ResNet-LDDMM: Advancing the LDDMM Framework Using Deep Residual Networks

Sylvain Arguillere, Boulbaba Ben Amor, Ling Shao

To cite this version:

Sylvain Arguillere, Boulbaba Ben Amor, Ling Shao. ResNet-LDDMM: Advancing the LDDMM Framework Using Deep Residual Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. hal-03413643

HAL Id: hal-03413643
https://hal.science/hal-03413643v1

Submitted on 3 Nov 2021

ResNet-LDDMM: Advancing the LDDMM Framework Using Deep Residual Networks

Boulbaba Ben Amor, *Senior, IEEE*, Sylvain Arguillère and Ling Shao, *Fellow, IEEE*

Abstract—In deformable registration, the geometric framework – large deformation diffeomorphic metric mapping or LDDMM, in short – has inspired numerous techniques for comparing, deforming, averaging and analyzing shapes or images. Grounded in flows, which are akin to the equations of motion used in fluid dynamics, LDDMM algorithms solve the flow equation in the space of plausible deformations, i.e. diffeomorphisms. In this work, we make use of deep residual neural networks to solve the non-stationary ODE (flow equation) based on a Euler's discretization scheme. The central idea is to represent time-dependent velocity fields as fully connected ReLU neural networks (building blocks) and derive optimal weights by minimizing a regularized loss function. Computing minimizing paths between deformations, thus between shapes, turns to find optimal network parameters by back-propagating over the intermediate building blocks. Geometrically, at each time step, ResNet-LDDMM searches for an optimal partition of the space into multiple polytopes, and then computes optimal velocity vectors as affine transformations on each of these polytopes. As a result, different parts of the shape, even if they are close (such as two fingers of a hand), can be made to belong to different polytopes, and therefore be moved in different directions without costing too much energy. Importantly, we show how diffeomorphic transformations, or more precisely bilipshitz transformations, are predicted by our algorithm. We illustrate these ideas on diverse registration problems of 3D shapes under complex topology-preserving transformations. We thus provide essential foundations for more advanced shape variability analysis under a novel joint geometric-neural networks Riemannian-like framework, i.e. ResNet-LDDMM.

Index Terms—Diffeomorphic Registration, LDDMM, Deep Residual Neural Networks, Computational Anatomy, Riemannian Geometry.

◆

1 INTRODUCTION AND RELATED WORK

THE deformable registration problem involves finding a single coordinate system in which to pinpoint several different shapes. This allows, for example, for the statistical analysis of shape data that takes into account their geometric properties. Many applications of this principle can be found in computational anatomy, in which the shapes are extracted from medical images (MRI, PET scans,...) [1]. Various methods have been used to register different shapes or anatomical organs (see for example [2] and [3]). The LDDMM (large deformation diffeomorphic metric mapping) approach is one of the most popular for estimating plausible transformations (diffeomorphisms). Taking advantage of the group structure of the manifold of diffeomorphisms of $\mathbb{R}^3$, it also comes with a proper metric for comparing shapes based on a certain kinetic energy of the deformation [4].

1.1 Problem Formulation

In the present work, we will restrict our study to discrete, unparametrized surfaces and more generally point sets/clouds of $\mathbb{R}^3$, denoted by $q = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^{n\times 3}$, $i \neq j \Rightarrow x_i \neq x_j$ (n is the number of vertices/points in the point cloud or the meshed surface). In the language of shape analysis, this means that we work on the so-called spaces of landmarks. Given $q_S = (x_1, \ldots, x_n)^T$ (source/template shape) and q_T (target/reference shape), two point sets representing the same physical object or anatomical organ (liver, kidney, femur, heart, hippocampus, brain cortex, or simply a hand) of the human body, our goal is to find a *reasonable* transformation $\phi : \mathbb{R}^3 \to \mathbb{R}^3$ such that a transformed version of the template shape $\phi.q_S = (\phi(x_1), \ldots, \phi(x_n))^T$ is similar to the reference q_T. If the transformation ϕ is *diffeomorphic* (i.e. smooth with smooth inverse), then $(\phi, q) \mapsto \phi.q$ is the associated *group action* on the space of landmarks [5]. In our special case of point sets, this smooth action is given by $\phi.q = (\phi(x_1), \ldots, \phi(x_n))$. That is, the i-th landmark of $\phi.q$ is the position of the i-th landmark of q after being moved in $\mathbb{R}^3$ by ϕ.

