, which is more robust to intensity variations found across scans and datasets [11]. Let $\hat{f}(\mathbf{p})$ and $[\hat{m} \circ \boldsymbol{\phi}](\mathbf{p})$ denote images with local mean intensities subtracted out: $\hat{f}(\mathbf{p}) = f(\mathbf{p}) - \frac{1}{n^3} \sum_{\mathbf{p}_i} f(\mathbf{p}_i)$, where $\mathbf{p}_i$ iterates over a n^3 volume around $\mathbf{p}$, with $n = 9$ in our experiments. The local cross-correlation of f and $m \circ \boldsymbol{\phi}$ is written as:

$$CC(f, m \circ \boldsymbol{\phi}) = \sum_{\mathbf{p} \in \Omega} \frac{\left(\sum_{\mathbf{p}_i} (f(\mathbf{p}_i) - \hat{f}(\mathbf{p}))([m \circ \boldsymbol{\phi}](\mathbf{p}_i) - [\hat{m} \circ \boldsymbol{\phi}](\mathbf{p})) \right)^2}{\left(\sum_{\mathbf{p}_i} (f(\mathbf{p}_i) - \hat{f}(\mathbf{p}))^2 \right) \left(\sum_{\mathbf{p}_i} ([m \circ \boldsymbol{\phi}](\mathbf{p}_i) - [\hat{m} \circ \boldsymbol{\phi}](\mathbf{p}))^2 \right)}. \tag{6}$$

[3]The absolute value is implemented with a subgradient of 0 at 0.

A higher CC indicates a better alignment, yielding the loss function: $\mathcal{L}_{sim}(f, m, \boldsymbol{\phi}) = -CC(f, m \circ \boldsymbol{\phi})$.

Minimizing $\mathcal{L}_{sim}$ will encourage $m \circ \boldsymbol{\phi}$ to approximate f, but may generate a non-smooth $\boldsymbol{\phi}$ that is not physically realistic. We encourage a smooth displacement field $\boldsymbol{\phi}$ using a diffusion regularizer on the spatial gradients of displacement $\mathbf{u}$:

$$\mathcal{L}_{smooth}(\boldsymbol{\phi}) = \sum_{\mathbf{p} \in \Omega} \|\nabla \mathbf{u}(\mathbf{p})\|^2, \tag{7}$$

and approximate spatial gradients using differences between neighboring voxels. Specifically, for $\nabla \mathbf{u}(\mathbf{p}) = \left(\frac{\partial \mathbf{u}(\mathbf{p})}{\partial x}, \frac{\partial \mathbf{u}(\mathbf{p})}{\partial y}, \frac{\partial \mathbf{u}(\mathbf{p})}{\partial z} \right)$, we approximate $\frac{\partial \mathbf{u}(\mathbf{p})}{\partial x} \approx \mathbf{u}((p_x + 1, p_y, p_z)) - \mathbf{u}((p_x, p_y, p_z))$, and use similar approximations for $\frac{\partial \mathbf{u}(\mathbf{p})}{\partial y}$ and $\frac{\partial \mathbf{u}(\mathbf{p})}{\partial z}$.

2) Auxiliary Data Loss Function: Here, we describe how VoxelMorph can leverage auxiliary information available during training but not during testing. Anatomical segmentation maps are sometimes available during training, and can be annotated by human experts or automated algorithms. A segmentation map assigns each voxel to an anatomical structure. If a registration field $\boldsymbol{\phi}$ represents accurate anatomical correspondences, the regions in f and $m \circ \boldsymbol{\phi}$ corresponding to the same anatomical structure should overlap well.

Let $s_f^k, s_m^k \circ \boldsymbol{\phi}$ be the voxels of structure k for f and $m \circ \boldsymbol{\phi}$, respectively. We quantify the volume overlap for structure k using the Dice score [65]:

$$\text{Dice}(s_f^k, s_m^k \circ \boldsymbol{\phi}) = 2 \cdot \frac{|s_f^k \cap (s_m^k \circ \boldsymbol{\phi})|}{|s_f^k| + |s_m^k \circ \boldsymbol{\phi}|}. \tag{8}$$

A Dice score of 1 indicates that the anatomy matches perfectly, and a score of 0 indicates that there is no overlap. We define the segmentation loss $\mathcal{L}_{seg}$ over all structures $k \in [1, K]$ as:

$$\mathcal{L}_{seg}(s_f, s_m \circ \boldsymbol{\phi}) = -\frac{1}{K} \sum_{k=1}^{K} \text{Dice}(s_f^k, s_m^k \circ \boldsymbol{\phi}). \tag{9}$$

$\mathcal{L}_{seg}$ alone does not encourage smoothness and agreement of image appearance, which are essential to good registration. We therefore combine $\mathcal{L}_{seg}$ with (4) to obtain the objective:

$$\mathcal{L}_a(f, m, s_f, s_m, \boldsymbol{\phi}) = \mathcal{L}_{us}(f, m, \boldsymbol{\phi}) + \gamma \mathcal{L}_{seg}(s_f, s_m \circ \boldsymbol{\phi}), \tag{10}$$

where γ is a regularization parameter.

In our experiments, which use affinely aligned images, we demonstrate that loss (10) can lead to significant improvements. In general, and depending on the task, this loss can also be computed in a multiscale fashion as introduced in [43], depending on quality of the initial alignment.

Since anatomical labels are categorical, a naive implementation of linear interpolation to compute $s_m \circ \boldsymbol{\phi}$ is inappropriate, and a direct implementation of (8) might not be amenable to auto-differentiation frameworks. We design s_f and s_m to be image volumes with K channels, where each channel is a binary mask specifying the spatial domain of a particular structure. We compute $s_m \circ \boldsymbol{\phi}$ by spatially transforming each channel of s_m using linear interpolation. We then compute the numerator and denominator of (8) by multiplying and adding s_f and $s_m \circ \boldsymbol{\phi}$, respectively.

D. Amortized Optimization Interpretation

Our method substitutes the pair-specific optimization over the deformation field ϕ with a global optimization of function parameters θ for function $g_\theta(\cdot,\cdot)$. This process is sometimes referred to as amortized optimization [66]. Because the function $g_\theta(\cdot,\cdot)$ is tasked with estimating registration between any two images, the fact that parameters θ are shared globally acts as a natural regularization. We demonstrate this aspect in Section V-C (Regularization Analysis). In addition, the quality and generalizability of the deformations outputted by the function will depend on the data it is trained on. Indeed, the resulting deformation can be interpreted as simply an approximation or initialization to the optimal deformation ϕ^*, and the resulting difference is sometimes referred to as the amortization gap [15], [66]. If desired, this initial deformation field could be improved using any instance-specific optimization. In our experiments, we accomplish this by treating the resulting displacement $\mathbf{u}$ as model parameters, and fine-tuning the deformation for each particular scan independently using gradient descent. Essentially, this implements an auto-differentiation version of conventional registration, using VoxelMorph output as initialization. However, most often we find that the *initial* deformation, the VoxelMorph output, is already as accurate as state of the art results. We explore these aspects in experiments presented in Section V-D.

V. Experiments

We demonstrate our method on the task of brain MRI registration. We first (Section V-B) present a series of atlas-based registration experiments, in which we compute a registration field between an atlas, or reference volume, and each volume in our dataset. Atlas-based registration is a common formulation in population analysis, where inter-subject registration is a core problem. The atlas represents a reference, or average volume, and is usually constructed by jointly and repeatedly aligning a dataset of brain MR volumes and averaging them together [67]. We use an atlas computed using an external dataset [1], [68]. Each input volume pair consists of the atlas (image f) and a volume from the dataset (image m). Fig. 4 shows example image pairs using the same fixed atlas for all examples. In a second experiment (Section V-C), we perform hyper-parameter sensitivity analysis. In a third experiment (Section V-D), we study the effect of training set size on registration, and demonstrate instance-specific optimization. In the fourth experiment (Section V-E) we present results on a dataset that contains *manual* segmentations. In the next experiment (Section V-F), we train VoxelMorph using random pairs of training subjects as input, and test registration between pairs of unseen test subjects. Finally (Section V-G), we present an empirical analysis of registration with auxiliary segmentation data. All figures that depict brains in this paper show 2D slices, but all registration is done in 3D.

