# IMAGE REGISTRATION IS A GEOMETRIC DEEP LEARNING TASK

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Data-driven deformable image registration methods predominantly rely on operations that process grid-like inputs. However, applying deformable transformations to an image results in a warped space that deviates from a rigid grid structure. Consequently, data-driven approaches with sequential deformations have to apply grid resampling operations between each deformation step. While artifacts caused by resampling are negligible in high-resolution images, the resampling of sparse, high-dimensional feature grids introduces errors that affect the deformation modeling process. Taking inspiration from Lagrangian reference frames of deformation fields, our work introduces a novel paradigm for data-driven deformable image registration that utilizes geometric deep-learning principles to model deformations without grid requirements. Specifically, we model image features as a set of nodes that freely move in Euclidean space, update their coordinates under graph operations, and dynamically readjust their local neighborhoods. We employ this formulation to construct a multi-resolution deformable registration model, where deformation layers iteratively refine the overall transformation at each resolution without intermediate resampling operations on the feature grids. We investigate our method's ability to fully deformably capture large deformations across a number of medical imaging registration tasks. In particular, we apply our approach (GeoReg) to the registration of inter-subject brain MR images and inhale-exhale lung CT images, showing on par performance with the current state-of-the-art methods. We believe our contribution open up avenues of research to reduce the black-box nature of current learned registration paradigms by explicitly modeling the transformation within the architecture.

## 1 INTRODUCTION

Image registration is an indispensable tool in medical image analysis that aligns anatomically or functionally corresponding regions across images, often from different modalities and time points (Sotiras et al., 2013). In particular, deformable registration aims to estimate a non-linear transformation that maps the *source* image to the coordinate space of the *target* image. Since the advent of deep learning, data-driven methods have been proposed (Haskins et al., 2020; Xiao et al., 2021) to leverage learned transformation priors over an image cohort, reducing the search space of plausible transformations.

**Single-stream approaches.** Since images are usually represented as grids of pixels, data-driven approaches typically employ convolutional kernels to model transformations. Early works have commonly adopted a simplistic approach of concatenating the source and target images as input channels to a convolutional network (Balakrishnan et al., 2019; Dalca et al., 2018; Mok & Chung, 2020a; Qiu et al., 2021; Zhao et al., 2019). However, this concatenation produces different inputs to a network on every possible misalignment of the counterpart image. This subsequently causes the feature extraction to learn distinct representations throughout the network on each possible misalignment, increasing the task's complexity.

A common necessary preprocessing technique employed to mitigate this issue involves an exhaustive search for an initial affine alignment. This reduces the degrees of freedom in the transformation parameters by guaranteeing that similar features are captured in a consistent spatial context, thus reducing the range of representations experienced by the network. Recent works combat the misalignment-

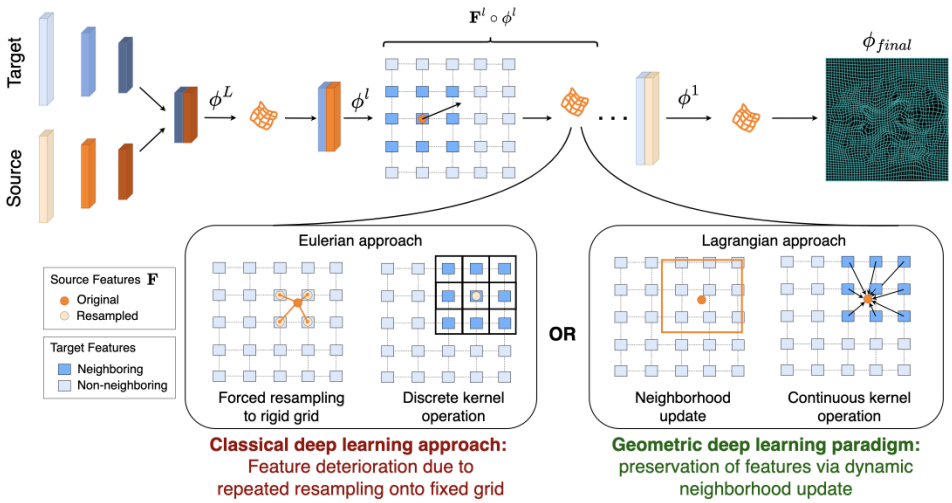


Figure 1: A classical deep learning setting operates on an Eulerian framework, where an implicit grid is required to predict a transformation. After a deformation is applied, this forces resampling to perform subsequent predictions. Our approach employs geometric deep learning to formulate the registration task as a grid-independent process using a Lagrangian reference frame.

dependent complexity by incorporating transformer layers throughout the network (Chen et al., 2022; 2023; Liu et al., 2022; Meng et al., 2022; Wang et al., 2023; Zhu & Lu, 2022). This enables greater flexibility in the feature extraction process as the transformer layer's attention mechanism is able to establish non-local spatial relationships at the cost of increased learnable parameters. Similarly, cascading approaches have shown increased accuracy by recovering the misalignment progressively, modeling the transformation as a sequence of deformations (Hu et al., 2022; Sandkühler et al., 2019; Zhao et al., 2019).

**Dual-stream approaches.** A commonly adopted technique to avoid concatenating the images at the input is using dual-stream encoders (Hu et al., 2022; Wang et al., 2023; Kang et al., 2022; Liu et al., 2022; Meng et al., 2022). This approach utilizes two separate encoders that individually extract features from source and target images, allowing for misalignment-independent representations throughout most of the network. Moreover, multi-resolution methods (Hu et al., 2022; Kang et al., 2022; Meng et al., 2022; Mok & Chung, 2020a; Wang et al., 2023) estimate the transformation at multiple levels during the decoding process in a coarse-to-fine fashion. This proves to be a strong architectural prior towards capturing large transformations in a parameter-efficient manner.

Despite their effectiveness, the layers of these architectures remain strictly confined to grid-structured features due to their dependence on discrete kernel operations. Consequently, as warping operations are applied to the feature grids in the decoding process, the space needs to be resampled to adhere to the fixed grid positions (Figure 1). Although negligible on the pixel level, resampling operations become increasingly inaccurate on sparse, high-dimensional feature spaces. This issue, termed *the curse of dimensionality*, causes naive interpolation to deteriorate the quality of features being propagated to later sections of the prediction process Verleysen & François (2005).

**Reference frames.** Current data-driven registration methods predominantly rely on an Eulerian (Batchelor, 1967) frame of reference for modeling deformation fields, whereby the deformation field is observed at specific locations in space. Conversely, the Lagrangian (Batchelor, 1967) specification is an alternative choice of modeling motion that tracks the positions of infinitesimal parcels (Brun et al., 2010; Lester et al., 1999; Thirion, 1998; Vercauteren et al., 2009). These concepts carry striking similarities with recent research in the field of geometric deep learning, where the motion of sparse point-like objects is modeled using learned functions (Farahani & Hamker, 2022; Fuchs et al., 2020; Kashefi et al., 2021). This design choice obviates the aforementioned resampling issues by not being confined to a grid-based frame of reference, removing the costly memory requirement of tracking an arbitrarily precise voxelized volume. While geometric deep learning has been used in certain contexts for point cloud and cortical surface registration (Hansen & Heinrich, 2021a;b; Suliman et al.,

2022; Shen et al., 2021; Hoopes et al., 2022), to the best of our knowledge, no previous work offers a framing of *deformable* image registration within the geometric deep learning paradigm.

**Contributions.** In this work, we propose a novel paradigm for data-driven image registration by viewing the deformation modeling process through the lens of geometric deep learning. We formulate the task as a multi-scale process of deformation operations, where feature grids are modeled under a Lagrangian framing of free-floating nodes influenced by neighborhood interactions. Unlike existing approaches, our method explicitly models coordinates and features independently, performing node-wise operations using continuous learnable kernels. This formulation enables us to completely avoid grid-based constraints on inter-node structure, removing the requirement for intermediate warping operations between transformations.

Our contributions can be summarized as follows:

- We establish a mathematical foundation to formulate deformable registration in a continuous domain, avoiding the need for interpolation in the feature space. We achieve this under a Lagrangian reference frame utilizing the geometric deep-learning paradigm.
- We propose a data-driven form of interpolation demonstrating local support, which facilitates multi-scale deformation modeling by learning to propagate deformations across resolutions.
- We demonstrate the effectiveness of our formulation by reporting improvements over current state-of-the-art deformable registration in the context of medical imaging and showing our model's ability to recover large deformations. We make our code publicly available[1].

