MIRNF: Medical Image Registration via Neural Fields

Shanlin Sun  
University of California, Irvine  
shanlins@uci.edu

Kun Han  
University of California, Irvine  
khan7@uci.edu

Deying Kong  
University of California, Irvine  
deryngk@uci.edu

Chenyu You  
Yale University  
chenyu.you@yale.edu

Xiaohui Xie  
University of California, Irvine  
xhx@uci.edu

Abstract

Image registration is widely used in medical image analysis to provide spatial correspondences between two images. Recently learning-based methods utilizing convolutional neural networks (CNNs) have been proposed for solving image registration problems. The learning-based methods tend to be much faster than traditional optimization based methods, but the accuracy improvements gained from the complex CNN-based methods are modest. Here we introduce a new deep-neural net based image registration framework, named MIRNF, which represents the correspondence mapping with a continuous function implemented via Neural Fields. MIRNF outputs either a deformation vector or velocity vector given a 3D coordinate as input. To ensure the mapping is diffeomorphic, the velocity vector output from MIRNF is integrated using the Neural ODE solver to derive the correspondences between two images. Furthermore, we propose a hybrid coordinate sampler along with a cascaded architecture to achieve the high-similarity mapping performance and low-distortion deformation fields. We conduct experiments on two 3D MR brain scan datasets, showing that our proposed framework provides state-of-art registration performance while maintaining comparable optimization time.

1. Introduction

3D image registration is essential in many medical applications, e.g., motion correction, tracking the progression of lesions, and atlas-based image segmentation. Traditional methods solving the task of image registration are usually based on iterative numerical solutions of an optimization problem. The objective of these methods is to align voxels with similar appearance while enforcing the smoothness constraint. However, due to the large number of parameters involved, these methods tend to be computationally intensive and are usually slow in practice.

Recent deep learning image registration (DLIR) methods [5, 9, 21, 22] show significant advantages in terms of inference time compared to traditional iterative registration algorithms. DLIR methods typically adopt convolutional neural networks (CNNs) to capture the correspondences and perform an end-to-end inference of the displacement or velocity field between pairs of images, under the regularizations such as smoothness and orientation consistency.

Although DLIR methods have achieved fast registration and comparable accuracy, in practice, traditional methods are still dominating the field of image registration for the following reasons. Firstly, DLIR methods usually require a large number of training data to improve their generalizability. However, unlike natural image datasets, medical image datasets tend to be small in size. Secondly, the generalization gap between training data and testing data restricts the performance of a pre-trained neural network on new testing images. For example, if the registration model is trained on healthy brain image pairs while the application is to align the historical scans of patients with brain tumors, it is very likely that a pre-trained registration model will perform well.

Here we propose MIRNF, an optimization-based framework of applying Neural Fields to medical image registration. Neural Field (NF), often called coordinate-based multiple-layer perceptrons (MLPs) or neural implicit functions, takes the 3D coordinates as input and outputs the field quantity. Specifically, we use NF to represent a continuous deformation field or velocity field between two images. Given one coordinate, the output of MIRNF can be either a deformation vector indicating correspondence between two images, or a velocity vector which can be integrated to provide a flow of transformation. In order to avoid the interpolation that may occur during the integration, we leverage the power of Neural ODE solver [2] to calculate the deformation trajectory.
Our main contributions are summarized as follows:

- We introduce a novel Neural Field-based optimization framework named MIRNF, which supports deformable as well as diffeomorphic medical image registration by modeling a continuous deformation field.
- We extend MIRNF to a cascaded version and achieve higher alignment accuracy and less-distorted deformation field. By applying a hybrid coordinate sampler and efficient optimization strategies, the cascaded MIRNF can take similar optimization cost as the single-layer version.
- We test MIRNF on two brain MRI datasets and demonstrate that it leads to state-of-the-art registration results in terms of dice score and ratio of negative Jacobian determinant with respect to the predicted deformation field.

2. Related Works

2.1. Optimization-based Registration Methods

Extensive works have been conducted in 3D deformable image registration through the decades \([1,3,4,6,10]\). Several studies target the task of image registration as an optimization problem in the space of displacement vector field. They optimize the deformable model iteratively with the constraint from a smoothness regularizer which is typically a Gaussian smooth filtering. These include elastotype models \([4]\), free-form deformation with B-splines \([27]\), statistic parametric mapping \([2]\), local affine models \([15]\) and Demons \([33]\). Diffeomorphic image registration with the attributes of topology preserving and transformation invertibility also achieve remarkable progress in various anatomical studies. Some of the popular methods include Large Diffeomorphic Distance Metric Mapping (LDDMM) \([6]\), DARTEL \([1]\) and standard symmetric normalization (SyN) \([3]\). In this field, the deformation is modeled by integrating its velocity over time according to the Lagrange transport equation \([8,12]\) to achieve a global one-to-one smooth and continuous mapping.

2.2. Neural Fields for Visual Computing

Recently, neural field advances as a popular technique in solving visual computing problems. It uses coordinate-based neural networks to parameterize the physical properties of scenes and objects across space and time \([36]\). Commonly, neural field is represented with a feed-forward fully connected neural network. Given one sampled coordinate \(x\), the output of neural field can be either a scalar or vector representing the physical field value. \([24]\) learned a continuous Sign Distance Function (SDF) to represent the 3D shape where the magnitude represents the distance to boundary and sign indicates the direction. Neural Radial Field introduced in \([20]\) was utilized to achieve view-dependent scene representation. Periodic sinusoidal activation function surpassed the relu-based functions in representing complex natural signals and their derivatives as discussed in \([30]\).