A. Experimental Setup

1) Dataset: We use a large-scale, multi-site, multi-study dataset of 3731 T1–weighted brain MRI scans from eight publicly available datasets: OASIS [69], ABIDE [70], ADHD200 [71], MCIC [72], PPMI [73], HABS [74], Harvard GSP [75], and the FreeSurfer Buckner40 [1]. Acquisition details, subject age ranges and health conditions are different for each dataset. All scans were resampled to a $256\times256\times256$ grid with 1mm isotropic voxels. We carry out standard pre-processing steps, including affine spatial normalization and brain extraction for each scan using FreeSurfer [1], and crop the resulting images to $160 \times 192 \times 224$. All MRIs were anatomically segmented with FreeSurfer, and we applied quality control using visual inspection to catch gross errors in segmentation results and affine alignment. We include all anatomical structures that are at least 100 voxels in volume for all test subjects, resulting in 30 structures. We use the resulting segmentation maps in evaluating our registration as described below. We split our dataset into 3231, 250, and 250 volumes for train, validation, and test sets respectively, although we highlight that we do not use any supervised information at any stage. In addition, the Buckner40 dataset is only used for testing, using manual segmentations.

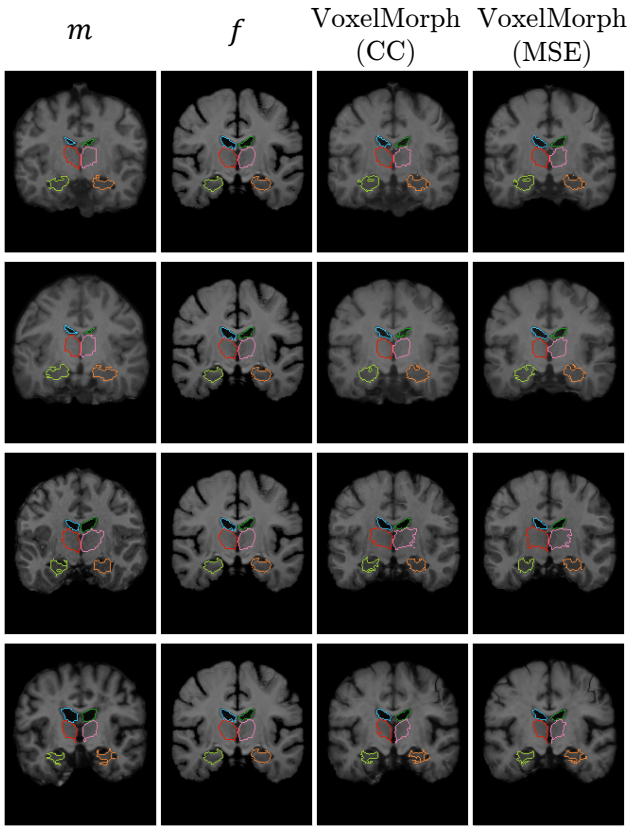

Fig. 4: Example MR coronal slices extracted from input pairs (columns 1-2), and resulting $m \circ \phi$ for VoxelMorph using different loss functions. We overlaid boundaries of a few structures: ventricles (blue/dark green), thalami (red/pink), and hippocampi (light green/orange). A good registration will cause structures in $m \circ \phi$ to look similar to structures in f. Our models are able to handle various changes in shape of structures, including expansion/shrinkage of the ventricles in rows 2 and 3, and stretching of the hippocampi in row 4.

2) Evaluation Metrics: Obtaining dense *ground truth* registration for these data is not well-defined since many reg-

Method	Dice	GPU sec	CPU sec	$\lvert J_\phi\rvert \leq 0$	% of $\lvert J_\phi\rvert \leq 0$
Affine only	0.584 (0.157)	0	0	0	0
ANTs SyN (CC)	0.749 (0.136)	-	9059 (2023)	9662 (6258)	0.140 (0.091)
NiftyReg (CC)	0.755 (0.143)	-	2347 (202)	41251 (14336)	0.600 (0.208)
VoxelMorph (CC)	0.753 (0.145)	0.45 (0.01)	57 (1)	19077 (5928)	0.366 (0.114)
VoxelMorph (MSE)	0.752 (0.140)	0.45 (0.01)	57 (1)	9606 (4516)	0.184 (0.087)

TABLE I: Average Dice scores and runtime results for affine alignment, ANTs, NiftyReg and VoxelMorph for the first experiment. Standard deviations across structures and subjects are in parentheses. The average Dice score is computed over all structures and subjects. Timing is computed after preprocessing. Our networks yield comparable results to ANTs and NiftyReg in Dice score, while operating orders of magnitude faster during testing. We also show the number and percentage of voxels with a non-positive Jacobian determinant for each method, for our volumes with 5.2 million voxels within the brain. All methods exhibit less than 1 percent such voxels.

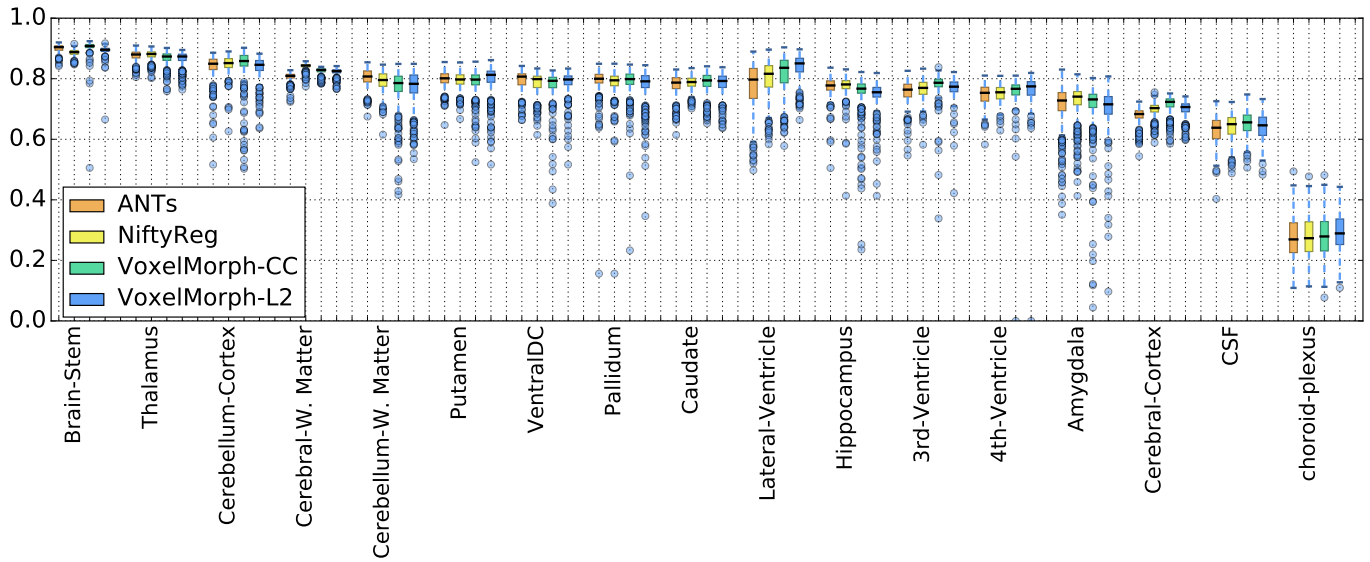

Fig. 5: Boxplots of Dice scores for various anatomical structures for ANTs, NiftyReg, and VoxelMorph results for the first (unsupervised) experiment. We average Dice scores of the left and right brain hemispheres into one score for this visualization. Structures are ordered by average ANTs Dice score.

istration fields can yield similar looking warped images. We first evaluate our method using volume overlap of anatomical segmentations. If a registration field ϕ represents accurate correspondences, the regions in f and $m \circ \phi$ corresponding to the same anatomical structure should overlap well (see Fig. 4 for examples). We quantify the volume overlap between structures using the Dice score (8). We also evaluate the regularity of the deformation fields. Specifically, the Jacobian matrix $J_\phi(\mathbf{p}) = \nabla\phi(\mathbf{p}) \in \mathcal{R}^{3\times 3}$ captures the local properties of ϕ around voxel $\mathbf{p}$. We count all non-background voxels for which $|J_\phi(\mathbf{p})| \leq 0$, where the deformation is not diffeomorphic [16].

3) Baseline Methods: We use Symmetric Normalization (SyN) [11], the top-performing registration algorithm in a comparative study [2] as a first baseline. We use the SyN implementation in the publicly available Advanced Normalization Tools (ANTs) software package [3], with a cross-correlation similarity measure. Throughout our work with medical images, we found the default ANTs smoothness parameters to be sub-optimal for applying ANTs to our data. We obtained improved parameters using a wide parameter sweep across multiple datasets, and use those in these experiments. Specifically, we use SyN step size of 0.25, Gaussian parameters (9, 0.2), at three scales with at most 201 iterations each. We also use the NiftyReg package, as a second baseline. Unfortunately, a GPU implementation is not currently available, and instead we build a multi-threaded CPU version[4]. We searched through various parameter settings to obtain improved parameters, and use the CC cost function, grid spacing of 5, and 500 iterations.