## 2 Method

In this section, we first formally establish the limitations imposed on deformable image registration by the grid constraints of Eulerian frameworks. Afterwards, we establish a Lagrangian formulation that does not make any grid assumptions (section 2.1). Within this context, we highlight the advantages offered by geometric deep learning in modeling deformations as interactions between free-floating features (section 2.2). Next, we propose a data-driven form of local interpolation, which facilitates multi-scale deformation modeling by learning to propagate deformations across resolutions (section 2.3). Finally, we combine these ideas to construct an end-to-end trainable neural network capable of learning deformable registration in continuous domains in a coarse-to-fine fashion (section 2.4).

### 2.1 Deformation without grid constraints

An image $I$ can be interpreted as a finite grid of measurements $I \in \mathbb{R}^{H \times W \times D}$ which represents a discrete subset of a continuous domain $\Omega$, and where $H, W, D$ are the spatial dimensions of the voxel grid in the 3-dimensional case.

Given a target $T$ and a source $S$ image, deformable image registration aims to find an optimal spatial deformation field $\phi^* = \arg\min_\phi \mathcal{J}(T, S, \phi)$, with $\phi : \mathbb{R}^n \to \mathbb{R}^n$, such that the transformed source image $S \circ \phi$ is most similar to the target image $T$. As such, the overall objective $\mathcal{J}$ is defined as:

$$\mathcal{J}(T, S, \phi) = \mathcal{D}(T, S \circ \phi) + \lambda \mathcal{R}(\phi), \tag{1}$$

where $\mathcal{D} : \Omega \times \Omega \to \mathbb{R}^n$ is an image dissimilarity measure responsible for driving the deformation, and $\mathcal{R}$ is a smoothness regularization on the transformation whose magnitude is weighted by $\lambda$.

Data-driven methods implement learnable functions responsible for modeling a deformable transformation $\phi$ as a neural network $\tau_\theta(T, S)$ parametrized by a set of learnable weights $\theta$. These approaches model the transformation based on spatial features $\boldsymbol{F}^S$ and $\boldsymbol{F}^T$ extracted from $S$ and $T$.

Images are usually represented as grids of nodes containing pixel information. By making relative positions between neighboring nodes constant across the structure, grid representations carry implicit assumptions about the homogeneity of the structure. As such, attempting to construct a neural function $\tau$ by learning grid-based operations (such as convolutions with discrete kernels) on the

[1] https://anonymous.4open.science/r/GeoReg-D567

features $\boldsymbol{F}$ of a given image may appear enticing. However, these grid-reliant architectures generally confine the transformation prediction to a grid, resulting in an Eulerian framework whereby the deformation field is only defined on the specific locations in space defined by the grid. As spatial transformations are applied to the source domain $S$, the rigid assumption on grid structure requires a resampling of the feature space to estimate the feature values at the grid positions, which may result in deteriorated features.

Instead of relying on grid representations, we propose to explicitly represent the source domain under a Lagrangian framework via an unstructured set of feature-coordinate tuples $S = (\boldsymbol{F}^S, \boldsymbol{X}^S)$, where $\boldsymbol{F}^S \in \mathbb{R}^{|S| \times d}, \boldsymbol{X}^S \in \mathbb{R}^{|S| \times n}$. Now, each node $s \in S$ can be regarded as a discrete observation of a feature $\mathbf{f}^s$ embedded in Euclidean space at coordinate $\mathbf{x}^s$. Here, the feature component $\mathbf{f}^s$ of a given node $s$ may represent anything from gray-scale intensities ($d = 1$) to higher dimensional feature descriptors such as those extracted by a feature encoder. This formulation allows us to model the deformation process as a function $\tau$ acting on the coordinate component $\mathbf{x}^s$ of each $s \in S$ without requiring the feature $\mathbf{f}^s$ to be modified. This Lagrangian framework is especially enticing to coarse, high-dimensional feature spaces that would otherwise suffer from the curse of dimensionality under interpolation operations (Bronstein et al., 2021).

### 2.2 Deformation modeling function $\boldsymbol{\tau}$

Formulating the source domain $S$ as a set of nodes with real-valued coordinates, with no necessity of adhering to a rigid grid, requires a generalized form of learned functions that can handle continuous domains. Precisely, we need to define a deformation function $\tau(\cdot)$ that is able to act node-wise on the set of source nodes $s \in S$, while being capable of handling neighborhoods of target nodes $T_{\mathcal{N}_s} \subset T$ with arbitrary real-valued coordinates relative to a given $s$. This formulation naturally leads us to the realm of geometric deep learning, where graph neural networks are leveraged to model learned functions on geometric graph structures.

We begin by using the previously introduced notation of feature-coordinate pairs for the nodes of the target domain $T = (\boldsymbol{F}^T, \boldsymbol{X}^T)$. Using the source $S$ and target $T$ sets, we define the domain of the deformation function $\tau$ as a directional graph $\mathcal{G}^\tau = (\boldsymbol{A}^\tau, [\boldsymbol{F}^T, \boldsymbol{F}^S], [\boldsymbol{X}^T, \boldsymbol{X}^S])$ describing how each of the source nodes $s \in S$ interacts with the target domain $T$. The adjacency matrix $\boldsymbol{A}^\tau \in \{0, 1\}$ describes the presence or absence of edges between all node pairs. In practice, since we only want to model how a given source node $s$ should deform given its local neighborhood in the target domain, we only need to model one quadrant of the full adjacency matrix $\boldsymbol{A}^\tau \in \{0, 1\}^{T \times S}$. A row in $\boldsymbol{A}^\tau$ represents the neighborhood $T_{\mathcal{N}_s}$ of a source node $s$ within the target domain $T$ by defining which subset of target nodes contain directional edges to $s$, such that $T_{\mathcal{N}_s} = \{t \mid \forall t \in \boldsymbol{A}^\tau_{[s]}, \boldsymbol{A}^\tau_{[s,t]} = 1\}$. The values of a row $\boldsymbol{A}^\tau_{[s]}$ are computed using k-nearest neighbors based on $|\mathbf{x}^t - \mathbf{x}^s|$ node distances.

Given a graph $\mathcal{G}^\tau$, we can define a generalized framing of a convolution by modeling a learnable kernel as a continuous function $\psi$. This formulation can operate on both grid and non-grid layouts alike. When centered on a node $s$, the function $\psi$ computes an activation to a set of neighboring target nodes $T_{\mathcal{N}_s}$ based on their features $\mathbf{f}^t$ and relative coordinates to the source node $(\mathbf{x}^t - \mathbf{x}^s)$:

$$\mathbf{f}' = \frac{1}{|T_{\mathcal{N}_s}|} \sum_{t \in T_{\mathcal{N}_s}} \psi\left(\mathbf{f}^t, \mathbf{f}^s, \left(\mathbf{x}^t - \mathbf{x}^s\right)\right) \tag{2}$$

The function $\psi$ is typically implemented as a learnable linear projection with a non-linearity. The output feature of this convolution operation can then be further projected into the vector of size $n$ to predict a deformation $\phi \in \mathbb{R}^n$ for that given node. This establishes a basis for learning a continuous deformation model $\tau(\mathcal{G}^\tau)$ that can be applied to a continuous domain in order to predict deformations $\phi$ for any node $s$.

This process can be iteratively performed, creating a chain of deformations $\boldsymbol{X}^S \circ \tau(\mathcal{G}^\tau_1) \circ ... \circ \tau(\mathcal{G}^\tau_N)$ that refine the transformation prediction in a cascading fashion. As the source nodes are transformed relative to $T$, the graph $\mathcal{G}^\tau$ can simply be recomputed using the newly transformed coordinates (see Figure 2 (iv)). Unlike other state-of-the-art cascading approaches, our grid-independent formulation requires no intermediate resampling-to-a-grid operations, maintaining feature integrity along the deformation chain. We refer the reader to the pseudocode in Appendix B1 for a detailed overview.