2.2.1 Deformation

Neural Fields can be used to represent continuous transformation with more flexibility. As target geometry and appearance are often modeled with neural fields, it is natural to use neural field to represent the transformation. \([23]\) performed 4D reconstruction via learned temporal and spatially continuous vector field. Neural Mesh Flow \([14]\) focused on generating manifold mesh from images or point clouds via conditional continuous diffeomorphic flow. PointFlow \([38]\) incorporated continuous normalizing flows with a principle probabilistic framework to reconstruct 3D point clouds.

2.2.2 Medical Image

Neural fields have been involved in medical image processing, such as 3D image reconstruction or representation. \([31]\) tried to augment the quantities measured in the sensor domain and reconstructed images with less measurement noise. \([29]\) predicted the density value at a 3D spatial coordinate, and was supervised by mapping its value back to the sensor domain. \([35]\) viewed the 2D slice as the samples from 3D continuous function and tried to reconstruct 3D images from the observed tissue anatomy.

3. Method

Our framework MIRNF aims to solve pairwise medical image registration by optimizing the parameters of neural networks which models the deformation between two medical volumes. In Sec. \([3.1]\) we will briefly overview our proposed MIRNF. In Sec. \([3.2]\) we will introduce how we build up the continuous deformation field using neural representation. In Sec. \([3.3]\) we go over the other components make our framework realized. In Sec. \([3.4]\) we will introduce the cascaded MIRNF and the related coordinates sampling and the position encoding strategies.

3.1. Overview

Let \(T \in \mathbb{R}^{D \times H \times W}\) and \(M \in \mathbb{R}^{D \times H \times W}\) denote the target and moving volumes in the task of image registration, \(\Phi(p, t) : \mathbb{R}^3 \times [0, 1] \rightarrow \mathbb{R}^3\) be the continuous trajectory of a 3D point during the time interval \([0, 1]\) where the starting points \(p \in \mathbb{R}^3\) and destination points \(p' \in \mathbb{R}^3\) respectively located in the coordinates of \(T\) and \(M\). Here are two options to model the transformation between two volumes: 1) directly estimate the deformation vector from \(t = 0\) to
t = 1; 2) integrate the velocity vector to generate a trajectory along the time interval. Either way can be modeled via Neural Field (NF), which are basically multi-layer perceptrons $f$ parameterized by weight $\theta$. The NF takes as input a point coordinate and outputs its corresponding field quantity. In Sec. 3.2 we will elaborate how we formulate the regular deformable registration and diffeomorphic registration using NF.

MIRNF solves the pairwise registration problem by updating the weight $\theta$ of neural field through minimizing $\mathcal{L}(T, M, \theta)$ via gradient descent. The loss term $\mathcal{L}(T, M, \theta)$ is composed of the image intensity dissimilarity $\mathcal{L}_{NLCC}(\cdot, \cdot)$ and the negative Jacobian determinant regularization $\mathcal{L}_{Jdet}(\cdot)$. Unlike 3D shape registration, 3D images have more dense data points $(D \times H \times W)$ as well as the points connections. In order to preserve the local orientation consistency, in Sec. 3.3, we propose Intensity Sampler and Coordinate Sampler by which we can achieve a high alignment accuracy and reduce the distortion of the deformation field.

To further improve the registration accuracy and decrease the ratio of voxels with negative Jacobian determinant, we propose a cascaded framework as shown in Fig. 1 which will be described in Sec. 3.4.

3.2. Neural Field

3.2.1 Discrete Grid Deformation Field

Instead of using neural network to model the deformation field, the most straightforward way is to use a trainable volumetric parameter (with shape $D \times H \times W$) to represent the discrete field. The parameters to be optimized are the deformations $\Delta p$ with respect to the grid coordinates $p$.

3.2.2 Neural Deformation Field

We can use NF to describe a continuous deformation field. NF outputs deformations of 3D positions directly, i.e., $p' = p + \Phi(p, 1) = p + f_{\theta}(p)$. Since NF has encoded the deformation field into neural network, the optimization parameters are the weights of neural network.

3.2.3 Neural Diffeomorphic Deformation Field

When it comes to the diffeomorphic transformation, the deformation trajectory $\Phi$ is the solution of the initial value problem (IVP) of an ordinary differential equation (ODE) as below:

$$\frac{\partial \Phi}{\partial t}(p, t) = v(\Phi(p, t), t) \quad \text{s.t.} \quad \Phi(p, 0) = p \quad (1)$$

where $v(p, t): \mathbb{R}^3 \times [0, 1] \mapsto \mathbb{R}^3$ is the velocity field that governs the transformation trajectory $\Phi$. If velocity field $v$ is globally Lipschitz continuous, the solution for the IVP exists and is unique in the interval $[0, 1]$, which means any two ODE trajectories do not cross each other. This can provide the diffeomorphic registration with the property of topology preservation to maintain structure consistency. Usually, we assume $v$ is stationary, which means, the velocity of each 3D point in the field only depends on its coordinate. The diffeomorphic deformation field $\Phi(p, 1)$ can be achieved by integrating stationary velocity field $v(\cdot)$.