4) VoxelMorph Implementation: We implemented our method using Keras [76] with a Tensorflow backend [77]. We extended the 2D linear interpolation spatial transformer layer to n-D, and here use $n = 3$. We use the ADAM optimizer [78] with a learning rate of 10^{-4}. While our implementation allows for mini-batch stochastic gradient descent, in our experiments each training batch consists of one pair of volumes. Our implementation includes a default of 150,000 iterations. Our code and model parameters are available online at http://voxelmorph.csail.mit.edu.

B. Atlas-based Registration

[4]We used the latest source code, updated March, 2018 (tree [4e4525]).

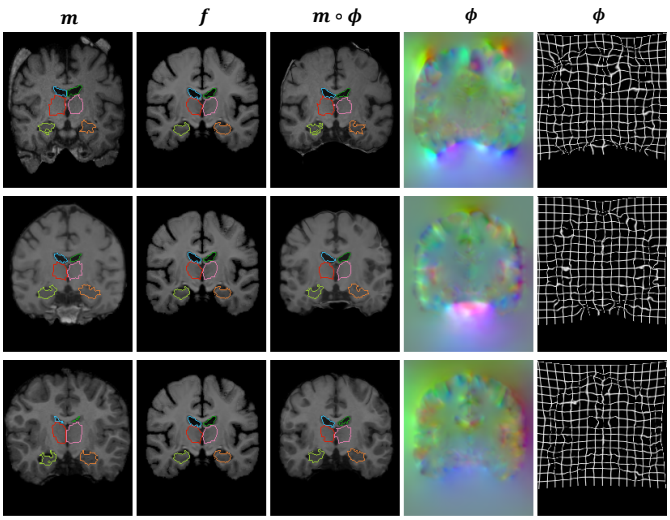

Fig. 6: Example deformation fields ϕ (columns 4-5) extracted by registering the moving image (column 1) to the fixed image (column 2) in the unsupervised experiment (Section V-B) . The warped volume $m \circ \phi$ is shown in column 3. Displacement in each spatial dimension is mapped to each of the RGB color channels in column 4. The deformation fields produced by VoxelMorph (MSE) are smooth within the brain, even when registering moving images that are significantly different from the fixed image.

In this experiment, we train VoxelMorph for atlas-based registration. We train separate VoxelMorph networks with different λ regularization parameters. We then select the network that optimizes Dice score on our validation set, and report results on our test set.

Table I presents average Dice scores computed for all subjects and structures for baselines of only global affine alignment, ANTs, and NiftyReg, as well as VoxelMorph with different losses. VoxelMorph variants perform comparably to ANTs and NiftyReg in terms of Dice[5], and are significantly better than affine alignment. Example visual results of the warped images from our algorithms are shown in Figs. 4 and 6. VoxelMorph is able to handle significant shape changes for various structures.

Fig. 5 presents the Dice scores for each structure as a boxplot. For ease of visualization, we average Dice scores of the same structures from the two hemispheres into one score, e.g., the left and right hippocampi scores are averaged. The VoxelMorph models achieve comparable Dice measures to ANTs and NiftyReg for all structures, performing slightly better on some structures such as the lateral ventricles, and worse on others such as the hippocampi.

Table I includes a count of voxels for which the Jacobian determinant is non-positive. We find that all methods result in deformations with small islands of such voxels, but are diffeomorphic at the vast majority of voxels (99.4% - 99.9%). Figs. 6 and Fig. 11 in the supplemental material illustrate several example VoxelMorph deformation fields. VoxelMorph has no explicit constraint for diffeomorphic deformations, but in this setting the smoothness loss leads to generally smooth and well-behaved results. ANTs and NiftyReg include implementations that can enforce or strongly encourage diffeomorphic deformations, but during our parameter search these negatively affected runtime or results. In this work, we ran the baseline implementations with configurations that yielded the best Dice scores, which also turned out to produce good deformation regularity.

1) Runtime: Table I presents runtime results using an Intel Xeon (E5-2680) CPU, and a NVIDIA TitanX GPU. We report the elapsed time for computations following the affine alignment preprocessing step, which all of the presented methods share, and requires just a few minutes even on a CPU. ANTs requires two or more hours on the CPU, while NiftyReg requires roughly 39 minutes for the given setting. ANTs runtimes vary widely, as its convergence depends on the difficulty of the alignment task. Registering two images with VoxelMorph is, on average, 150 times faster on the CPU compared to ANTs, and 40 times faster than NiftyReg. When using the GPU, VoxelMorph computes a registration in under a second. To our knowledge, there is no publicly available

[5]Both VoxelMorph variants are different from ANTs with paired t-test p-values of 0.003 and 0.008 and with slightly higher Dice values. There is no difference between VoxelMorph (CC) and NiftyReg (p-value of 0.21), and no significant difference between VoxelMorph (CC) and VoxelMorph (MSE) (p-value of 0.09)

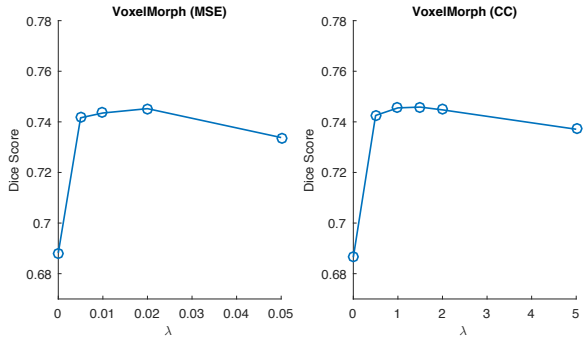

Fig. 7: Dice score of validation data for VoxelMorph with varied regularization parameter λ.

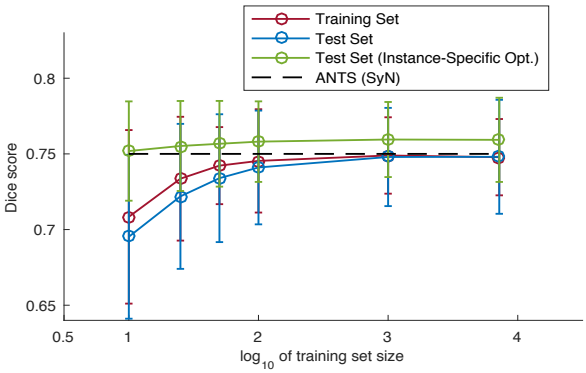

Fig. 8: Effect of training set size on accuracy. Also shown are results of instance-specific optimization of deformations, after these are initialized with VoxelMorph outputs using the optimal global parameters resulting from the training phase.

Method	Dice
Affine only	0.608 (0.175)
ANTs SyN (CC)	0.776 (0.130)
NiftyReg (CC)	0.776 (0.132)
VoxelMorph (MSE)	0.766 (0.133)
VoxelMorph (MSE) inst.	0.776 (0.132)
VoxelMorph (CC)	0.774 (0.133)
VoxelMorph (CC) inst.	0.786 (0.132)

TABLE II: Results for manual annotation experiment. We show affine, ANTs, NiftyReg, and VoxelMorph, where "inst." indicates additional instance-specific optimization, as described in *Section V-D*. The average Dice score is computed over all structures and subjects, with standard deviations across structures and subjects in parentheses.

ANTs implementation for GPUs. It is likely that the SyN algorithm would benefit from a GPU implementation, but the main advantage of VoxelMorph comes from not requiring an optimization on each test pair, as can be seen in the CPU comparison. Unfortunately, the NiftyReg GPU version is unavailable in the current source code on all available repository history.

C. Regularization Analysis

Fig. 7 shows average Dice scores for the validation set for different values of the smoothness regularization parameter λ. The results vary smoothly over a large range of λ values, illustrating that our model is robust to choice of λ. Interestingly, even setting $\lambda = 0$, which enforces no explicit regularization on registration, results in a significant improvement over affine registration. This is likely because the optimal network parameters θ need to register all pairs in the training set well, yielding an implicit dataset regularization for the function $g_\theta(\cdot, \cdot)$.