### 2.3 MULTI-RESOLUTION INTERPOLATION FUNCTION $\boldsymbol{\delta}$

The optimization process of real-world registration tasks is highly non-convex. A prevalent strategy in literature to overcome local minima when dealing with complex transformations is the usage of a multi-resolution strategy. In line with this established paradigm, we explore the application of $\tau$ deformation functions across entire feature pyramids in coarse-to-fine multi-resolution settings. In this section, we aim to define a function $\delta(\cdot)$ capable of chaining deformations in a coarse-to-fine fashion across resolutions without feature interpolation.

#### 2.3.1 LOCALLY-WEIGHTED INTERPOLATION

A commonly adopted technique in parametric image registration involves predicting deformations at a coarse spacing and interpolating to the desired resolution via continuous mapping functions. Generally, these mappings are formulated in the context of a set of control points $C$ exerting influences on the interpolation at a given point in space via local basis functions. Particularly in the case of b-spline basis functions, the interpolation process exhibits the property of local support, implying that a small, localized change has a restricted impact and does not influence the entire domain.

The transformation $\phi$ at an arbitrary point in space $i$ with coordinates $\mathbf{x}^i$ is the resulting interpolation of the transformation values $\phi^c$ of its neighboring control points $c \in \mathcal{N}^C$. This interpolation is weighted using basis functions $v$, based on relative positions between the given point $i$ and each control point $c$:

$$\phi(\mathbf{x}^i) = \sum_{c \in C_{\mathcal{N}_i}} \overbrace{v\left(\mathbf{x}^c - \mathbf{x}^i\right)}^{\text{weight coefficient}} \phi^c \tag{3}$$

This concept of locally weighting a transformation, based on relative location to control points, serves as a powerful heuristic for introducing local support. However, we argue that making the interpolation mechanism aware of image features is the key to building improved interpolation functions.

#### 2.3.2 CROSS-ATTENTION AS DATA-DRIVEN INTERPOLATION

The attention mechanism has been applied to illustrate a more general version of the convolution operation (Bronstein et al., 2021) described in Eq. (2). In the context of registration, graph convolutions already display desired local properties by restricting message-passing within local neighborhoods. The attention operation extends this principle by dynamically "masking out" irrelevant neighboring nodes. Unlike the convolution's simple uniformly-weighted aggregation of neighbors' responses, the attention mechanism allows a node to compute a form of learned weighted averaging based on its neighbors' features and relative positions.

$$\mathbf{f}' = \sum_{c \in C_{\mathcal{N}_i}} \overbrace{a(\mathbf{f}^c, \mathbf{f}^i, \left(\mathbf{x}^c - \mathbf{x}^i\right))}^{\text{weight coefficient}} \psi\left(\mathbf{f}^c, \mathbf{f}^i, \left(\mathbf{x}^c - \mathbf{x}^i\right)\right) \tag{4}$$

where $C_{\mathcal{N}_i}$ is the neighborhood of control points to node $i$. When nodes $i$ and $c$ belong to different domains (e.g., different images or resolution levels), the operation described in Eq. (4) is referred to as *cross-attention*. Here, the attention function $a$ is constrained to be in the range $[0, 1]$ by applying a softmax operation over all neighboring control points such that $\sum_{c \in C_{\mathcal{N}_i}} a(\cdot) = 1$. For further implementation details, we refer the reader to section 2.4.

The concept of attention as dynamically weighting neighboring nodes as outlined in Eq. (4) offers strong similarities to the principles of parametric registration methods outlined in Eq. (3). Similarly to how parametric interpolation uses a preset weighting function $v\left(\mathbf{x}^c - \mathbf{x}^i\right)$ on neighboring control points, local attention uses a learned weighting function $a(\mathbf{f}^i, \mathbf{f}^c, \left(\mathbf{x}^c - \mathbf{x}^i\right))$. In the local attention setting, since a given node only interacts with its spatially restricted neighborhood (and not with the entire space), a localized change does not affect the entire domain, effectively offering properties of local support. The benefit of the attention mechanism here is the ability to condition the weighting coefficients not only on relative coordinates but also on the learned features present in the operation.

While not a direct form of deformation interpolation, local cross-attention offers a way to refine a node's location given its current neighborhood of control points. By placing a node near a set of

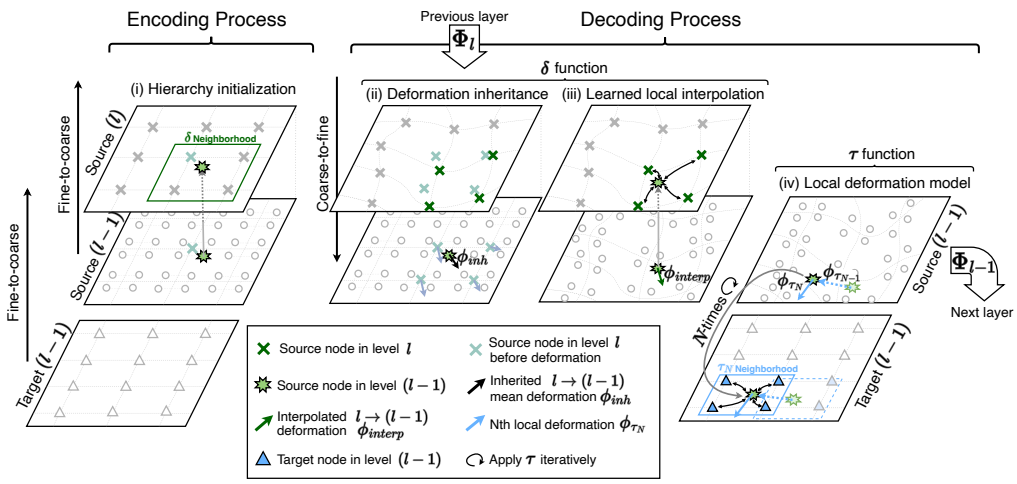


Figure 2: Architectural outline of a given resolution layer. (i) During feature extraction, pooling operations track cross-resolution hierarchies. (ii) During decoding, each child node inherits and further refines its transformation by performing local attention over its neighborhood of parents. (iii) Deformations are predicted by performing local attention over local target nodes. This operation can be iteratively performed an arbitrary number of times by dynamically updating a source node's current neighborhood in the target domain.

control points, the attention mechanism produces a weighted feature vector, which can be subsequently projected into the vector of size $n$ to predict a learned deformation $\phi \in \mathbb{R}^n$. This gives us a building block to define a local function $\delta(\cdot)$ that interpolates a deformation through local cross-attention on a continuous domain.

As depicted in Figure 2 (ii) & (iii), this component enables us to carry deformations across feature pyramid levels in the multi-scale decoding process, further circumventing the need for grid-based resampling operations.

### 2.3.3 Hierarchy graphs as interpolation domains

Our aim is to define a hierarchical process by which a finer resolution level inherits and subsequently refines the transformation of a coarser level. As such, we refer to control nodes at level $l$ as *parents* and the nodes at the following level $l-1$ as *children*.

To define the domain of the cross-resolution interpolation function $\delta(\cdot)$, we need to establish a graph representing parent-child connections. Concretely, given a set of control nodes $S^l$ at the $l$-th pyramid level and a set of children nodes $S^{l-1}$, we define a directional graph $\mathcal{G}^l = (\boldsymbol{A}^l, [\boldsymbol{F}^l, \boldsymbol{F}^{l-1}], [\boldsymbol{X}^l, \boldsymbol{X}^{l-1}])$. Here, $\boldsymbol{A}^l$ is a static matrix exclusively containing edges connecting children nodes to their k-nearest neighboring parent nodes.

**Hierarchy graph initialization.** As depicted in Figure 2 (i), we use the feature extraction process at the encoder to define parent-child hierarchies based on pooling operations. The feature extraction process establishes a hierarchy of children-parent nodes $S^{\{1,\ldots,L\}}$, which all live in a combined Euclidean space across $L$ resolution levels. A parent node at level $l$ derived from a region of children nodes at level $l-1$ has its position initialized at the center of the pooling window. This hierarchy establishes the cross-resolution neighborhoods that facilitate the interpolation operation of each given child node $s \in S^{l-1}$ during the decoding process.

**Transformation inheritance.** As depicted in Figure 2 (ii), an initial transformation is inherited by each child node at level $l-1$ based on the average transformation of their neighborhoods at resolution $l$. This rough initial approximation significantly reduces the remaining transformation component left to be modeled by the feature-aware learned interpolation. We refer the reader to the pseudocode in Appendix B2 for a detailed overview of this process.