$$\Phi_i(p, 1) = \Phi_i(p, 0) + \int_0^1 v(\Phi(p, t))dt \quad (2)$$

In our work, we solve the integral of the velocity field using a Differentiable ODE Solvers (NODE) where the neural network $f_{\theta}$ serves as the dynamic function, representing the neural velocity field. The ODE solver applied in our experiments is Runge-Kutta method with order and step size as hyperparameters. As the order increases and step size decreases, the better approximation of deformation trajectories can be provided by Runge-Kutta method but the increasing number of function evaluation also results in the increasing runtime. For the backward pass, NODE utilizes a technique called the Adjoint Sensitivity method which helps retrieve the gradient of optimization loss with respect to the network parameters by solving another ODE backwards in time.

3.3. Optimization

The objective function of optimization is NLCC regularized by jacobian determinant. Based on the choice of loss functions, we will discuss the advantages and disadvantages of four different coordinate samplers in the following.

3.3.1 Objective Functions

In each optimization iteration, we sample $N$ voxels with coordinates $p$ and get destination coordinates $p'$ by warping the estimated deformation $\Phi$. Single-layer framework and Cascaded framework share the same similarity loss and regularization term. The overall loss is:

$$\mathcal{L} = \mathcal{L}_{NLCC} + \lambda_{Jdet} \cdot \mathcal{L}_{Jdet} \quad (3)$$

We apply normalized local cross-correlation (NLCC) with a window size 9 as our similarity loss of alignment results, written as:

$$\mathcal{L}_{NLCC} = NLCC(\mathcal{S}^{\text{linear}}(T, p), \mathcal{S}^{\text{linear}}(M, p')) \quad (4)$$

Besides, in order to maintain the local orientation consistency, we follow [21] to impose a Jacobian determinant regularization $\mathcal{L}_{Jdet}$. If the Jacobian determinant at a given point $p$ is positive, then the deformation field preserves the
Figure 1. **Overview of MIRNF**, where **CS** denotes Coordinate Sampler, **IS** denotes Intensity Sampler, **PE** denotes Position Encoding and **NF** is short for Neural Field. Both single-layer and cascaded framework rely on NF to predict the coordinate deformations between moving volume $M$ and target volume $T$ and the deformations are spatially continuous. The cascaded framework is composed of two single-layer MIRNFs with different coordinate samplers and position embedding frequencies. As the indicated by the (dashed) arrows in the image, the two single-layer MIRNFs are separately optimized.

$$L_{Jdet} = \frac{1}{N} \sum_{i=1}^{N} \max(0, -J_{\Phi}(p^i)) \quad (5)$$

where $p^i$ is the $i$th coordinate in the sampled batch and $J_{\Phi}(p^i)$ denotes the determinant of Jacobian matrix of the deformation field $\Phi$ at position $p^i$.

Figure 2. **Fields for Coordinate Deformation** – In the above figure, blue modules indicate the optimization parameters. All three field representation can be optimized via gradient-based methods. (a) is an discrete representation of a deformation field while (b) and (c) shows our NF-based representation of a deformation / velocity field in the continuous fashion.

### 3.3.2 Intensity Sampler

estimates the intensities of sub-voxel positions given source images. We applied linear interpolation as intensity sampler in our experiments, notated as $I_{\text{linear}}$. We will also present the registration results in the supplementary materials, using another intensity sampler $I_{\text{neural}}$ by a neural field for medical images.

### 3.3.3 Coordinate Sampler

Most coordinate-based neural networks apply *Random Sampler* because random sampling allows for unbiased co-
Figure 3. **Coordinate Samplers** – illustration of four different coordinate samplers, sampling 4 coordinates per batch from all 36 2D coordinates. Yellow and green blocks are different batches of sampling coordinates.

ordinates collection. But in our case, random sampler is unsuitable because the complexity of registration in image domain is much higher than other tasks and the calculation of $\mathcal{L}_{NLCC}$ and jacobian determinant requires a grid-structure deformation. Therefore, instead of random sampling, we prefer coordinate samplers with spatial regularity as shown in Fig. 3b-d.

Coordinate samplers with spatial regularity enable efficient $NLCC$ computation and discretized approximation of Jacobian matrix. **Downsize Sampler** sample coordinates every pre-defined step size along every dimension. Fig. 3b illustrates how to downsample the coordinates with a step size 3. Positions sampled by downsize sampler can well cover the distribution of the field’s entirety but the approximation of Jacobian matrix might be of significant flaws due to downsizing. The consequence is, the single-layer framework applied downsize sampler shows great alignment accuracy but bad local orientation consistency. **Patch Sampler** samples single high-resolution patch per batch and the locations of sampled patches are random. Compared to the downsize sampler, it can provide more accurate Jacobian matrix approximation but the sampled coordinates vary greatly across different optimization iterations. **Mini-Patch Sampler** select positions from more high-resolution patches with smaller size for one iteration. It combines the advantages of both downsize sampler and patch sampler, but the drawback lies in the computation of $NLCC$. Specifically, a large portion of points along patch borders get inaccurate $NLCC$, resulting from padding operation. We choose downsize sampler for single-layer framework because registration accuracy is the primary concern in most times, rather than local orientation consistency.