A common way to model the *deformable registration* problem is to consider the minimization of an energy functional (Eq. (1)) over the set of plausible deformations ϕ,

$$\mathcal{J}(\phi; q_S, q_T) = \underbrace{\mathcal{D}(\phi.q_S, q_T)}_{\text{Data term}} + \underbrace{\mathcal{R}(\phi)}_{\text{Regularizer}} \tag{1}$$

where $\mathcal{D}(.,.)$ is the *data term* that measures the discrepancy between the deformed shape $\phi.q_S$ and the target shape q_T. The term $\mathcal{R}(.)$ plays the role of a *regularizer* and thus controls the plausibility of the solution ϕ^*. In several applications, particularly when analyzing the anatomical parts/organs of the body, it is desirable to have a deformation ϕ that preserves local and global topology, preventing the deformation from creating holes or folding when applied to the source shape. This is true for diffeomorphisms, for example. It is also true for the slightly more general *bilipshitz maps*, which are essentially diffeomorphisms as well; they are homeomorphisms Φ such that both Φ and Φ^{-1} have a bounded rate of change (specifically, they are differentiable almost everywhere, with invertible differential). To

---

- *B. Ben Amor and L. Shao are with the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates. S. Arguillère is with Laboratoire CNRS Paul Painlevé, Université de Lille, France.*
  E-mail: boulbaba.amor@inceptioniai.org

Manuscript received February, 2021; revised XXX, 2021.

better understand the advantage of transformations that are (essentially) diffeomorphisms over simple displacements, we will compare in the following the elastic deformation models [6] and the LDDMM framework introduced for the first time in [2] to deal with large deformations.

– *Elastic deformation models*, pioneered by *C. Broit* [6], compute a deformation $x \mapsto \phi(x)$, $x \in \mathbb{R}^3$ by perturbations $v(x)$ from the identity, such that,

$$\phi(x) = Id(x) + v(x) = x + v(x).$$

However, for large deformations, these models cannot guarantee diffeomorphic transformations (i.e., topology-preserving and invertible mappings).

– *LDDMM and variants*, introduced by *Beg et al.* in [2], compute diffeomorphic transformations through the integration of smooth, time-dependent *velocity fields* $f : [0,1] \times \mathbb{R}^3 \to \mathbb{R}^3$ over time. Accordingly, time-dependent transformations $\phi : [0,1] \times \mathbb{R}^3 \to \mathbb{R}^3$ are derived. They are governed by an Ordinary Differential Equation (ODE), known as the *flow equation* [4] formulated as follows (Eq. (2)):

$$\begin{gathered}\dot{\phi}(t,x) = f(t,\phi(t,x)), \quad \phi(0,x) = x \\ \text{for all } x \in \mathbb{R}^3 \text{ and } t \in [0,1],\end{gathered} \tag{2}$$

where $\dot{\phi} = \frac{\partial \phi}{\partial t}$ denotes the partial derivative over the variable time t and $\phi(0,x)$ is the initial state taken to be the identity (diffemorphism) of x. Under adequate assumptions on f (globally Lipschitz in space for fixed t, with a time-dependent Lipshitz constant integrable in time, for example), $f \mapsto \phi^f$ is a well-defined mapping into the space of time-dependent (essentially) diffeomorphisms of $\mathbb{R}^3$ by the Cauchy-Lipshitz theorem. A smoother f also yields a smoother ϕ^f (and, hence, a diffeomorphism), as seen in [7] and [4] (see also [5], Theorem 8.7). The goal is then to find an optimal trajectory $\phi(t,.)$ connecting $\phi(0,.) = I_{\mathbb{R}^3}$ (starting point) to the end point $\phi(1,.)$ by finding a minimizer $f^* : [0,1] \times \mathbb{R}^3 \to \mathbb{R}^3$ of the following updated version of Eq. (1):