### 2.4 IMPLEMENTATION DETAILS

**Image encoder.** Since our approach requires the source and target features to interact, it is necessary to implement a dual-stream encoder architecture. As a result, feature pyramids are extracted for the source $S$ and target $T$ images independently using the same set of weights. We refer the reader to Appendix A for a schematic of the overall model. The encoder consists of two convolutional residual blocks per resolution followed by pooling layers. The encoder is composed of 6 layers, each made up of two residual blocks, each with $[16, 32, 32, 64, 64, 128]$ channels. Average pooling $[2 \times 2 \times 2]$ operations are applied between each encoder layer.

**Deformation decoder.** The feature pyramid is then decoded in a coarse-to-fine fashion across the 6 resolutions. Across our experiments, we find that local deformations are sufficiently well-modeled at coarser resolutions in the decoder. Estimating the majority of the transformation at coarser levels reduces the registration burden of finer resolution layers, as only smaller local deformations are left to be recovered (see Appendix F for further detail). These insights allow us to refrain from applying $\tau$ at the finest layers where resolutions are the largest, massively decreasing our memory footprint.

**Memory efficient neighborhood computation.** While the methodology outlined in sections 2.2 and 2.3 are formalized using graph notation, the grid structure of our data allows us to design highly memory efficient implementations of $\tau$ and $\delta$ layers. Finding nearest neighbors in sparse data is the biggest memory bottleneck due to having to compute $O(N^2)$ distance calculations relative to the number of nodes $N$. Our convolutional encoder providing us with feature grids, lets us use grid-unfolding operations to find the nearest neighbors in a $(k_x, k_y, k_z)$ kernel around a central node. The $\tau$ function's neighborhood computations are performed by first unfolding the target domain into all its possible neighbourhoods. Then, we can index a source node's corresponding target neighborhood by mapping the current source coordinates into the index space of the target grid and rounding to the closest integer. Similarly, the $\delta$ function applies a repeated interleaving operation to upsample the parent unfolded neighborhoods into the same dimensions as the children grid. This is all implemented using standard built-in Pytorch functions that allow for efficient GPU parallelism. Wherever possible, we make use of pointers to the original data structures for minimal memory footprints. We refer readers to Appendix D for an overview on VRAM requirements of various registration baselines.

**Attention layers.** The attention mechanism described in Eq. (4) is implemented using standard transformer layers. The function takes the following matrix form:

$$\mathbf{f}' = \text{softmax}\left(\frac{Q\left(\mathbf{f}^i\right) \cdot K(\mathbf{F}_{\mathcal{N}_i}, \left(\mathbf{X}_{\mathcal{N}_i} - \mathbf{x}^i\right))^\top}{\sqrt{d}}\right) V(\mathbf{F}_{\mathcal{N}_i}, \left(\mathbf{X}_{\mathcal{N}_i} - \mathbf{x}^i\right)) \tag{5}$$

Here, $V$ is the *value* matrix, playing an equivalent role to the function $\psi$ in Eq. (2) and (4). The attention scores are computed by taking the dot product *query* $Q$ and *key* $K$ vectors. We represent $Q$, $K$, and $V$ as functions to indicate the presence of positional embedding steps, whereby information about the relative locations of neighboring nodes is encoded into the node's feature vector. We use Fourier features ((Tancik et al., 2020)) as our choice of embedding function. The concept of an input set of nodes computing attention scores to a different set of nodes is referred to as *cross-attention*.

**Hyperparameters.** Tuning neighborhood sizes required striking a balance between receptive field width and memory limitations. For $\tau$, we selected neighborhoods of $5^3$ for coarser resolutions (to allow for wide receptive fields), while finer resolutions used $3^3$ neighborhoods to reduce memory footprint. Neighborhoods in $\delta$ always used the closest $3^3$ neighboring parent nodes. To calculate the loss at each resolution level, we employ normalized cross-correlation (NCC) as the dissimilarity metric. Furthermore, we utilize bending energy (Rueckert et al., 1999) as a regularizer to ensure a smooth final transformation at each resolution. The approach is trained end-to-end using the ADAM optimizer with a $10^{-4}$ learning rate for a maximum of $1000$ epochs. Model training was carried out on an NVIDIA A40 GPU with 40GB VRAM over the course of 3 days. For further parameter details, we refer to our repository.

# 3 Results and Discussion

## 3.1 Datasets, Baselines and evaluation metrics

We evaluate our work using the CamCAN T1w-T1w and T1w-T2w brain datasets (Shafto et al., 2014; Taylor et al., 2017) and the publicly available benchmark National Lung Screening Trial (NLST) dataset (team, 2011) from the Learn2Reg challenge 2022 (Hering et al., 2022). We refer the reader to section C of the Appendix for more information about the data, the pre-processing steps, the segmentation labels, and the key-point extraction.

We compare our method (GeoReg) against several conventional iterative methods and learning-based image registration models. Regarding the iterative optimization methods, we choose from the Medical Image Registration ToolKit (MIRTK) (Schuh et al., 2014), a widely-used free-form deformation (FFD) iterative optimization method that supports multi-resolution and parametric b-spline-based registration. From the ANTs (Avants et al., 2009) framework we use the widely used symmetric diffeomorphic alorithm SyN (Avants et al., 2008). Additionally, we compare against a Large deformation diffeomorphic metric mapping (LDDMM (Beg et al., 2005)) algorithm implemented in PyTorch [2]. Our learning-based baselines are comprised of Voxelmorph (Balakrishnan et al., 2019), a single-stage CNN, LapIRN (Mok & Chung, 2020b) a multi-resolution registration CNN that aims to capture large deformations in a coarse-to-fine manner, Transmorph (Chen et al., 2022) that uses a SwinTransformer-based encoder, Recursive Cascaded Networks (RCN) (Zhao et al., 2019) which estimates the deformation progressively using a cascading CNN architecture, the dual-stream pyramid registration network (D-PRNet) (Kang et al., 2022) that gradually refines the multi-level predicted deformation fields in a coarse-to-fine manner via sequential warping, and FourierNet (Jia et al., 2023) that learns a low-representation displacement filed in a band-limited Fourier domain and then uses a model-driven decoder to obtain the dense, full-resolution displacement field. To ablate the contribution of the proposed interpolation mechanism ($\delta$) on top of our multi-resolution $\tau$ design, we replace the proposed learned interpolation component ($\delta$) with bilinear feature warping. In the following tables, we denote this ablation baseline as "feat. warp".

The accuracy of the registration is evaluated using the segmentation metrics Dice Similarity Coefficient (DSC) and 95th percentile Hausdorff distance (HD95). For the NLST dataset, we additionally report the target registration error (TRE) between landmarks. In the synthetic deformation experiments, we also report the average end-point error (AEE) to the ground truth deformation. We calculate the percentage of points with a negative Jacobian determinant $|\nabla\phi| < 0$, indicating the extent of space folding, to measure the regularity of the transformation.

### 3.1.1 Experiment 1: Large intra-subject synthetic transformations

First, we begin by investigating how our method captures varying kinds of large synthetic deformations without any form of affine registration preprocessing. We create a dataset of intra-subject brain pairs with varying ranges of non-rigid deformations comprised of a combination of an affine and Brownian noise components. Although not guaranteed to produce deformations equivalent to real-world medical registration tasks, this experiment allows us to generate ground-truth deformations serving as a useful proof-of-concept to better evaluate recovery of large misalignments. First, a base component of fractal Brownian deformation is applied, followed by randomly uniformly sampled rotations, scaling, and translations along each dimension (see displacement field in Figure 3). In addition to the deformed image, we obtain a ground truth deformation field, which we use to quantitatively assess a method's ability to deformably recover large misalignments.

The results reported in Table 1 demonstrate that our model consistently outperforms other baselines while producing the lowest amount of spatial folding. While other models struggle with large deformations, our geometric registration method is capable of fully deformably capturing the overall global transformation while still being able to model local deformations (see Figure 3).