D. Training Set Size and Instance-Specific Optimization

We evaluate the effect of training set size on accuracy, and the relationship between amortized and instance-specific optimization. Because MSE and CC performed similarly for atlas-based registration, in this section we use MSE. We train VoxelMorph on subsets of different sizes from our training dataset, and report Dice scores on: (1) the training subset, (2) the held out test set, and (3) the test set when each deformation is further individually optimized for each test image pair. We perform (3) by fine-tuning the displacements $\mathbf{u}$ obtained from VoxelMorph using gradient descent for 100 iterations on each test pair, which took 23.7 ± 0.4 seconds on the GPU or 628.0 ± 4.2 seconds on a single-threaded CPU.

Fig. 8 presents our results. A small training set size of 10 scans results in slightly lower train and test Dice scores compared to larger training set sizes. However, there is no significant difference in Dice scores when training with 100 scans or the full dataset. Further optimizing the VoxelMorph parameters on each test image pair results in better test Dice scores regardless of training set size, comparable to the state-of-the-art.

Method	Dice
Affine only	0.579 (0.173)
ANTs SyN (CC)	0.761 (0.117)
NiftyReg (CC)	0.772 (0.117)
VoxelMorph (MSE)	0.727 (0.146)
VoxelMorph x2 (MSE)	0.750 (0.058)
VoxelMorph x2 (MSE) inst.	0.764 (0.048)
VoxelMorph (CC)	0.737 (0.139)
VoxelMorph x2 (CC)	0.763 (0.049)
VoxelMorph x2 (CC) inst.	0.772 (0.119)

TABLE III: Results for subject-to-subject alignment using affine, ANTs, and VoxelMorph variants, where "x2" refers to a model where we doubled the number of features to account for the increased inherent variability of the task, and "inst." indicates additional instance-specific optimization.

E. Manual Anatomical Delineations

Since manual segmentations are not available for most datasets, the availability of FreeSurfer segmentations enabled the broad range of experiments above. In this experiment, we use VoxelMorph models already trained in *Section V-B* to test registration on the (unseen) Buckner40 dataset containing 39 scans. This dataset contains expert manual delineations of the same anatomical structures used in previous experiments, which we use here for evaluation. We also compute VoxelMorph with instance-specific optimization, as described in *Section V-D*. The Dice score results, shown in Table II, show that VoxelMorph using cross-correlation loss behaves comparably to ANTs and NiftyReg using the same cost function, consistent with the first experiment where we evaluated on FreeSurfer segmentations. VoxelMorph with instance-specific optimization further improves the results, similar to the previous experiment. On this dataset, results using VoxelMorph with MSE loss obtain slightly lower scores, but are improved by the instance-specific optimization procedure to be comparable to ANTs and NiftyReg.

F. Subject-to-Subject Registration

In this experiment, we train VoxelMorph for subject-to-subject registration. Since there is more variability in each registration, we double the number of features for each network layer. We also compute VoxelMorph with instance-specific optimization, as described in *Section V-D*. Table III presents average test Dice scores on 250 randomly selected test pairs for registration. Consistent with literature, we find that the normalized cross correlation loss leads to more robust results compared to using the MSE loss. VoxelMorph (with doubled feature counts) Dice scores are comparable with ANTs and slightly below NiftyReg, while results from VoxelMorph with instance-specific optimization are comparable to both baselines.

G. Registration with Auxiliary Data

In this section, we evaluate VoxelMorph when using segmentation maps during training with loss function (10). Because MSE and CC performed similarly for atlas-based registration, in this section we use MSE with $\lambda = 0.02$. We

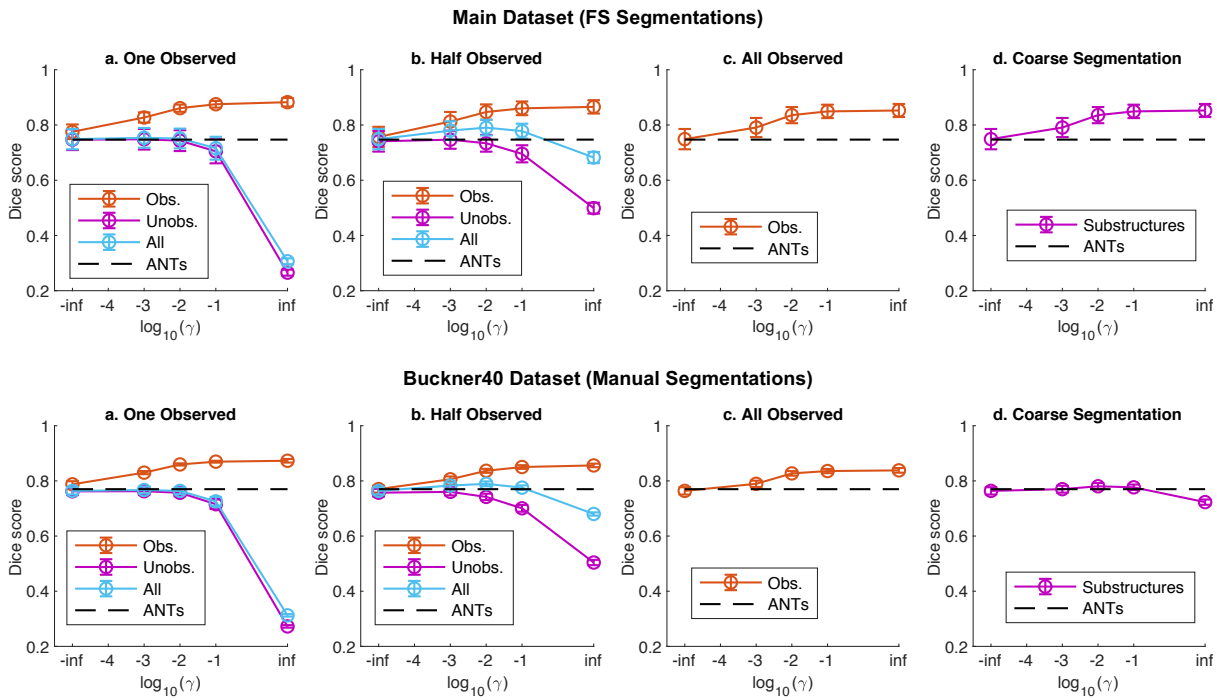

Fig. 9: Results on test scans when using auxiliary data during training. Top: testing on the FreeSurfer segmentation of the general test set. Bottom: testing the same models on the manual segmentation of the Buckner40 test set. We test having varying number of observed labels (a-c), and having coarser segmentation maps (d). Error bars indicate standard deviations across subjects. The leftmost datapoint in each graph for all labels, corresponding to $\gamma = 0$, indicates results of VoxelMorph without using auxiliary data (unsupervised). $\gamma = \infty$ is achieved by setting the image and smoothness terms to 0. We show Dice scores for results from ANTs with optimal parameters, which does not use segmentation maps, for comparison.

present an evaluation of our model in two practical scenarios: (1) when subsets of anatomical structure labels are available during training, and (2) when coarse segmentations labels are available during training. We use the same train/validation/test split as the previous experiments.

1) Training with a subset of anatomical labels: In many practical settings, it may be infeasible to obtain training segmentations for all structures. We therefore first consider the case where segmentations are available for only a subset of the 30 structures. We refer to structures present in segmentations as *observed*, and the rest as *unobserved*. We considered three scenarios, when: one, 15 (half), and 30 (all) structure segmentations are observed. The first two experiments essentially simulate different amounts of partially observed segmentations. For each experiment, we train separate models on different subsets of observed structures, as follows. For single structure segmentations, we manually selected four important structures for four folds (one for each fold) of the experiment: hippocampi, cerebral cortex, cerebral white matter, and ventricles. For the second experiment, we randomly selected 15 of the 30 structures, with a different selection for each of five folds. For each fold and each subset of observed labels, we use the segmentation maps at training, and show results on test pairs where segmentation maps are not used.

Fig. 9a-c shows Dice scores for both the observed and unobserved labels when sweeping γ in (10), the auxiliary regularization trade-off parameter. We train our models with FreeSurfer annotations, and show results on both the general test set using FreeSurfer annotations (top) and the Buckner40 test set with manual annotations (bottom). The extreme values $\gamma = 0$ (or $\log \gamma = -\infty$) and $\gamma = \infty$ serve as theoretical extremes, with $\gamma = 0$ corresponding to unsupervised VoxelMorph, and $\gamma = \infty$ corresponding to VoxelMorph trained *only* with auxiliary labels, without the smoothness and image matching objective terms.

In general, VoxelMorph with auxiliary data significantly outperforms (largest p-value $< 10^{-9}$ among the four settings) unsupervised VoxelMorph (equivalent to $\gamma = 0$ or $\log \gamma = -\infty$) and ANTs on observed structures in terms of Dice score. Dice score on observed labels generally increases with an increase in γ.