### 3.4. Cascaded Framework

We further improve the performance of our work by stacking single-layer models, motivated by the observation that the single-layer framework with downsized coordinate sampler is good at accurate alignment but weak at topology preserving. While the single-layer framework sampling coordinates in mini patches shows opposite strengths and weakness. The goal of our cascaded registration framework is to enhance the complementarity of two different training strategies.

As is shown in Fig. 3b, our cascaded framework can be separated into two stages, which are basically single-layer frameworks having different coordinate samplers, position embedding layers as well as neural fields. $p_1$ and $p_2$ are sampled by downsize coordinate sampler and mini-patch coordinate sampler respectively and their mapping destinations $p'_1$ and $p'_2$ are both estimated by neural fields. The difference is, $p'_1$ is obtained through one-time deformation but $p'_2$ is the results of cascaded transformations.

#### 3.4.1 Hybrid Coordinate Sampler

What we observed is that, at the beginning iterations of neural field optimization with downsize coordinate sampler, the ratio of positions with negative Jacobian determinant is low, but with more optimization iterations, the negative Jacobian determinant ratio goes up. It can be explained by two reasons. The initially optimized neural field are blurred because of spectral bias, which means the discretized approximation of Jacobian matrix with respect to such blurred deformation field will be more accurate, so that the negative Jacobian determinant constraint is fully functional. Also, a blurred neural velocity field is friendly for ODE solvers to calculate more precise integration.

Having achieved the blurred deformation field using downsize sampler, we choose to apply mini-patch coordinate sampler for another neural representation of residual deformations because to model the high frequency residual deformations, the high-resolution patch-based sampler is a better option. Compared with patch coordinate sampler, mini-patch sampler is able to cover coordinates sparsely spreading the whole space. Therefore, we apply the mini-patch coordinate sampler for optimizing the second stage of the cascaded framework.
3.4.2 Position Encoding

maps input coordinates to a higher dimensional embedding in order to learn an encoding with high frequencies \([20, 32]\). The mappings can be realized by a family of functionals \(e_i : \mathbb{R}^d \to \mathbb{R}\), and the Position Encoding (PE) can be written as:

\[ PE(p) = (e_1(p), e_2(p), \ldots, e_n(p)) \]  

We follow the suggestion from Tancik et al. [32], encoding coordinates via fourier feature mapping, such that

\[ e_n(p) = \left[ \cos \left( 2\pi b_n^\top p \right), \sin \left( 2\pi b_n^\top p \right) \right]^\top \]  

where \(b_n\) are frequency vectors randomly sampled i.i.d. from a Gaussian distribution with standard deviation \(\sigma\). The higher the \(\sigma\), the more likely the model would fit small details. In our cascaded framework, the \(\sigma\) is set as 3 and 6 separately for encoding downsize sampled and mini-patch sampled coordinates because we want the first stage to play the role of global alignment and the second stage to focus on the regions where deformations change at a high rate. As you can tell from Fig. 1, we don’t apply position encoding in the single-layer framework. Instead, we choose to replace the ReLU activation layers with sinusoidal function [30] to mitigate the issue of spectral bias. We found the there is no significant performance differences caused by the choice of fourier feature mapping or sinusoidal activation function under the single-layer MIRNF framework.

3.4.3 Optimization

Similar to single-layer framework, the optimization goal of cascaded framework is given by:

\[ L = w_1 \cdot (L_{NLCC}^1 + \lambda_{J_{det}} \cdot L_{J_{det}}^1) + w_2 \cdot (L_{NLCC}^2 + \lambda_{J_{det}} \cdot L_{J_{det}}^2) \]  

where \(w_1\) and \(w_2\) are either 0 or 1 and sum up to 1. In other words, the two single-layer models of the cascaded framework are optimized separately. More specifically, one neural field is optimized firstly, then its NF is fixed and provides the initial deformed coordinates for the second NF. First-stage optimization usually stops at very early iteration and only the parameters of the first NF is updated. Second-stage optimization takes longer and only update the parameters of the second NF. \(L_{NLCC}^1, L_{NLCC}^2, L_{J_{det}}^1, L_{J_{det}}^2\) follows the same design for the single-layer framework, introduced in Sec. 3.3.1

3.5. Implementation Details

The down-sampling step size used in downsize coordinate sampler is set as 3 along all dimensions. Mini-patch coordinate sampler selects 8 patches in size of \(32 \times 32 \times 32\) per optimization iteration. The neural velocity field or deformation field in MIRNF is a feed-forward network with sinusoidal function (SIREN [30]) as the activation function. The numbers of network layers applied in single-layer framework and cascaded framework are different. The SIREN model for single-layer deformable framework has 5 fully connected layers with 256 hidden feature size. The SIREN model for cascaded deformable framework is made up with 4 fully connected layers with 256 hidden feature size on top of mapping coordinates to 128-dimension fourier features. Due to the computation inefficiency of NODE, we design shallower SIREN models as the dynamic functions (neural velocity field) which contain 4-layer MLPs for single-layer diffeomorphic registration framework and 3-layer MLPs for cascaded case. We choose Runge-Kutta method of order \(k = 4\) with step size 0.25 as the ODE solver in our diffeomorphic registration experiments. The network parameters are optimized using Adam algorithm with a learning rate of \(1e^{-4}\). During the optimization, \(\lambda_{J_{det}}\) are set as 100 for both single-layer and cascaded MIRNFs. The maximum optimization iterations of single-layer MIRNF is set to be 900. The first stage of the cascaded MIRNF is optimized for 200 iterations and the second stage is further optimized for 900 iterations. Our framework is implemented using PyTorch [25] and all experiments are deployed on a machine with a GTX 2080Ti GPU and an Intel (i7-7700K) CPU.