### 3.1.2 Experiment 2: Deformable transformations on pre-aligned images

In practice, it is common practice to pre-align scans in roughly the same coordinate system using affine registration, targeting the larger components of the transformation. This initial alignment

[2] https://github.com/brianlee324/torch-lddmm

Table 1: Quantitative results for intra-subject deformable registration using non-rigid synthetic deformations (multi-resolution Brownian) alongside varying degrees of uniformly-sampled rotations, scalings, and translations. Lowest setting in Brownian experiment row is used as default across all other rigid rows. Experiments consist of 100 subjects, each sampled using 10 different synthetic deformations. The performance of GeoReg with bilinear feature warping instead of a learned interpolation component $\delta$ is shown under 'feat. warp'.

| | # Param | HD95 ↓ | $AEE_{\phi_{GT}}$ ↓ | Folding (%) ↓ | HD95 ↓ | $AEE_{\phi_{GT}}$ ↓ | Folding (%) ↓ | HD95 ↓ | $AEE_{\phi_{GT}}$ ↓ | Folding (%) ↓ |
|---|---|---|---|---|---|---|---|---|---|---|
| **Brownian** | | Up to 16.41 pixels per axis (Default) | | | Up to 25.25 pixels per axis | | | Up to 33.98 pixels per axis | | |
| Affine | - | $4.695 \pm 0.979$ | $1.813 \pm 0.316$ | - | $4.821 \pm 0.992$ | $1.933 \pm 0.336$ | - | $5.118 \pm 1.113$ | $2.063 \pm 0.345$ | - |
| MIRTK | - | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ |
| VoxelMorph | 320 k | $1.656 \pm 0.159$ | $3.561 \pm 0.245$ | $0.003 \pm 0.002$ | $3.672 \pm 0.791$ | $8.323 \pm 1.282$ | $0.132 \pm 0.161$ | $6.199 \pm 1.910$ | $11.466 \pm 1.502$ | $0.249 \pm 0.203$ |
| TransMorph | 46.8 M | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ |
| D-PRNet | 1.2 M | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ |
| RCN | 282 M | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ |
| **Ours** (feat. warp) | 1.5 M | $2.621 \pm 0.502$ | $1.637 \pm 0.161$ | $0.000 \pm 0.000$ | $3.023 \pm 0.557$ | $1.820 \pm 0.156$ | $0.000 \pm 0.000$ | $3.618 \pm 0.547$ | $1.770 \pm 0.191$ | $0.000 \pm 0.000$ |
| **Ours** (GeoReg) | 1.7 M | $1.347 \pm 0.397$ | $1.328 \pm 0.152$ | $0.000 \pm 0.000$ | $1.403 \pm 0.360$ | $1.328 \pm 0.135$ | $0.000 \pm 0.000$ | $1.423 \pm 0.369$ | $1.483 \pm 0.195$ | $0.000 \pm 0.000$ |
| **Rotation + Brownian** | | $\pm 12.75°$ per axis | | | $\pm 22.5°$ per axis | | | $\pm 45.0°$ per axis | | |
| Affine | - | $4.573 \pm 0.291$ | $3.686 \pm 0.098$ | - | $4.599 \pm 0.331$ | $3.682 \pm 0.109$ | - | $4.600 \pm 0.369$ | $3.809 \pm 0.110$ | - |
| MIRTK | - | $1.000 \pm 0.000$ | $3.097 \pm 1.152$ | $0.087 \pm 0.212$ | $3.406 \pm 4.144$ | $4.213 \pm 0.953$ | $0.0800 \pm 0.163$ | $8.000 \pm 7.669$ | $4.901 \pm 0.879$ | $0.843 \pm 0.122$ |
| VoxelMorph | 320 k | $1.816 \pm 0.298$ | $6.673 \pm 1.054$ | $0.034 \pm 0.020$ | $3.474 \pm 0.713$ | $13.591 \pm 2.318$ | $0.097 \pm 0.041$ | $8.997 \pm 2.353$ | $27.090 \pm 5.130$ | $0.292 \pm 0.077$ |
| TransMorph | 46.8 M | $\mathbf{1.057 \pm 0.073}$ | $5.087 \pm 0.775$ | $3.030 \pm 0.514$ | $\mathbf{1.420 \pm 0.385}$ | $11.334 \pm 2.999$ | $3.560 \pm 0.414$ | $5.747 \pm 2.289$ | $26.394 \pm 5.301$ | $4.012 \pm 0.329$ |
| D-PRNet | 1.2 M | $1.557 \pm 0.367$ | $7.002 \pm 1.202$ | $1.422 \pm 0.188$ | $3.580 \pm 1.082$ | $14.058 \pm 2.896$ | $1.629 \pm 0.265$ | $9.200 \pm 2.444$ | $28.278 \pm 5.769$ | $2.192 \pm 0.313$ |
| RCN | 282 M | $1.364 \pm 0.130$ | $4.262 \pm 0.518$ | $3.640 \pm 0.698$ | $1.902 \pm 0.218$ | $11.082 \pm 2.042$ | $3.945 \pm 0.558$ | $4.951 \pm 1.777$ | $26.537 \pm 6.120$ | $4.029 \pm 0.505$ |
| **Ours** (feat. warp) | 1.5 M | $2.068 \pm 0.484$ | $1.585 \pm 0.312$ | $0.000 \pm 0.000$ | $2.620 \pm 1.358$ | $1.989 \pm 0.753$ | $0.000 \pm 0.000$ | $2.818 \pm 0.546$ | $2.477 \pm 0.928$ | $0.000 \pm 0.000$ |
| **Ours** (GeoReg) | 1.7 M | $1.520 \pm 0.332$ | $\mathbf{1.511 \pm 0.260}$ | $0.000 \pm 0.000$ | $1.630 \pm 0.415$ | $\mathbf{1.604 \pm 0.363}$ | $0.000 \pm 0.000$ | $\mathbf{2.054 \pm 0.385}$ | $\mathbf{1.951 \pm 0.598}$ | $0.026 \pm 0.145$ |
| **Scaling + Brownian** | | $\pm 10\%$ of image size per axis | | | $\pm 30\%$ of image size per axis | | | $\pm 50\%$ of image size per axis | | |
| Affine | - | $4.529 \pm 0.368$ | $3.685 \pm 0.101$ | - | $4.891 \pm 0.431$ | $3.687 \pm 0.113$ | - | $5.138 \pm 0.515$ | $3.749 \pm 0.116$ | - |
| MIRTK | - | $1.000 \pm 0.000$ | $3.125 \pm 0.508$ | $0.039 \pm 0.002$ | $4.545 \pm 1.598$ | $4.837 \pm 7.788$ | $0.398 \pm 0.699$ | $8.345 \pm 1.828$ | $8.336 \pm 4.061$ | $2.355 \pm 1.385$ |
| VoxelMorph | 320 k | $1.706 \pm 0.167$ | $3.542 \pm 0.242$ | $0.002 \pm 0.002$ | $3.389 \pm 0.642$ | $7.074 \pm 1.045$ | $0.122 \pm 0.130$ | $7.592 \pm 2.113$ | $12.287 \pm 1.798$ | $0.228 \pm 0.116$ |
| Transmorph | 46.8 M | $1.074 \pm 0.079$ | $3.308 \pm 0.214$ | $1.536 \pm 0.213$ | $1.370 \pm 0.363$ | $6.384 \pm 0.635$ | $3.753 \pm 0.612$ | $3.154 \pm 1.531$ | $10.307 \pm 1.705$ | $5.086 \pm 0.621$ |
| D-PRNet | 1.2 M | $1.250 \pm 0.137$ | $3.598 \pm 0.270$ | $1.388 \pm 0.318$ | $2.225 \pm 0.347$ | $7.505 \pm 1.028$ | $3.200 \pm 0.445$ | $5.391 \pm 2.160$ | $11.986 \pm 2.402$ | $3.910 \pm 0.465$ |
| RCN | 282 M | $1.337 \pm 0.188$ | $3.307 \pm 0.181$ | $1.600 \pm 0.370$ | $2.593 \pm 0.252$ | $5.396 \pm 0.586$ | $4.642 \pm 0.806$ | $3.785 \pm 0.661$ | $7.834 \pm 1.276$ | $5.644 \pm 0.687$ |
| **Ours** (feat. warp) | 1.5 M | $1.910 \pm 0.355$ | $1.490 \pm 0.277$ | $0.000 \pm 0.000$ | $2.566 \pm 0.617$ | $2.073 \pm 0.925$ | $0.000 \pm 0.000$ | $2.961 \pm 0.796$ | $2.400 \pm 1.290$ | $0.000 \pm 0.000$ |
| **Ours** (GeoReg) | 1.7 M | $\mathbf{1.040 \pm 0.122}$ | $\mathbf{1.375 \pm 0.357}$ | $0.000 \pm 0.000$ | $\mathbf{1.274 \pm 0.312}$ | $\mathbf{1.714 \pm 0.904}$ | $0.000 \pm 0.000$ | $\mathbf{1.486 \pm 0.523}$ | $\mathbf{2.234 \pm 1.586}$ | $0.000 \pm 0.000$ |
| **Translation + Brownian** | | $\pm 10\%$ of image size per axis | | | $\pm 30\%$ of image size per axis | | | $\pm 50\%$ of image size per axis | | |
| Affine | - | $4.791 \pm 1.106$ | $2.092 \pm 0.362$ | - | $4.683 \pm 1.241$ | $2.076 \pm 0.386$ | - | $4.768 \pm 1.088$ | $2.025 \pm 0.497$ | - |
| MIRTK | - | $2.217 \pm 0.256$ | $4.680 \pm 1.643$ | $0.0302 \pm 0.172$ | $15.754 \pm 2.543$ | $5.602 \pm 3.420$ | $0.248 \pm 0.346$ | $25.357 \pm 12.085$ | $7.682 \pm 1.638$ | $2.102 \pm 0.123$ |
| VoxelMorph | 320 k | $3.468 \pm 0.529$ | $5.436 \pm 0.495$ | $0.033 \pm 0.034$ | $18.075 \pm 2.677$ | $16.027 \pm 1.977$ | $0.628 \pm 0.194$ | $31.645 \pm 4.145$ | $26.826 \pm 3.932$ | $1.479 \pm 0.395$ |
| TransMorph | 46.8 M | $1.641 \pm 0.385$ | $6.065 \pm 0.634$ | $2.601 \pm 0.421$ | $17.865 \pm 4.339$ | $20.212 \pm 3.201$ | $4.378 \pm 0.275$ | $40.148 \pm 7.138$ | $37.221 \pm 5.569$ | $5.920 \pm 0.147$ |
| D-PRNet | 1.2 M | $3.720 \pm 0.879$ | $6.045 \pm 0.751$ | $2.631 \pm 0.483$ | $5.477 \pm 0.692$ | $12.834 \pm 1.757$ | $5.556 \pm 0.428$ | $6.669 \pm 0.833$ | $17.522 \pm 2.954$ | $6.636 \pm 0.509$ |
| RCN | 282 M | $2.217 \pm 0.193$ | $3.559 \pm 0.163$ | $4.465 \pm 1.071$ | $4.632 \pm 0.352$ | $6.427 \pm 0.608$ | $6.491 \pm 0.883$ | $5.123 \pm 0.430$ | $10.281 \pm 1.381$ | $7.166 \pm 0.565$ |
| **Ours** (feat. warp) | 1.5 M | $2.193 \pm 0.334$ | $1.474 \pm 0.156$ | $0.000 \pm 0.000$ | $3.397 \pm 0.446$ | $1.949 \pm 0.189$ | $0.000 \pm 0.000$ | $4.605 \pm 2.827$ | $3.279 \pm 2.685$ | $0.000 \pm 0.000$ |
| **Ours** (GeoReg) | 1.7 M | $\mathbf{1.293 \pm 0.308}$ | $\mathbf{1.288 \pm 0.161}$ | $0.000 \pm 0.000$ | $\mathbf{1.603 \pm 0.329}$ | $\mathbf{1.434 \pm 0.205}$ | $0.000 \pm 0.000$ | $\mathbf{2.260 \pm 0.358}$ | $\mathbf{1.760 \pm 0.295}$ | $0.000 \pm 0.000$ |
| **Affine + Brownian** | | $\pm 12.75°$ Rot., $\pm 10\%$ Scale, $\pm 10\%$ Transl. | | | $\pm 22.5°$ Rot., $\pm 30\%$ Scale, $\pm 30\%$ Transl. | | | $\pm 45.0°$ Rot., $\pm 50\%$ Scale, $\pm 50\%$ Transl. | | |
| Affine | - | $4.646 \pm 1.339$ | $3.1062 \pm 1.009$ | - | $4.741 \pm 1.381$ | $4.612 \pm 0.294$ | - | $4.931 \pm 2.165$ | $6.388 \pm 3.766$ | - |
| MIRTK | - | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ |
| VoxelMorph | 320 k | $3.144 \pm 2.181$ | $4.464 \pm 1.065$ | $0.034 \pm 0.061$ | $23.367 \pm 12.870$ | $9.264 \pm 1.842$ | $0.983 \pm 0.714$ | $34.688 \pm 15.383$ | $13.938 \pm 5.523$ | $2.237 \pm 1.810$ |
| TransMorph | 46.8 M | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ |
| D-PRNet | 1.2 M | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ |
| RCN | 282 M | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ | $- \pm -$ |
| **Ours** (feat. warp) | 1.5 M | $2.384 \pm 0.491$ | $1.854 \pm 0.369$ | $0.000 \pm 0.000$ | $2.982 \pm 1.089$ | $3.031 \pm 1.016$ | $0.000 \pm 0.000$ | $4.109 \pm 2.371$ | $6.436 \pm 3.811$ | $0.001 \pm 0.001$ |
| **Ours** (GeoReg) | 1.7 M | $1.880 \pm 0.431$ | $1.614 \pm 0.239$ | $0.000 \pm 0.000$ | $10.253 \pm 5.109$ | $8.325 \pm 3.200$ | $0.000 \pm 0.000$ | $16.941 \pm 9.477$ | $18.312 \pm 8.686$ | $0.000 \pm 0.000$ |