Interestingly, VoxelMorph (trained with auxiliary data) yields improved Dice scores for unobserved structures compared to the unsupervised variant for a range of γ values (see Fig. 9a-b), even though these segmentations were not explicitly observed during training. When *all* structures that we use during evaluation are observed during training, we find good Dice results at higher γ values (Fig 9c.). Registration accuracy for unobserved structures starts declining when γ is large, in the range $\log \gamma \in [-3, -2]$. This can be interpreted as the range where the model starts to over-fit to the observed structures - that is, it continues to improve the Dice score for observed structures while harming the registration accuracy for the other structures (Fig. 9c)

2) Training with coarse labels: We consider the scenario where only coarse labels are available, such as when all the

Setting	0	0.001	0.01	0.1	∞
one (count)	9606 (4471)	10435 (4543)	22998 (3171)	121546 (12203)	685811 (6878)
one (%)	0.18 (0.09)	0.20 (0.09)	0.44 (0.06)	2.33 (0.23)	13.14 (0.13)
half (count)	9606 (4471)	9470 (4008)	17886 (4919)	86319 (13851)	516384 (7210)
half (%)	0.18 (0.09)	0.18 (0.08)	0.34 (0.09)	1.65 (0.27)	9.90 (0.14)
all (count)	9606 (4471)	10824 (5029)	19226 (4471)	102295 (14366)	528552 (8720)
all (%)	0.18 (0.09)	0.21 (0.10)	0.37 (0.09)	1.96 (0.28)	10.13 (0.17)
coarse (count)	9606 (4471)	9343 (4117)	15190 (4416)	76677 (11612)	564493 (7379)
coarse (%)	0.18 (0.09)	0.18 (0.08)	0.29 (0.08)	1.47 (0.22)	10.82 (0.14)

TABLE IV: Regularity of deformation fields when training with auxiliary segmentations obtained using FreeSurfer, MSE loss function and smoothness parameter of 0.02, measured using count and percentage of the number of voxels with non-positive Jacobian determinants.

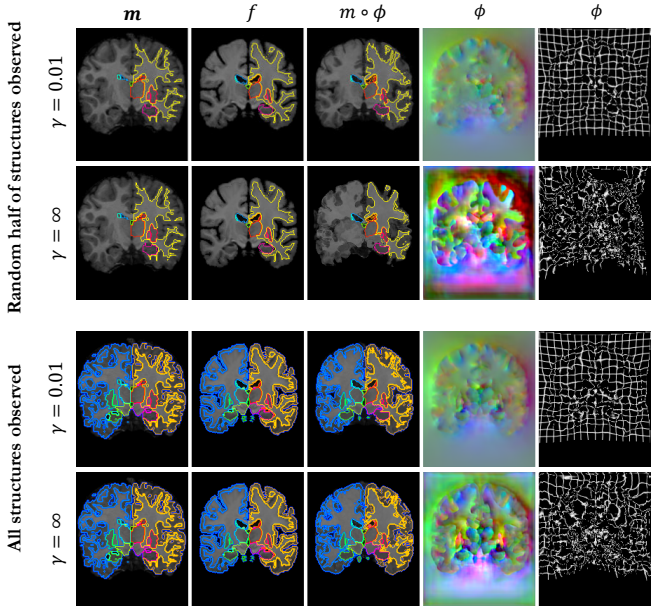

Fig. 10: Effect of γ on warped images and deformation fields. We show the moving image, fixed image, and warped image (columns 1-3) with the structures that were observed at train time overlaid. The resulting deformation field is visualized in columns 4 and 5. While providing better Dice scores for observed structures, the deformation fields resulting from training with $\gamma = \infty$ are far more irregular than those using $\gamma = 0.01$. Similarly, the warped image are visually less coherent for $\gamma = \infty$.

white matter is segmented as one structure. This situation enables evaluation of how the auxiliary data affects anatomical registration at finer scales, within the coarsely delineated structures. To achieve this, we merge the 30 structures into four broad groups: white matter, gray matter, cerebral spinal fluid (CSF) and the brain stem, and evaluate the accuracy of the registration on the original structures.

Fig. 9d (top) presents mean Dice scores over the original 30 structures with varying γ. With γ of 0.01, we obtain an average Dice score of 0.78 ± 0.03 on FreeSurfer segmentations. This is roughly a 3 Dice point improvement over VoxelMorph without auxiliary information (p-value $< 10^{-10}$).

3) Regularity of Deformations: We also evaluate the regularity of the deformation fields both visually and by computing the number of voxels for which the determinant of the Jacobian is non-positive. Table IV provides the quantitative regularity measure for all γ values, showing that VoxelMorph deformation regularity degrades slowly as a function of γ (shown on a log scale), with roughly 0.2% of the voxels exhibiting folding at the lowest parameter value, and *at most* 2.3% when $\gamma = 0.1$. Deformations from models that don't encourage smoothness, at the extreme value of $\gamma = \infty$, exhibit 10–13% folding voxels. A lower γ value such as $\gamma = 0.01$ therefore provides a good compromise of high Dice scores for all structures while avoiding highly irregular deformation fields, and avoiding overfitting as described above. Fig 10 shows examples of deformation fields for $\gamma = 0.01$ and $\gamma = \infty$, and we include more figures in the supplemental material for each experimental setting.

4) Testing on Manual Segmentation Maps: We also test these models on the manual segmentations in the Buckner40 dataset used above, resulting in Fig. 9 (bottom). We observe a behavior consistent with the conclusions above, with smaller Dice score improvements, possibly due to the higher baseline Dice scores achieved on the Buckner40 data.

VI. Discussion and Conclusion

VoxelMorph with unsupervised loss performs comparably to the state-of-the-art ANTs and NiftyReg software in terms of Dice score, while reducing the computation time from hours to minutes on a CPU and under a second on a GPU. VoxelMorph is flexible and handles both partially observed or coarsely delineated auxiliary information during training, which can lead to improvements in Dice score while still preserving the runtime improvement.

VoxelMorph performs amortized optimization, learning global function parameters that are optimal for an entire training dataset. As Fig. 8 shows, the dataset need not be large: with only 100 training images, VoxelMorph leads to state-of-the-art registration quality scores for both training and test sets. Instance-specific optimization further improves VoxelMorph performance by one Dice point. This is a small increase, illustrating that amortized optimization can lead to nearly optimal registration.

We performed a thorough set of experiments demonstrating that, for a reasonable choice of γ, the availability of anatomical segmentations during training significantly improves test registration performance with VoxelMorph (in terms of Dice score) while providing smooth deformations (e.g. for $\gamma = 0.01$, less than 0.5% folding voxels). The performance gain varies

based on the quality and number of anatomical segmentations available. Given a single labeled anatomical structure during training, the accuracy of registration of test subjects *for that label* increases, without negatively impacting other anatomy. If half or all of the labels are observed, or even a coarse segmentation is provided at training, registration accuracy improves for all labels during test. While we experimented with one type of auxiliary data in this study, VoxelMorph can leverage other auxiliary data, such as different modalities or anatomical keypoints. Increasing γ also increases the number of voxels exhibiting a folding of the registration field. This effect may be alleviated by using a diffeomorphic deformation representation for VoxelMorph, as introduced in recent work [5].

VoxelMorph is a general learning model, and is not limited to a particular image type or anatomy – it may be useful in other medical image registration applications such as cardiac MR scans or lung CT images. With an appropriate loss function such as mutual information, the model can also perform multimodal registration. VoxelMorph promises to significantly speed up medical image analysis and processing pipelines, while opening novel directions in learning-based registration.

REFERENCES

[1] B. Fischl, "Freesurfer," *Neuroimage*, vol. 62, no. 2, pp. 774–781, 2012.

[2] A. Klein, J. Andersson, B. A. Ardekani, J. Ashburner, B. Avants, M.-C. Chiang, G. E. Christensen, D. L. Collins, J. Gee, P. Hellier *et al.*, "Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration," *Neuroimage*, vol. 46(3), pp. 786–802, 2009.

[3] 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.

[4] M. Modat, G. R. Ridgway, Z. A. Taylor, M. Lehmann, J. Barnes, D. J. Hawkes, N. C. Fox, and S. Ourselin, "Fast free-form deformation using graphics processing units," *Computer methods and programs in biomedicine*, vol. 98, no. 3, pp. 278–284, 2010.

[5] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning for fast probabilistic diffeomorphic registration," *arXiv preprint arXiv:1805.04605*, 2018.

[6] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9252–9260.

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

[8] R. Bajcsy and S. Kovacic, "Multiresolution elastic matching," *Computer Vision, Graphics, and Image Processing*, vol. 46, pp. 1–21, 1989.