$$f^* = \underset{f(t,.) \in \mathcal{A}}{\operatorname{argmin}} \frac{1}{2\sigma^2} \mathcal{D}(\phi^f(1).q_S, q_T) + \frac{1}{2} \int_0^1 \|f(t,.)\|^2_{\mathcal{A}} dt. \tag{3}$$

Here, $\mathcal{A}$ is the space of *admissible* velocity fields (i.e. with the smoothness condition) which give rise to $\mathrm{Diff}_{\mathcal{A}}(\mathbb{R}^3)$, defined by $\mathrm{Diff}_{\mathcal{A}}(\mathbb{R}^3) = \{\phi^f(1), f \in L^1([0,1],\mathcal{A})\}$, the Group of Diffeomorphisms associated to $\mathcal{A}$.

From the formulation above, we obtain the following interesting geometric interpretations and properties,

– The quantity $\mathcal{S}_t = \|f(t,.)\|^2_{\mathcal{A}}$ is the *kinetic energy* of the whole system at the time step t (refer to [5]).

– For $q, \tilde{q}$ two sets of landmarks with the n points, $n \in \mathbb{N}$,

$$d^n_{\mathcal{A}}(q,\tilde{q}) = \inf_{f(t,.) \in \mathcal{A}} \{ \int_0^1 \|f(t,.)\|_{\mathcal{A}} dt, \tilde{q} = \phi^f.q \}$$

is a *metric* on the space of landmarks with n points. One can even deduce a metric $d_{\mathcal{A}}$ on $\mathrm{Diff}_{\mathcal{A}}$ itself in a similar way, making $(\mathrm{Diff}_{\mathcal{A}}, d_{\mathcal{A}})$ a *complete metric space* ([7] and [5], Theorem 8.15). This is a very important characteristic of the LDDMM method and can be used to assess the similarity of various objects by analysis of the velocity fields f that induce the transformation $\phi(1)$, which aligns these objects.

– The measure of the kinetic energy of paths in the space of considered shapes is used as a regularizer term in the registration problem formulated in Eq. (3). Thus, the functional Eq. (3) to be minimized is the sum of a first term defined as a geometric norm of the control (kinetic energy of the deformation) and of a data term providing a geometric distance to the target shape. The weighting factor $1/2\sigma^2$ balances the influence of the data attachment $\mathcal{D}$ and the regularizer term $\mathcal{R}$ of the general formulation in Eq. (1).

This elegant LDDMM framework has served (and is serving) as a starting point for several approaches in the literature which can be categorized into: *(i) relaxation methods* (e.g. [2]), which compute velocities for multiple points in time, and *(ii) shooting methods* (e.g. [3]) which take advantage of the conservation of momentum (and in particular a constant kinetic energy $\mathcal{S}^*_t = \|f^*(t,.)\|^2_{\mathcal{A}}$) and determine the evolution of the transformed template based solely on the initial velocity. Importantly, both approaches define an admissible Hilbert space $\mathcal{A}$ of velocity fields as an RKHS (Reproducing Kernel Hilbert Space). The kernel is taken to be the Green's kernel $K = L^{-1}$, (where L is a differential operator which defines the inner product of $\mathcal{A}$ as well as the induced norm $\|.\|_{\mathcal{A}}$) [5]. In practice, smoothing of the velocity fields is achieved by convolution with suitable kernels, e.g., Gaussian kernels with positive scale. With this definition of $\mathcal{A}$ as an RKHS, we complete the formulation of the conventional LDDMM framework.