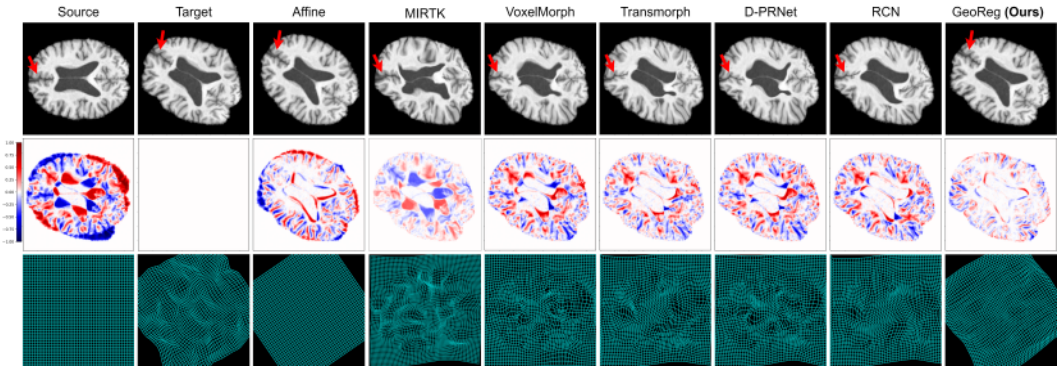


Figure 3: Qualitative results of all compared methods for an intra-subject $(45°, 0°, 0°)$ rotation registration experiment with added random Brownian noise deformation. Red arrows indicate the same brain structure across all registration methods. Our method is able to recover affine and deformable components despite modeling the transformation fully deformably.

enables a more accurate recovery of the smaller, deformable components in a subsequent step, thereby facilitating the overall registration process. To assess the capability of our method to recover deformable transformations, we conduct a comparative evaluation against several baseline methods in three distinct tasks. First, we evaluate our method on inter-subject registration of T1w-T1w images and on multi-contrast inter-subject T1w-T2w MRI brain images from the CamCAN dataset. Next, we

recover intra-subject inhale-exhale motion using the National Lung Screening Trial (NLST) dataset. Results are outlined in Table 2.

Our model demonstrates high accuracy in all three different registration tasks and comes on par with the state-of-the-art methods, using fewer learnable parameters. To test whether the differences are statistically significant, we perform t-tests of our method (GeoReg) over each baseline. In Table 2, the asterisk * denotes non-statistically significant results.