[9] 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, pp. 139–157, 2005.

[10] P. Viola and W. M. Wells III, "Alignment by maximization of mutual information," *International journal of computer vision*, vol. 24, no. 2, pp. 137–154, 1997.

[11] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, "Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain," *Medical image analysis*, vol. 12, no. 1, pp. 26–41, 2008.

[12] Y. Kim, S. Wiseman, A. C. Miller, D. Sontag, and A. M. Rush, "Semi-amortized variational autoencoders," *preprint arXiv:1802.02550*, 2018.

[13] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, "Amortised map inference for image super-resolution," *arXiv preprint arXiv:1610.04490*, 2016.

[14] C. Zhang, J. Butepage, H. Kjellstrom, and S. Mandt, "Advances in variational inference," *arXiv preprint arXiv:1711.05597*, 2017.

[15] C. Cremer, X. Li, and D. Duvenaud, "Inference suboptimality in variational autoencoders," *arXiv preprint arXiv:1801.03558*, 2018.

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

[17] A. V. Dalca, A. Bobu, N. S. Rost, and P. Golland, "Patch-based discrete registration of clinical brain images," in *International Workshop on Patch-based Techniques in Medical Imaging*. Springer, 2016, pp. 60–67.

[18] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios, "Dense image registration through mrfs and efficient linear programming," *Medical image analysis*, vol. 12, no. 6, pp. 731–741, 2008.

[19] J. Thirion, "Image matching as a diffusion process: an analogy with maxwell's demons," *Medical Image Analysis*, vol. 2, no. 3, pp. 243–260, 1998.

[20] B. T. Yeo, M. R. Sabuncu, T. Vercauteren, D. J. Holt, K. Amunts, K. Zilles, P. Golland, and B. Fischl, "Learning task-optimal registration cost functions for localizing cytoarchitecture and function in the cerebral cortex," *IEEE transactions on medical imaging*, vol. 29, no. 7, pp. 1424–1441, 2010.

[21] M. Zhang, R. Liao, A. V. Dalca, E. A. Turk, J. Luo, P. E. Grant, and P. Golland, "Frequency diffeomorphisms for efficient image registration," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 559–570.

[22] C. Davatzikos, "Spatial transformation and registration of brain images using elastically deformable models," *Computer Vision and Image Understanding*, vol. 66, no. 2, pp. 207–222, 1997.

[23] D. Shen and C. Davatzikos, "Hammer: Hierarchical attribute matching mechanism for elastic registration," *IEEE Transactions on Medical Imaging*, vol. 21, no. 11, pp. 1421–1439, 2002.

[24] J. Ashburner and K. J. Friston, "Voxel-based morphometry-the methods," *Neuroimage*, vol. 11, pp. 805–821, 2000.

[25] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformation: Application to breast mr images," *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 712–721, 1999.

[26] X. Pennec, P. Cachier, and N. Ayache, "Understanding the "demon's algorithm": 3d non-rigid registration by gradient descent," pp. 597–605, 1999.

[27] Y. Cao, M. I. Miller, R. L. Winslow, and L. Younes, "Large deformation diffeomorphic metric mapping of vector fields," *IEEE transactions on medical imaging*, vol. 24, no. 9, pp. 1216–1230, 2005.

[28] C. Ceritoglu, K. Oishi, X. Li, M.-C. Chou, L. Younes, M. Albert, C. Lyketsos, P. C. van Zijl, M. I. Miller, and S. Mori, "Multi-contrast large deformation diffeomorphic metric mapping for diffusion tensor imaging," *Neuroimage*, vol. 47, no. 2, pp. 618–627, 2009.

[29] M. Hernandez, M. N. Bossa, and S. Olmos, "Registration of anatomical images using paths of diffeomorphisms parameterized with stationary vector field flows," *International Journal of Computer Vision*, vol. 85, no. 3, pp. 291–306, 2009.

[30] S. C. Joshi and M. I. Miller, "Landmark matching via large deformation diffeomorphisms," *IEEE transactions on image processing*, vol. 9, no. 8, pp. 1357–1370, 2000.

[31] M. I. Miller, M. F. Beg, C. Ceritoglu, and C. Stark, "Increasing the power of functional maps of the medial temporal lobe by using large deformation diffeomorphic metric mapping," *Proceedings of the National Academy of Sciences*, vol. 102, no. 27, pp. 9685–9690, 2005.

[32] K. Oishi, A. Faria, H. Jiang, X. Li, K. Akhter, J. Zhang, J. T. Hsu, M. I. Miller, P. C. van Zijl, M. Albert *et al.*, "Atlas-based whole brain white matter analysis using large deformation diffeomorphic metric mapping: application to normal elderly and alzheimer's disease participants," *Neuroimage*, vol. 46, no. 2, pp. 486–499, 2009.

[33] 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.

[34] M. Modat, D. M. Cash, P. Daga, G. P. Winston, J. S. Duncan, and S. Ourselin, "Global image registration using a symmetric block-matching approach," *Journal of Medical Imaging*, vol. 1, no. 2, p. 024003, 2014.

[35] X. Cao, J. Yang, J. Zhang, D. Nie, M. Kim, Q. Wang, and D. Shen, "Deformable image registration based on similarity-steered cnn regression," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 300–308.

[36] J. Krebs, T. Mansi, H. Delingette, L. Zhang, F. C. Ghesu, S. Miao, A. K. Maier, N. Ayache, R. Liao, and A. Kamen, "Robust non-rigid registration through agent-based action learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2017, pp. 344–352.

[37] M.-M. Rohé, M. Datar, T. Heimann, M. Sermesant, and X. Pennec, "Svf-net: Learning deformable image registration using shape matching," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2017, pp. 266–274.

[38] H. Sokooti, B. de Vos, F. Berendsen, B. P. Lelieveldt, I. Išgum, and M. Staring, "Nonrigid image registration using multi-scale 3d convolutional neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2017, pp. 232–239.

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

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

[41] H. Li and Y. Fan, "Non-rigid image registration using fully convolutional networks with deep self-supervision," *preprint arXiv:1709.00799*, 2017.

[42] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[43] Y. Hu, M. Modat, E. Gibson, W. Li, N. Ghavami, E. Bonmati, G. Wang, S. Bandula, C. M. Moore, M. Emberton *et al.*, "Weakly-supervised convolutional neural networks for multimodal image registration," *Medical image analysis*, vol. 49, pp. 1–13, 2018.

[44] Y. Hu, M. Modat, E. Gibson, N. Ghavami, E. Bonmati, C. M. Moore, M. Emberton, J. A. Noble, D. C. Barratt, and T. Vercauteren, "Label-driven weakly-supervised learning for multimodal deformable image registration," in *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*. IEEE, 2018, pp. 1070–1074.

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

[46] B. K. Horn and B. G. Schunck, "Determining optical flow," 1980.

[47] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2432–2439, 2010.

[48] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, 2011.

[49] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2011.

[50] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu, "Large displacement optical flow from nearest neighbor fields," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2450, 2013.

[51] A. Ahmadi and I. Patras, "Unsupervised convolutional neural networks for motion estimation," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1629–1633.

[52] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2758–2766.

[53] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE conference on computer vision and pattern recognition (CVPR)*, vol. 2, 2017, p. 6.

[54] J. Y. Jason, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *European Conference on Computer Vision*. Springer, 2016, pp. 3–10.

[55] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2017, p. 2.

[56] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Deep end2end voxel2voxel prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 17–24.

[57] C. Bailer, K. Varanasi, and D. Stricker, "Cnn-based patch matching for optical flow with thresholded hinge embedding loss," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, no. 3, 2017, p. 7.

[58] D. Gadot and L. Wolf, "Patchbatch: a batch augmented loss for optical flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4236–4245.

[59] J. Thewlis, S. Zheng, P. H. Torr, and A. Vedaldi, "Fully-trainable deep matching," *arXiv preprint arXiv:1609.03532*, 2016.

[60] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1385–1392.

[61] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, "Transformation-grounded image generation network for novel 3D view synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 702–711.

[62] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," *European Conference on Computer Vision (ECCV)*, pp. 286–301, 2016.

[63] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[64] 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 (MICCAI)*. Springer, 2015, pp. 234–241.

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

[66] J. Marino, Y. Yue, and S. Mandt, "Iterative amortized inference," *arXiv preprint arXiv:1807.09356*, 2018.

[67] M. De Craene, A. du Bois dAische, B. Macq, and S. K. Warfield, "Multi-subject registration for unbiased statistical atlas construction," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2004, pp. 655–662.