4. Experiments

4.1. Dataset

We use two public 3D brain MR datasets: the Mindboggle101 and the OASIS to evaluate the performance of our MIRNF and other image registration algorithms in comparisons. Mindboggle101 [13] consists of 101 T1-weighted MR scans of healthy participants coming from 5 sata sources. As [37] did, we select 31 cortical regions for evaluation. OASIS dataset contains 425 T1-weighted MR images of subjects aged from 18 to 96. 35 subcortical regions are labelled for evaluation. Our framework preprocess these two dataset in the same way, including skull stripping, resampling to \((1mm \times 1mm \times 1mm)\) spacings, and normalizing intensities by the maximum intensity of each volume. For the other compared methods, we prepared datasets as they suggested in their paper.

4.2. Experimental Setup

For both datasets, we randomly selected 20 scans as target volumes and 3 scans as the atlas. All the methods in experiments are trying to map the moving scans to the atlas, and our evaluation matrices is Dice’s coefficient (DSC) and ratio of coordinates with negative Jacobian determinant (\(J_{\leq 0}\)).

We compare our methods with traditional optimization-
based and recent learning-based methods. Regarding optimization-based algorithms, we select SyN [3] and NiftyReg [27], which are representations of traditional diffeomorphic and non-rigid registration methods. We run SyN implementation using ANTs toolbox with 3 different iteration settings ([40, 20, 0], [80, 60, 40] and [160, 120, 80]). Similarly, we run NiftyReg using two configurations with maximum level and iteration set to [3, 500] and [5, 1000]. We also implemented a discrete grid diffeomorphic deformation field optimizer (named Grid in our experiments) using PyTorch, which is similar to Diffeomorphic Demons [34] implemented by [28]. When it comes to learning-based models, we select VoxelMorph [5] (VM) and SYMNet [21]. Both VM and SYMNet were trained using NLLC similarity loss with the smoothness regularization. The orientation consistency weight and magnitude weight of the SYMNet were set to 100 and 0.001. We randomly sample 86 and 250 cases from Mindboggle and OASIS datasets to train the corresponding learning-based registration models.

4.3. Registration Results Comparison

Tab. 1 shows the registration performance comparisons between our MIRNF frameworks and other state-of-art algorithms regarding: 1) DSC of fixed mask and warped mask; 2) \( J_{\leq 0} \) of deformation fields; 3) GPU memory consumption during optimization or inference. Overall speaking, Tab. 1 shows our MIRNF framework can provide decent registration performance on different datasets with multiple settings. We provide numerical registration results generated by four variants of our MIRNF frameworks in Tab. 1. DS and DC denote the deformation frameworks of single layer and cascaded layers respectively. DDS and DDC represent the diffeomorphic deformation frameworks with single-layer and cascaded settings. The diffeomorphic registration methods are shaded in the Tab. 1.

Notably, the entries under ‘Memory’ title have different meanings. For learning-based model, we list their GPU memory consumption in training and inference. While for our cascaded MIRNF models, the numbers indicate the GPU memory cost in two stages of optimization. From the table, we can tell that our model cost much less memory compared with other neural net based methods (VM, SYMNet, Grid) during optimization (training). Our optimization memory consumption is even comparable with the inference memory consumption of two learning-based models.

Compared with other regular deformable registration models, our DS and DC shows significant advantages in terms of registration accuracy on Mindboggle and OASIS dataset. In the next section, we will show our deformable registration framework can achieve great performance within a short runtime. Our DDS and DDC frameworks present a very competitive results with respect to DSC and \( J_{\leq 0} \) on both two datasets. All the competing methods perform well on OASIS dataset because it is relatively easier compared to Mindboggle. We will discuss the difference between Mindboggle and OASIS when explaining Fig. 5 in Sec. 4.6.

4.4. Runtime and Performance

In this section, we will present our registration performance at different optimization iterations and compare them with the other optimization-based methods. As is shown in Tab. 2 for all registration methods of MIRNF, DSCs keep improving but \( J_{\leq 0} \) getting worse as optimization iterations grow. We can also observe that the cascaded framework can better resist the degeneration of local orientation consistency, which validates the effectiveness of hybrid coordinate samplers. Fig. 4 shows the relationship between model performance and running time. As we can see from the plot, our DS and DC framework optimize rapidly thanks to the efficient downsize coordinate sampler and lightweight network architecture. Under the similar running time, they can outperform the NiftyReg with great margins. Our diffeomorphic registration model runs slower but can still provide results with better DSC, compared to the SyN and Grid when taking the similar optimization time. In Tab. 1, SyN shows its strength in topology preserving, but Fig. 4 reveals that our DDS can achieve even lower \( J_{\leq 0} \) with slightly higher DSC and shorter optimization time.

Compared with single-layer frameworks, the cascaded frameworks can generate better performance for most of iterations. However, they running time is significantly longer. The issue lies in the mini-patch coordinate sampler we apply for the second stage of optimization, which takes much longer time than the downsize coordinate sampler.