Table 2: Quantitative results measuring the accuracy and regularity of different registration methods on brain T1w inter-subject (CamCAN) and inhale-exhale lung CT (NLST) registration. The performance of GeoReg with bilinear feature warping instead of a learned interpolation component $\delta$ is shown under 'feat. warp'. The asterisk * denotes non statistically significant results of GeoReg over a given baseline ($p < 0.05$).

| | | **Brain CamCAN** | | | **Brain CamCAN T1T2** | | | **Lung NLST** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | # Param. | DSC ↑ | HD95 ↓ | Folding (%) ↓ | DSC ↑ | HD95 ↓ | Folding (%) ↓ | DSC ↑ | HD95 ↓ | TRE↓ | Folding (%) ↓ |
| Affine | - | $0.604 \pm 0.04$ | $4.12 \pm 0.07$ | - | $0.604 \pm 0.04$ | $4.12 \pm 0.07$ | - | $0.928 \pm 0.09$ | $4.24 \pm 0.05$ | $9.07 \pm 0.12$ | - |
| MIRTK [FFD] | - | $0.836 \pm 0.03^*$ | $2.65 \pm 0.48$ | $0.10 \pm 0.08$ | $0.721 \pm 0.06$ | $3.65 \pm 1.22$ | $0.05 \pm 0.03$ | $0.971 \pm 0.02^*$ | $1.31 \pm 0.68$ | $\mathbf{2.76 \pm 0.05}$ | $0.03 \pm 0.01$ |
| ANTs [SyN] | - | $0.808 \pm 0.06$ | $2.55 \pm 0.08^*$ | $0.15 \pm 0.05$ | $0.767 \pm 0.02^*$ | $3.05 \pm 0.90^*$ | $0.10 \pm 0.01$ | $0.973 \pm 0.02^*$ | $1.09 \pm 0.12^*$ | $3.15 \pm 1.05$ | $0.02 \pm 0.01$ |
| LDDMM | - | $0.820 \pm 0.06$ | $2.85 \pm 0.88^*$ | $0.12 \pm 0.01$ | $0.752 \pm 0.06$ | $3.21 \pm 0.95^*$ | $0.08 \pm 0.01$ | $\mathbf{0.984 \pm 0.02}^*$ | $1.25 \pm 0.97^*$ | $3.06 \pm 1.15$ | $0.02 \pm 0.03$ |
| VoxelMorph | 320 k | $0.806 \pm 0.02$ | $3.60 \pm 0.93$ | $0.32 \pm 0.05$ | $0.753 \pm 0.03$ | $3.88 \pm 2.01$ | $0.23 \pm 0.03$ | $0.971 \pm 0.05^*$ | $2.01 \pm 1.30$ | $5.46 \pm 0.66$ | $0.13 \pm 0.10$ |
| LapIRN | 924 k | $0.820 \pm 0.04$ | $3.06 \pm 1.40$ | $0.31 \pm 0.02$ | $0.758 \pm 0.04$ | $3.57 \pm 2.27$ | $0.33 \pm 0.02$ | $0.976 \pm 0.02$ | $1.12 \pm 0.56$ | $3.12 \pm 1.01$ | $0.12 \pm 0.05$ |
| Transmorph | 46.8 M | $0.826 \pm 0.01$ | $2.75 \pm 0.81$ | $0.45 \pm 0.02$ | $0.768 \pm 0.04^*$ | $3.04 \pm 0.23^*$ | $0.32 \pm 0.06$ | $0.975 \pm 0.04^*$ | $1.45 \pm 0.53$ | $4.59 \pm 0.60$ | $0.20 \pm 0.16$ |
| D-PRNet | 1.2 M | $0.828 \pm 0.02$ | $2.89 \pm 0.94$ | $0.47 \pm 0.04$ | $0.733 \pm 0.04$ | $4.63 \pm 1.98$ | $0.28 \pm 0.03$ | $0.970 \pm 0.05$ | $1.92 \pm 1.00$ | $4.76 \pm 0.62$ | $0.19 \pm 0.12$ |
| RCN | 282 M | $0.814 \pm 0.01$ | $2.63 \pm 0.58^*$ | $0.17 \pm 0.02$ | $0.763 \pm 0.28^*$ | $3.21 \pm 1.65^*$ | $0.25 \pm 0.05$ | $0.965 \pm 0.08^*$ | $2.56 \pm 1.54$ | $4.60 \pm 0.64$ | $0.11 \pm 0.09$ |
| FourierNet | 1.1 M | $0.821 \pm 0.06$ | $2.52 \pm 0.57^*$ | $0.18 \pm 0.01$ | $0.767 \pm 0.18^*$ | $3.01 \pm 1.60^*$ | $0.25 \pm 0.05$ | $0.973 \pm 0.02$ | $1.14 \pm 0.12$ | $3.15 \pm 1.24$ | $0.22 \pm 0.01$ |
| **Ours** (feat. warp) | 657 k | $0.812 \pm 0.07$ | $2.51 \pm 0.95^*$ | $0.13 \pm 0.09$ | $0.727 \pm 0.07$ | $\mathbf{2.95 \pm 1.16}^*$ | $0.27 \pm 0.03$ | $0.962 \pm 0.09^*$ | $1.38 \pm 0.41$ | $5.32 \pm 2.20$ | $0.32 \pm 0.02$ |
| **Ours** (GeoReg) | 741 k | $\mathbf{0.838 \pm 0.06}$ | $\mathbf{2.45 \pm 0.82}$ | $0.42 \pm 0.06$ | $\mathbf{0.778 \pm 0.09}$ | $2.98 \pm 0.89$ | $0.16 \pm 0.04$ | $0.972 \pm 0.03$ | $\mathbf{1.08 \pm 0.16}$ | $3.62 \pm 0.63$ | $0.23 \pm 0.09$ |

### 3.2 Discussion and future work

In the absence of ground truth deformation fields, registration algorithms are evaluated using surrogate measures of accuracy that compete against transformation regularity. As a result, one has to resort to inspecting the registration results qualitatively. This can be observed in Table 1 in the rotation experiments, where Transmorph demonstrates high HD95 performance while suffering from a high folding ratio. The inadequacy of the transformation becomes prominent in Figure 3, where Transmorph displays a poor qualitative registration result. On the other hand, we believe our approaches' top-down composition of the transformation by predicting hierarchical refinements provides a strong architectural prior to regularizing a registration task. This also explains our method's high performance despite its comparatively low number of learnable parameters.

Despite optimizations on many aspects of the graph represented in memory, our method has a substantially larger memory footprint than its grid-based counterparts due to having to store explicit intermediate coordinates. Nonetheless, our layers' ability to directly incorporate volume spacing into the deformation prediction could present an interesting avenue to overcome the limitations of current registration approaches in anisotropic tasks.

Finally, this work has only explored large deformations under synthetically transformed intra-subject brain data. However, these synthetic deformations might not capture all the intricate differences of inter-subject registration scenarios. We plan to explore this in future work.

## 4 Conclusion

In this work, we introduce a novel formulation of deformable image registration by using geometric deep-learning principles. We discuss the benefits of estimating deformations on non-fixed grid locations by defining data-driven functions on continuous domains. We outline the need for two types of learned graph operations: A deformation modeling function $\tau$ and a cross-resolution interpolation function $\delta$. Our multi-resolution architecture shows the ability to fully deformably capture a wide range of rotations, translations, and scalings without explicitly modeling an affine component.

We believe that even though our work provides competitive results with state-of-the-art methods, the main contribution of our manuscript lies in establishing a theoretical foundation by which transformations can be explicitly propagated through a deep learning architecture. We think that this contribution opens up avenues of research to reduce the black-box nature of current learned registration paradigms, incorporate ideas from conventional image registration into deep learning architectures, and tackle known data issues such as anisotropicity.

## REFERENCES


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*, 12(1):26–41, 2008.

B. B. Avants, N. Tustison, G. Song, et al. Advanced normalization tools (ants). *Insight j*, 2(365): 1–35, 2009.

G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca. Voxelmorph: a learning framework for deformable medical image registration. *IEEE Transactions on Medical Imaging*, 38 (8):1788–1800, 2019.

G. K. Batchelor. *An introduction to fluid dynamics*. Cambridge university press, 1967.

M. F. Beg, M. I Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61:139–157, 2005.

M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

C. C. Brun, N. Leporé, X. Pennec, Y. Chou, A. D. Lee, G. De Zubicaray, K. L. McMahon, M. J. Wright, J. C. Gee, and P. M. Thompson. A nonconservative lagrangian framework for statistical fluid registration—safira. *IEEE transactions on medical imaging*, 30(2):184–202, 2010.