[68] R. Sridharan, A. V. Dalca, K. M. Fitzpatrick, L. Cloonan, A. Kanakis, O. Wu, K. L. Furie, J. Rosand, N. S. Rost, and P. Golland, "Quantification and analysis of large multimodal clinical image studies: Application to stroke," in *International Workshop on Multimodal Brain Image Analysis*. Springer, 2013, pp. 18–30.

[69] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults," *Journal of cognitive neuroscience*, vol. 19, no. 9, pp. 1498–1507, 2007.

[70] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto *et al.*, "The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism," *Molecular psychiatry*, vol. 19, no. 6, pp. 659–667, 2014.

[71] M. P. Milham, D. Fair, M. Mennes, S. H. Mostofsky *et al.*, "The ADHD-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience," *Frontiers in systems neuroscience*, vol. 6, p. 62, 2012.

[72] R. L. Gollub, J. M. Shoemaker, M. D. King, T. White, S. Ehrlich, S. R. Sponheim, V. P. Clark, J. A. Turner, B. A. Mueller, V. Magnotta *et al.*, "The mcic collection: a shared repository of multi-modal, multi-site brain image data from a clinical investigation of schizophrenia," *Neuroinformatics*, vol. 11, no. 3, pp. 367–388, 2013.

[73] K. Marek, D. Jennings, S. Lasch, A. Siderowf, C. Tanner, T. Simuni, C. Coffey, K. Kieburtz, E. Flagg, S. Chowdhury *et al.*, "The parkinson progression marker initiative (ppmi)," *Progress in neurobiology*, vol. 95, no. 4, pp. 629–635, 2011.

[74] A. Dagley, M. LaPoint, W. Huijbers, T. Hedden, D. G. McLaren, J. P. Chatwal, K. V. Papp, R. E. Amariglio, D. Blacker, D. M. Rentz *et al.*, "Harvard aging brain study: dataset and accessibility," *NeuroImage*, 2015.

[75] A. J. Holmes, M. O. Hollinshead, T. M. OKeefe, V. I. Petrov, G. R. Fariello, L. L. Wald, B. Fischl, B. R. Rosen, R. W. Mair, J. L. Roffman *et al.*, "Brain genomics superstruct project initial data release with structural, functional, and behavioral measures," *Scientific data*, vol. 2, 2015.

[76] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[77] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

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

SUPPLEMENTARY MATERIAL

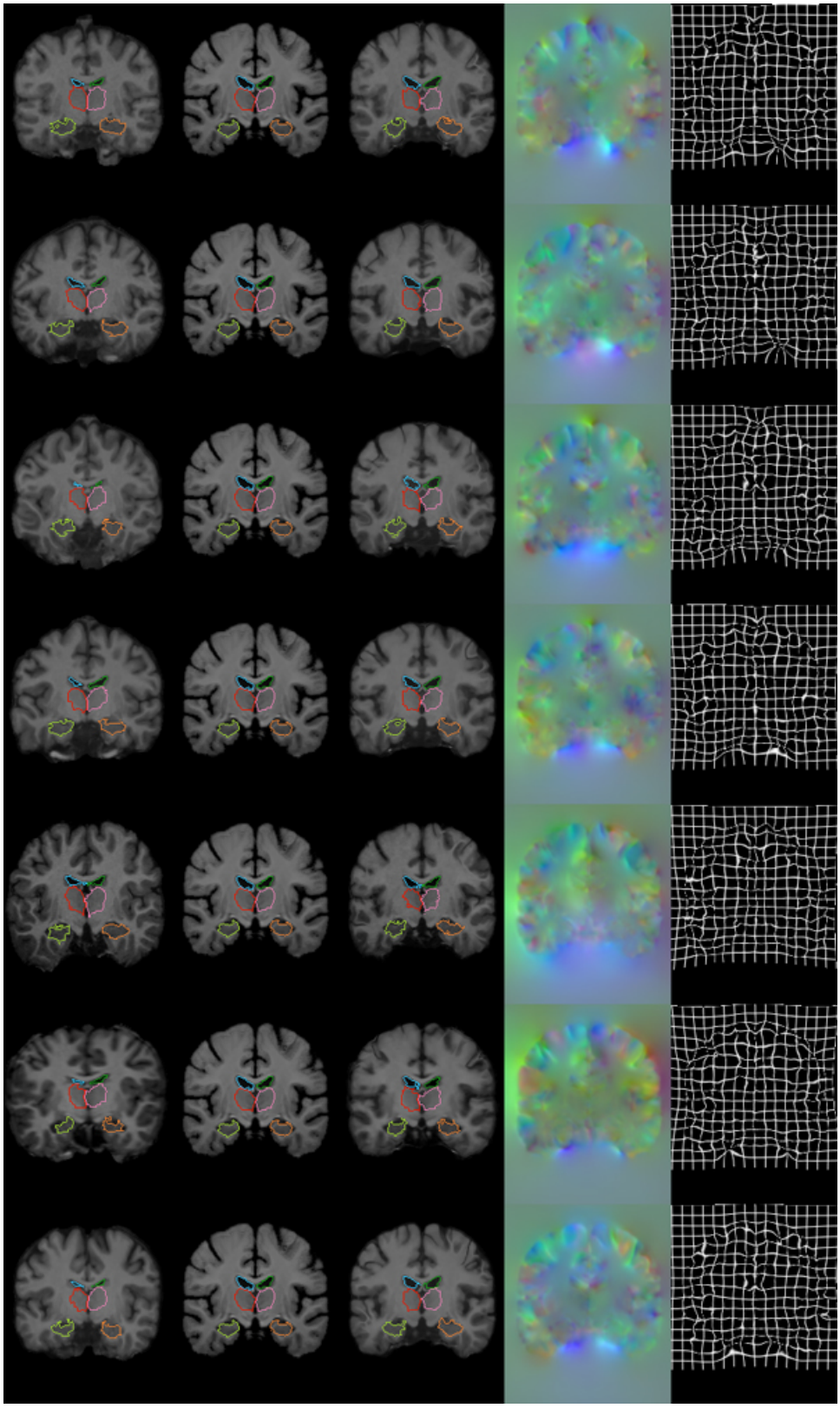

Fig. 11: Example atlas-based VoxelMorph flow fields ϕ (columns 4-5) extracted by registering the moving image (column 1) to the fixed image (column 2). The warped image $m \circ \phi$ is shown in column 3.

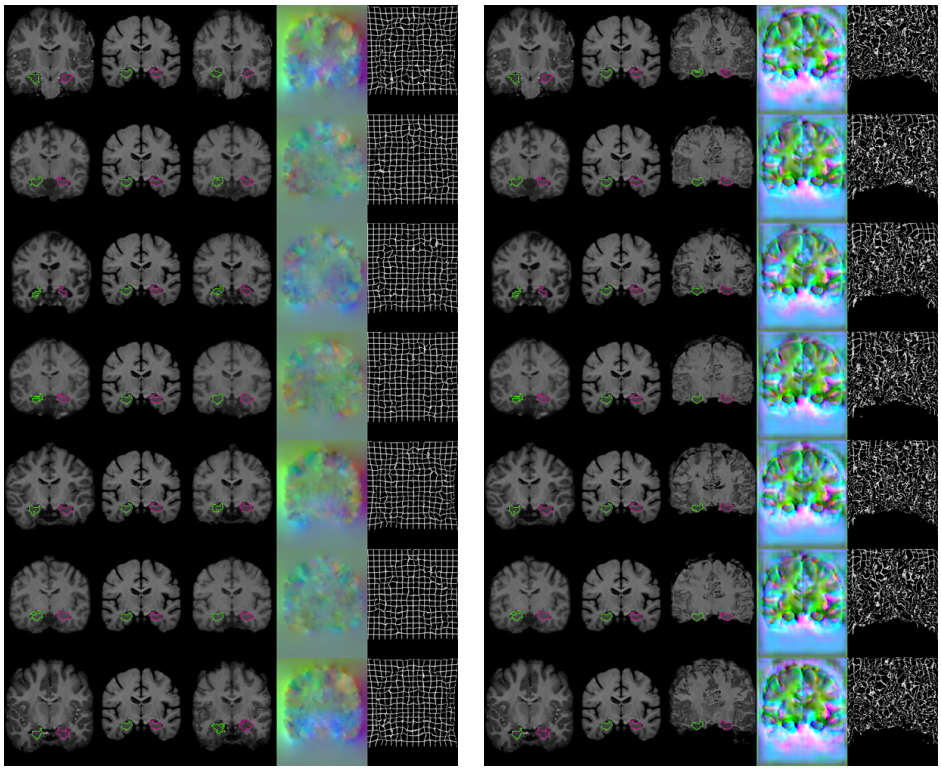

Fig. 12: Auxiliary data experiment where the left and right hippocampus labels are observed at train time. We show the moving image, fixed image and warped image (columns 1-3) with the observed labels overlaid, and the resulting deformation fields (columns 4-5). We use the optimal $\gamma = 0.01$ (left) and the extreme $\gamma = \infty$ (right).