1.2 Related Work

We restrict our review to two categories: *(1) diffeomorphic registration methods* primarily derived from LDDMM and variants. We focus, in particular, on recent end-to-end unsupervised deep neural network approaches for predicting diffeomorphic transformations; *(2) non-rigid point cloud registration methods* which densely match a source shape with a target shape without imposing any constraint on the output transformation. These approaches require, in general, initial correspondence. First, however, we discuss in more details the rich literature of LDDMM and its variants.

1.2.1 Large Deformation Diffeomorphic Metric Mapping

Under the LDDMM formulation, we can distinguish two paradigms [8]. Firstly, the *optimize-then-discretize* schema which first derives the *Hamiltonian* of the continuous problem, then, deduces optimality conditions of continuous optimization problems using the calculus of variations (e.g. [9], [10] [11], [12], [3], [13], [14], and [15]). Ultimately, these conditions are discretized and solved, typically using Gradient Descent. In computational anatomy, LDDMM algorithms (both *relaxation* and *shooting*) achieve accurate registration results, but are costly in running time (counting hours on CPU) and memory. As a response to this, the *discretize-then-optimize* paradigm was born. It involves first discretizing the objective functional and constraining equations, and then solving the discrete optimization problem using numerical optimization methods (e.g. [16], [17] and [8]). Within the *optimize-then-discretize* paradigm, solving the *Hamiltonian* of the continuous LDDMM problem guarantees diffeomorphic transformations, but after discretization this property could be violated. In contrast, under the *discretize-then-optimize* paradigm, piecewise diffeomorphic transformations, invertible and topology-preserving transformations are achieved as shown in [8]. Both approaches require a predefined Gaussian kernel with a fixed scale of deformation.

Similar to LDDMM, the SVF (Stationary Velocity Fields) framework is an alternative for finding a diffeomorphic transformation between shapes. It was first introduced by Arsigny et al. in [18]. As LDDMM, SVF works on a vector space of images and a Lie group of diffeomorphic transformations. SVF generalizes the principal logarithm to non-linear geometrical deformations which falls into parameterizing diffeomorphisms with stationary speed vector fields. Despite its satisfactory results in practice, the logarithm is well-defined only for transformations close enough to the identity (as the Lie group exponential map is usually not surjective), which makes the registration results under large deformations uncertain. That is, the optimal transformation is not smooth with regard to the images, so that a small change in images may lead to a large change in the path connecting them. Furthermore, the underlying space (of the SVF framework) is not a Riemannian manifold and there is no Riemannian metric, geodesic, or connection involved. While an LDDMM curve is obtained by integrating a time-dependent vector field specified by the Riemannian metric, an SVF curve is an integral curve of a stationary vector field, in the corresponding Lie algebra. Hence, SVF works on the structure of the Lie group of diffeomorphic transformations instead of the underlying Riemannian manifold [19]. In the literature, the SVF framework is always considered an approximation of the *shooting* formulation of LDDMM.