4.5. Ablation Study

Tab. 3 demonstrates the efficacy of hybrid coordinate samplers we design for cascaded MIRNF. With the same single-layer model, using the mini-patch coordinate sampler is better for decreasing \( J_{\leq 0} \) and using downsize coordinate sampler can provide higher DSC. Although applying downsize coordinate sampler at both stages of cascaded framework during optimization (Downsize\(^2\)) can further improve registration accuracy, \( J_{\leq 0} \) becomes worse. Our design to apply downsize coordinate samplers at the first stage and mini-patch sampler for the second stage can improve the registration accuracy and alleviate deformation distortion at the same time.

4.6. Visualization

Fig. 5 shows the registration results using different MIRNF framework on OASIS dataset, on which the regular deformation models outperforms diffeomorphic defor-
Table 1. Registration Results Comparison

| Category     | Metrics | DSC (↑) | J≤0 (↓) | Memory (MB) |
|--------------|---------|---------|---------|-------------|
|              | Model/Dataset | Mindboggle | OASIS | Mindboggle | OASIS |        |
| Learning-based | VM      | 0.5463  | 0.7589  | 7.62e-04  | 1.03e-03 | 8036 | 3837 |
|               | SYM_Net | 0.5672  | 0.7799  | 2.30e-05  | 1.69e-05 | 10565 | 3031 |
| Optimization-based | NiftyReg | 0.5093  | 0.7706  | -         | -       | -     | -    |
| SyN          | 0.5469  | 0.7787  | 2.30e-05 | 1.69e-05 | -       | -     | -    |
| Grid         | 0.5494  | 0.6830  | 1.16e-04 | 2.24e-04 | 5587    |        |       |
| MIRNF        | DS      | 0.5968  | 0.7766  | 2.80e-03  | 3.49e-03 | 3886  |       |
|              | NiftyReg | 0.5093  | 0.7706  | -         | -       | -     | -    |
| SyN          | 0.5469  | 0.7787  | 2.30e-05 | 1.69e-05 | -       | -     | -    |
| Grid         | 0.5494  | 0.6830  | 1.16e-04 | 2.24e-04 | 5587    |        |       |
|              | DDS     | 0.6018  | 0.7744  | 3.70e-05  | 1.27e-05 | 4011  | 4025 |
|              | DDC     | 0.6078  | 0.7744  | 3.70e-05  | 1.27e-05 | 4011  | 4025 |

Table 2. Optimization Iterations vs. Performance

| Metrics | DSC (↑) | J≤0 (↓) |
|---------|---------|---------|
| Iters / Model | DS | DC | DDS | DDC | DS | DC | DDS | DDC |
| 100     | 0.5451  | 0.5101 | 0.5510 | 0.5512 | 7.80e-04 | 1.04e-03 | 2.10e-07 | 9.64e-06 |
| 300     | 0.5767  | 0.5781 | 0.5884 | 0.5908 | 1.57e-03 | 1.24e-03 | 1.79e-05 | 2.01e-05 |
| 600     | 0.5903  | 0.5971 | 0.5973 | 0.6025 | 2.27e-03 | 1.43e-03 | 4.49e-05 | 3.06e-05 |
| 900     | 0.5986  | 0.6047 | 0.6018 | 0.6067 | 2.80e-03 | 1.58e-03 | 6.45e-05 | 3.70e-05 |

Figure 4. Runtime vs. Performance

5. Conclusion

In this paper, we propose a novel medical image registration framework called MIRNF. We model the registration task as optimizing a continuous deformation field represented by Neural Fields. MIRNF can provide solutions for fast non-rigid registration (DS), more accurate but slower non-rigid registration (DC), accurate diffeomorphic registration (DDS) as well as slow, accurate and more topology preserving diffeomorphic registration (DDC). Compared with other optimization-based registration method,
Table 3. Ablation Study on Coordinate Samplers

| Category    | Model + CS / Metrics | DSC (↑) | J≤0 (↓) |
|-------------|----------------------|---------|---------|
| Deformation | DS + Mini-Patch      | 0.5168  | 1.44e-04 |
|             | DS + Downsize        | 0.5968  | 2.80e-03 |
|             | DC + Downsize        | 0.6087  | 3.71e-03 |
|             | DC + Downsize & Mini-Patch | 0.6047  | 1.58e-03 |
| Diffeomorphic| DDS + Mini-Patch     | 0.5441  | 3.25e-07 |
|             | DDS + Downsize       | 0.6018  | 6.46e-05 |
|             | DDC + Downsize²      | 0.6089  | 3.19e-04 |
|             | DDC + Downsize & Mini-Patch | 0.6078  | 3.70e-05 |

Figure 5. Qualitative comparison between results generated by different MIRNF frameworks on OASIS dataset.

our framework shows advantages in alignment accuracy and local orientation consistency.