J. Chen, E. C. Frey, Y. He, W. P. Segars, Y. Li, and Y. Du. Transmorph: Transformer for unsupervised medical image registration. *Medical image analysis*, 82:102615, 2022.

Z. Chen, Y. Zheng, and J. C. Gee. Transmatch: a transformer-based multilevel dual-stream feature matching network for unsupervised deformable image registration. *IEEE Transactions on Medical Imaging*, 2023.

A. V. Dalca, G. Balakrishnan, J. V. Guttag, and M. R. Sabuncu. Unsupervised learning for fast probabilistic diffeomorphic registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018.

J. Farahani, A.and Vitay and F. H. Hamker. Deep neural networks for geometric shape deformation. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)*, pp. 90–95. Springer, 2022.

F. Fuchs, D. Worrall, V. Fischer, and M. Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

L. Hansen and M. P. Heinrich. Deep learning based geometric registration for medical images: How accurate can we get without visual features? In *Information Processing in Medical Imaging: 27th International Conference, IPMI 2021, Virtual Event, June 28–June 30, 2021, Proceedings 27*, pp. 18–30. Springer, 2021a.

L. Hansen and M. P. Heinrich. Graphregnet: Deep graph regularisation networks on sparse keypoints for dense registration of 3d lung cts. *IEEE Transactions on Medical Imaging*, 40(9):2246–2257, 2021b.

G. Haskins, U. Kruger, and P. Yan. Deep learning in medical image registration: a survey. *Machine Vision and Applications*, 31:1–18, 2020.

A. Hering, L. Hansen, T. CW Mok, A. CS Chung, H. Siebert, S. Häger, A. Lange, S. Kuckertz, S. Heldmann, W. Shao, et al. Learn2reg: comprehensive multi-task medical image registration challenge, dataset and evaluation in the era of deep learning. *IEEE Transactions on Medical Imaging*, 2022.

A. Hoopes, J. E. Iglesias, B. Fischl, D. Greve, and A. V. Dalca. Topofit: rapid reconstruction of topologically-correct cortical surfaces. *Proceedings of machine learning research*, 172:508, 2022.

A. Horn. MNI T1 6thGen NLIN to MNI 2009b NLIN ANTs transform. 7 2016.

B. Hu, S. Zhou, Z. Xiong, and F. Wu. Recursive decomposition network for deformable image registration. *IEEE Journal of Biomedical and Health Informatics*, 26(10):5130–5141, 2022.

J. Iglesias, C. Liu, P. Thompson, and Z. Tu. Robust brain extraction across datasets and comparison with publicly available methods. *IEEE Transactions on Medical Imaging*, 30(9):1617–1634, 11 2011.

X. Jia, J. Bartlett, W. Chen, S. Song, T. Zhang, X. Cheng, W. Lu, Z. Qiu, and J. Duan. Fourier-net: Fast image registration with band-limited deformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 1015–1023, 2023.

M. Kang, X. Hu, W. Huang, M. R. Scott, and M. Reyes. Dual-stream pyramid registration network. *Medical image analysis*, 78:102379, 2022.

A. Kashefi, D. Rempe, and L.s J. Guibas. A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries. *Physics of Fluids*, 33(2), 2021.

C. Ledig, R. Heckemann, A. Hammers, J. López, V. Newcombe, A. Makropoulos, J. Lötjönen, D. Menon, and D. Rueckert. Robust whole-brain segmentation: Application to traumatic brain injury. *Medical image analysis*, 21 1:40–58, 2015.

H. Lester, S. R. Arridge, K. M. Jansons, L. Lemieux, J. V. Hajnal, and A. Oatridge. Non-linear registration with the variable viscosity fluid algorithm. In *Information Processing in Medical Imaging: 16th International Conference, IPMI'99 Visegrád, Hungary, June 28–July 2, 1999 Proceedings 16*, pp. 238–251. Springer, 1999.

Y. Liu, L. Zuo, S. Han, Y. Xue, J. L. Prince, and A. Carass. Coordinate translator for learning deformable medical image registration. In *International Workshop on Multiscale Multimodal Medical Imaging*, pp. 98–109. Springer, 2022.

B. Lowekamp, D. Chen, L. Ibáñez, and D. Blezek. The design of simpleitk. *Frontiers in Neuroinformatics*, 7, 2013.

M. Meng, L. Bi, D. Feng, and J. Kim. Non-iterative coarse-to-fine registration based on single-pass deep cumulative learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 88–97. Springer, 2022.

T. C. W. Mok and A. C. S. Chung. Large deformation diffeomorphic image registration with laplacian pyramid networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23*, pp. 211–221. Springer, 2020a.

T. C.W. Mok and A. C.S. Chung. Large deformation diffeomorphic image registration with laplacian pyramid networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23*, pp. 211–221. Springer, 2020b.

H. Qiu, C. Qin, A. Schuh, K. Hammernik, and D. Rueckert. Learning diffeomorphic and modality-invariant registration using b-splines. In *International Conference on Medical Imaging with Deep Learning*, 2021.

D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *IEEE Transactions on Medical Imaging*, 18:712–721, 1999.

R. Sandkühler, S. Andermatt, G. Bauman, S. Nyilas, C. Jud, and P. C. Cattin. Recurrent registration neural networks for deformable image registration. *Advances in Neural Information Processing Systems*, 32, 2019.

A. Schuh, M. Murgasova, A. Makropoulos, C. Ledig, S. Counsell, J. Hajnal, P. Aljabar, and D. Rueckert. Construction of a 4d brain atlas and growth model using diffeomorphic registration. In *STIA*, 2014.

M. Shafto, L. Tyler, M. Dixon, Jason R. Taylor, J. Rowe, R. Cusack, A. Calder, W. D. Marslen-Wilson, J. Duncan, T. Dalgleish, R. Henson, C. Brayne, and F. Matthews. The cambridge centre for ageing and neuroscience (cam-can) study protocol: a cross-sectional, lifespan, multidisciplinary examination of healthy cognitive ageing. *BMC Neurology*, 14, 2014.

Z. Shen, J. Feydy, P. Liu, A. H. Curiale, R. San Jose Estepar, R. San Jose Estepar, and M. Niethammer. Accurate point cloud registration with robust optimal transport. *Advances in Neural Information Processing Systems*, 34:5373–5389, 2021.

A. Sotiras, C. Davatzikos, and N. Paragios. Deformable medical image registration: A survey. *IEEE Transactions on Medical Imaging*, 32:1153–1190, 2013.

M. A. Suliman, L. Z. J. Williams, A. Fawaz, and E. C. Robinson. Geomorph: Geometric deep learning for cortical surface registration. In *Geometric Deep Learning in Medical Image Analysis*, 2022.

M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020.

J. Taylor, N. Williams, R. Cusack, T. Auer, M. Shafto, M. Dixon, L. Tyler, Cam-CAN Group, and R. Henson. The cambridge centre for ageing and neuroscience (cam-can) data repository: Structural and functional mri, meg, and cognitive data from a cross-sectional adult lifespan sample. *Neuroimage*, 144:262 – 269, 2017.

NLST team. The national lung screening trial: overview and study design. *Radiology*, 258(1): 243–253, 2011.

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

T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage*, 45(1):S61–S72, 2009.

M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In *International work-conference on artificial neural networks*, pp. 758–770. Springer, 2005.

H. Wang, D. Ni, and Y. Wang. Modet: Learning deformable image registration via motion decomposition transformer. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 740–749. Springer, 2023.

H. Xiao, X. Teng, C. Liu, T. Li, G. Ren, R. Yang, D. Shen, and J. Cai. A review of deep learning-based three-dimensional medical image registration methods. *Quantitative Imaging in Medicine and Surgery*, 11(12):4895, 2021.

S. Zhao, Y. Dong, E. I. Chang, Y. Xu, et al. Recursive cascaded networks for unsupervised medical image registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10600–10610, 2019.

Y. Zhu and S. Lu. Swin-voxelmorph: A symmetric unsupervised learning model for deformable medical image registration using swin transformer. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 78–87. Springer, 2022.