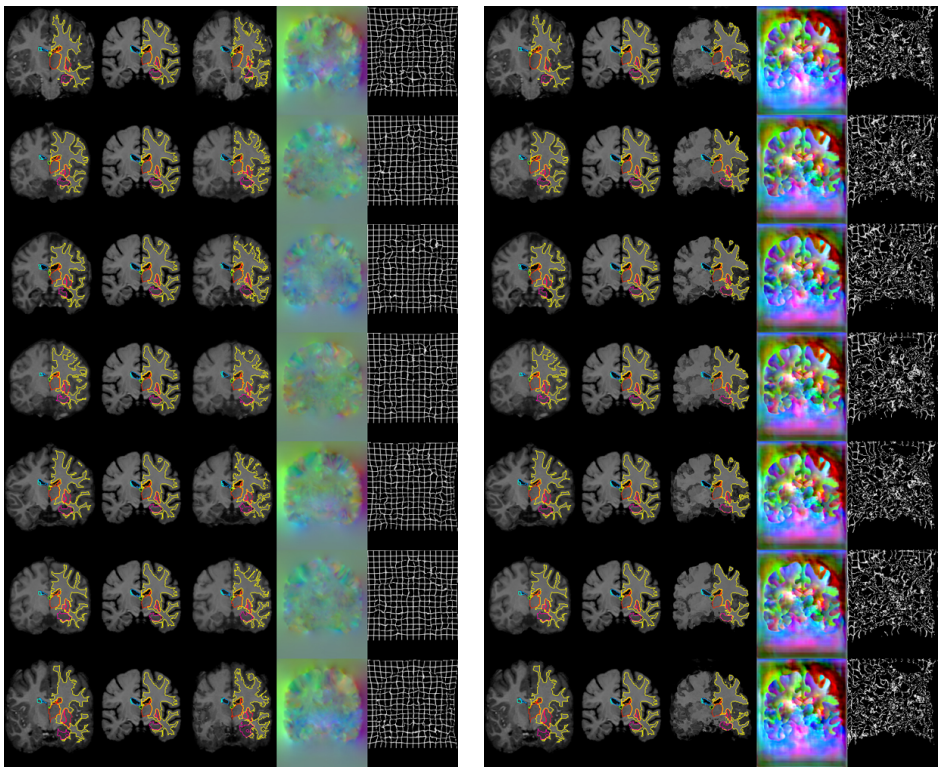

Fig. 13: Auxiliary data experiment where a random half of the labels are observed at train time. We show the moving image, fixed image and warped image (columns 1-3) with the observed labels overlaid, and the resulting deformation fields (columns 4-5). We use the optimal $\gamma = 0.01$ (left) and the extreme $\gamma = \infty$ (right).

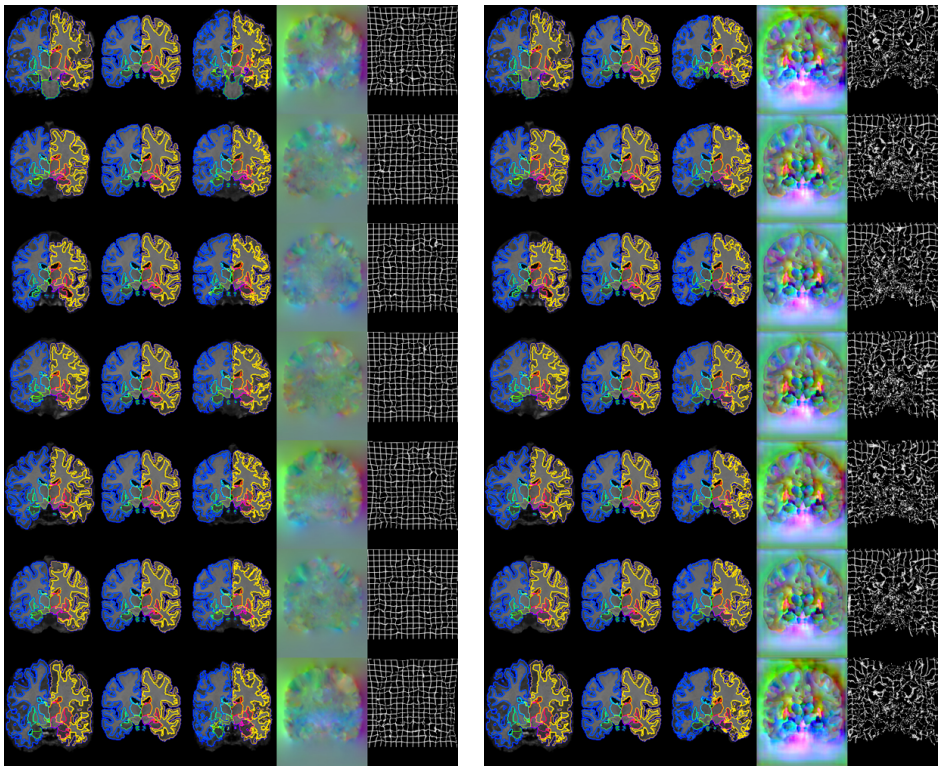

Fig. 14: Auxiliary data experiment where all labels are observed at train time. We show the moving image, fixed image and warped image (columns 1-3) with the observed labels overlaid, and the resulting deformation fields (columns 4-5). We use the optimal $\gamma = 0.01$ (left) and the extreme $\gamma = \infty$ (right).

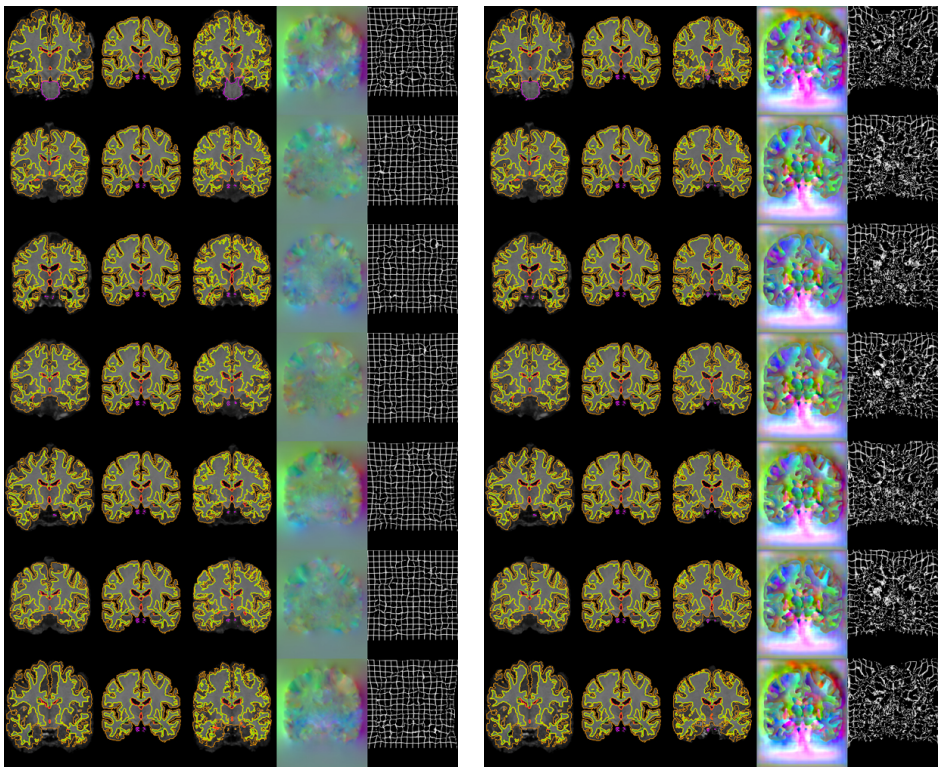

Fig. 15: Auxiliary data experiment where coarse labels are observed at train time. We show the moving image, fixed image and warped image (columns 1-3) with the observed labels overlaid, and the resulting deformation fields (columns 4-5). We use the optimal $\gamma = 0.01$ (left) and the extreme $\gamma = \infty$ (right).