References

[1] John Ashburner. A fast diffeomorphic image registration algorithm. Neuroimage, 38(1):95–113, 2007.
[2] John Ashburner and Karl J Friston. Voxel-based morphometry—the methods. Neuroimage, 11(6):805–821, 2000.
[3] 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.
[4] Ruzena Bajcsy and Stane Kovačič. Multiresolution elastic matching. Computer vision, graphics, and image processing, 46(1):1–21, 1989.
[5] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. Voxelmorph: a learning framework for deformable medical image registration. IEEE transactions on medical imaging, 38(8):1788–1800, 2019.
[6] M Faisal Beg, Michael I Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesics of diffeomorphisms. International journal of computer vision, 61(2):139–157, 2005.
[7] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. Advances in Neural Information Processing Systems, 2018.
[8] Gary E Christensen, Richard D Rabbitt, and Michael I Miller. Deformable templates using large deformation kinematics. IEEE transactions on image processing, 5(10):1435–1447, 1996.
[9] Adrian V Dalca, Guha Balakrishnan, John Guttag, and Mert R Sabuncu. Unsupervised learning for fast probabilistic diffeomorphic registration. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 729–738. Springer, 2018.
[10] Adrian V Dalca, Andreea Bobu, Natalia S Rost, and Polina Golland. Patch-based discrete registration of clinical brain images. In International Workshop on Patch-based Techniques in Medical Imaging, pages 60–67. Springer, 2016.
[11] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. arXiv preprint arXiv:1904.01681, 2019.
[12] Paul Dupuis, Ulf Grenander, and Michael I Miller. Variational problems on flows of diffeomorphisms for image matching. Quarterly of applied mathematics, pages 587–600, 1998.
[13] Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In International conference on machine learning, pages 3154–3164. PMLR, 2020.
[14] Kunal Gupta. Neural mesh flow: 3d manifold mesh generation via diffeomorphic flows. University of California, San Diego, 2020.
[15] Pierre Hellier, Christian Barillot, Etienne Mémin, and Patrick Pérez. Hierarchical estimation of a dense deforma-
tion field for 3-d robust registration. *IEEE transactions on medical imaging*, 20(5):388–402, 2001.

[16] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. SaPe: Spatially-adaptive progressive encoding for neural optimization. *Advances in Neural Information Processing Systems*, 34, 2021.

[17] Jacob Kelly, Jesse Bettencourt, Matthew J Johnson, and David K Duvenaud. Learning differential equations that are easy to solve. *Advances in Neural Information Processing Systems*, 33:4370–4380, 2020.

[18] Arno Klein and Jason Tourville. 101 labeled brain images and a consistent human cortical labeling protocol. *Frontiers in neuroscience*, 6:171, 2012.

[19] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.

[20] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[21] Tony CW Mok and Albert Chung. Fast symmetric diffeomorphic image registration with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4644–4653, 2020.

[22] Tony CW Mok and Albert Chung. Large deformation diffeomorphic image registration with laplacian pyramid networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 211–221. Springer, 2020.

[23] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5379–5389, 2019.

[24] Jeong Joon Park, Peter Florence, Julian Straub, Richard Mildenhall, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Neural Information Processing Systems*, 2020.

[25] J-P Thirion. Image matching as a diffusion process: an analogy with maxwell’s demons. *Medical image analysis*, 2(3):243–260, 1998.

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

[27] Qing Wu, Yuwei Li, Lan Xu, Ruiming Feng, Hongjiang Wei, Qing Yang, Boliang Yu, Xiaozhao Liu, Jingyi Yu, and Yuyao Zhang. Irem: High-resolution magnetic resonance image reconstruction via implicit neural representation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 65–74. Springer, 2021.

[28] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. arXiv preprint arXiv:2111.11426, 2021.

[29] Zhenlin Xu and Marc Niethammer. Deepatlas: Joint semi-supervised learning of image registration and segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 420–429. Springer, 2019.

[30] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019.

[31] Han Zhang, Xi Gao, Jacob Unterman, and Tom Arodz. Approximation capabilities of neural odes and invertible residual networks. In *International Conference on Machine Learning*, pages 11086–11095. PMLR, 2020.
A. Model Review

Fig. 6 is an illustration of three different representations of deformation fields. As is demonstrated in Fig. 6, discrete Grid Deformation Field represents the deformation vectors of infinite number of grid points. Neural Deformation Field (Fig. 6b) and Neural Diffeomorphic Deformation Field (Fig. 6c) are continuous field representations. Their difference is neural deformation field predicts deformation vector directly given a point coordinate while neural diffeomorphic deformation field predicts velocity vector given a point coordinate. Neural diffeomorphic deformation field relies on Neural ODE (NODE) [7] to integrate the velocity field to get the deformation field. In Fig. 6, we present deformation fields evaluated at three times but it requires more steps for integration in our implementation.

B. Neural Intensity Sampler

Initially, our primary choice for intensity sampler is neural representation (\(IS^{\text{neural}}\)) of medical images, which is basically a Siren model [30]. With the continuous representation of a medical volume, the intensity of destination points does not only depend on the local neighborhood so that the artifacts are introduced by interpolations. But in practice, the deformations optimization would be negatively affected by the imperfect intensity estimation by the neural intensity field. Here we will present the diffeomorphic registration results on Mindboggle dataset by replacing \(IS^{\text{linear}}\) with \(IS^{\text{neural}}\). We denote the model using neural field as intensity sampler with the superscript \(n\) in Tab. 4.

As is shown in Tab. 4 using neural field as intensity can bring about better \(J_{\leq 0}\) but worse DSC. The deformations optimization is negatively affected by the imperfect intensity estimation by the neural intensity field. We will keep exploring the high-quality neural representation of medical images in the future.

C. Velocity Field Integration

In our work we integrate the velocity field using NODE which is naturally compatible with the continuous neural representation of velocity field. Traditionally, we used scaling and squaring (ss) technique to obtain deformation field from velocity field that defined on the regular grid. We design an experiment to demonstrate the difference of integrating a 2D stationary velocity field via ss and NODE, where the the stationary velocity field is governed by the ode \(\frac{dp_x}{dt} = p_y\) and \(\frac{dp_y}{dt} = -p_x\), so that the deformation trajectory can be expressed as:

\[
x(t) = y(0) \ast \sin(t) + x(0) \ast \cos(t) \\
y(t) = y(0) \ast \cos(t) - x(0) \ast \sin(t)
\]  (10)

Fig. 7a shows the original grid points (lighter) and transformed grid points (brighter) defined by Eq. 10. The color of each points indicate its coordinates so that we can observe the point correspondence before and after the spatial transformation. Fig. 7b shows that ss performs bad at locations requiring extrapolation and Fig. 7c can provide accurate estimation of deformation at any point.

D. Qualitative Results

In the main paper, we have presented the qualitative results of four variants of our framework on OASIS dataset. Here we supplement the qualitative comparisons on Mindboggle dataset.

E. Limitations and Future Works

The primary limitation of our work is the optimization time of MIRNF framework. As we can see in Tab. 5 the optimization time of DS is 0.09s optimization iteration respectively. It takes 24s to finish the first stage optimization of DC for 200 iterations and takes 2s per optimization iteration of the second stage. The time cost per iteration in the second stage is much longer because the mini-patch coordinate sampler is less efficient than the downsize coordinate sampler applied in the first stage of DC and DS. To generate one batch of positions in our experiments, downsized coordinate sampler only needs to generate three integer number ranging between 0 and 3 while mini-patch coordinate sampler needs to generate three sets of 8 integer number from \([0, 192]\), \([0, 218]\) and \([0, 192]\) respectively. If we can improve the efficiency of the mini-patch coordinates, the optimization time needed for cascaded model can be greatly decreased.

From Tab. 5 we can also tell that the optimization time of diffeomorphic registration framework is much longer than the regular deformable registration method. The main bottleneck is the NODE module, which requires repeated functional evaluation to solve ODEs within a given error tolerance. To speed up the integration of velocity field, we can increase the step size or error tolerance of ODE solvers so that fewer functional evaluations are required, in the price of larger errors. Till now, we think Runge–Kutta methods with order 4 and step size 0.25 achieve the best balance between speed and accuracy in our task. We will look into recent work on regularizing ODEs to be easier to solve [13][17]. Another issue of regular Neural ODE is that it can only learn smooth deformations, we might need to augment our stage
Table 4. Neural Intensity Sampler vs. Linear Interpolation

| Iters / Model | DSC (↑)       | J<0 (↓)       |
|---------------|--------------|--------------|
|               | DDS  | DDS<sup>a</sup> | DDC | DDC<sup>a</sup> | DDS  | DDS<sup>a</sup> | DDC | DDC<sup>a</sup> |
| 100           | 0.5510 | 0.5384        | 0.5512 | 0.5385 | 2.10e-07 | 1.85e-07 | 9.64e-06 | 2.54e-06 |
| 300           | 0.5848 | 0.5821        | 0.5908 | 0.5869 | 1.79e-05 | 5.46e-06 | 2.01e-05 | 1.15e-05 |
| 600           | 0.5973 | 0.5940        | 0.6025 | 0.5988 | 4.49e-05 | 2.50e-05 | 3.06e-05 | 2.00e-05 |
| 900           | 0.6018 | 0.5991        | 0.6067 | 0.6031 | 6.45e-05 | 4.00e-05 | 3.70e-05 | 2.67e-05 |

Table 5. Optimization Iterations vs. Runtime

| Iters / Model | DS  | DC  | DDS  | DDC  |
|---------------|-----|-----|------|------|
| 100           | 9   | 44  | 96   | 326  |
| 300           | 27  | 84  | 288  | 594  |
| 600           | 54  | 144 | 576  | 996  |
| 900           | 81  | 204 | 864  | 1398 |

Our novel framework provides a solid benchmark for pairwise medical image registration using neural fields and many widely applied registration techniques such as symmetric mapping [3] can be easily incorporated into our framework to further improve our algorithm. Moreover, from the perspective of neural fields, there are three major directions to extend our framework. First, like DeepSDF [24], our framework can utilize a 'image pair codes' embedding to represent multiple deformations between image pairs from a training set. To optimize the deformation field between two test image, we don’t need to update all parameters of the neural field but only the 'image pair code'. So that the optimization speed can be significantly shorten and the prior information of image pairs can be implicitly learned by the neural field. Second, spatially-adaptive coordinate sampler [16] which enables our model to better fit a wide range of frequencies without any domain specific preprocessing. Also, we can define a set of voxel-bounded neural fields organized in a sparse voxel octree to model local properties in each cell to capture more detailed deformations [19]. Third, further improve the neural representation of medical volumes so that the registration model can fully defined on the continuous space.
Figure 7. Velocity Field Integration

Figure 8. Qualitative comparison between results generated by different MIRNF frameworks on Mindboggle dataset.