1.2.2 Deep Neural Network Methods

Most recent end-to-end unsupervised deep learning diffeomorphic registration approaches are inspired from the SVF framework. Indeed, they make use of spacial transformer networks (or STNs), first introduced in [20], to warp the source (moving image) to the target (fixed image) by optimizing a loss function including both an image similarity term and a regularization on the transformations term. One pioneering approach is called *VoxelMorph* and works on (structured) 3D medical images [21], [22]. The Network, a UNet, consists of two steps. First, it approximates posterior probability parameters representing the velocity field mean and variance using a convolutional encoding-decoding scheme. Second, obtained velocity fields are transformed into diffeomorphic transformation using differentiable squaring and scaling integration layers. Similarly, Krebs et al. proposed in [23] to use a conditional variational autoEncoder (*CVAE*) to infer a low-dimensional probabilistic deformation model under constraints making the transformations symmetric and diffeomorphic. The main idea is to build the target by warping the source image, where the latent space encodes deformations. Again, successive differentiable squaring and scaling layers, as previously used in [21], generate diffeomorphic transformations. The decoder extracts velocities and diffeomorphisms at different scales and a Gaussian smoothing layer is applied on the filter maps (latent variables) to smooth the initial velocity. Taking advantage of symmetric approaches, Mok and Chung proposed in [24] a symmetric diffeomorphic neural network (*SDNN*) which is an unsupervised symmetric image registration method which maximizes the similarity between images within the space of diffeomorphic maps and estimates both forward and inverse transformations simultaneously. On top of this, they proposed a selective Jacobian determinant regularization that imposes a local orientation, consistency constraint on the estimated deformation fields. Also, inspired by the SVF framework, Bone et al. have proposed in [25] a diffeomorphic autoencoder (*DAE*). In [26], Shen et al. have proposed a deep-learning framework which combines affine registration and a vector momentum-parameterized stationary velocity field (*vSVF*). Under the vSVF approach, the goal is also to solve a stationary ODE that represents a one-parameter subgroup of diffeomorphisms. The integration is achieved via multiple scaling and squaring layers. The way to accommodate a convolutional neural network to compute vSVF, under a supervised learning scheme, was first proposed in [27]. Taking a slightly different direction and inspired by the work of Vialard et al. [3], Niethammer et al. have proposed in [28] to learn a spatially-varying regularizer. They built their unsupervised registration model on top of *vSVF* [18]. Their approach jointly optimizes the regularizer (parameterized by a deep CNN) and the registration parameters of the *vSVF* model. To allow much freedom, Shen et al. have proposed in [29] a Region-specific Diffeomorphic Metric Mapping (*RDMM*) to allow for spatially-varying regularization. Unlike spatially-invariant regularizers, spatially-varying regularization allows anticipating different levels of deformations at different image locations. It was observed experimentally that *RDMM* may locally exhibit stronger deformations than *LDDMM* or *vSVF*.

1.2.3 Non-rigid Point Cloud Registration Methods

Less constrained, but more efficient approaches have also been developed with the increasing amount of 3D data available and depth-sensors. These approaches solve the problem of matching two 3D point clouds in the presence of non-rigid deformations. They typically adopt the ℓ_p type robust estimator to regularize the fitting and smoothness. For instance, Amberg et al. (who developed an ℓ_2-regularization method) have extended in [30] the popular ICP (iterative closest point) algorithm to cover non-rigid transformations. They include in their framework different regularizations, as long as they have an adjustable stiffness parameter. So, thus, the registration loops over a series of decreasing stiffness weights, and incrementally deforms the template towards the target, recovering the whole range of global and local deformations. However, the presence of noise, outliers, holes, or articulated motions between the point clouds can potentially result in alignment errors (we will therefore refer to this approach as *N-ICP*). Taking another direction, Li et al. proposed in [31] the *RPTS* method with ℓ_1-regularization using re-weighted sparsities on positions and transformations to estimate the deformations on point clouds. They formulated the energy function with dual sparsities on both the data term and the smoothness term, and defined the smoothness constraint using local rigidity. Starting from the observation of the existence of an intrinsic articulated subspace in most non-rigid motions, Guo et al. [32] proposed an ℓ_0-based motion regularizer with an iterative optimization solver that can implicitly constrain local deformation only on joints with articulated motions. As a consequence, the solution space is reduced to physical plausible deformations. We will refer to this approach as *SVR-ℓ_0*. Recently, in [33], Yao et al. proposed a formulation based on a globally smooth robust estimator for data fitting and regularization, which can handle outliers and partial overlaps by enforcing sparsity using the *Welsch's function*. They made use of the majorization-minimization (MM) algorithm to tackle the problem, which reduces each iteration to solve a simple least-squares problem with L-BFGS. Their approach (which we will call *QNS*) achieves lower registration errors compared to the previous approaches while improving the execution time. While these approaches bring efficient solutions to solve the non-rigid registration problem, they often require good initialization, i.e., a set of labeled landmarks on both meshes. Further, they do not guarantee diffeomorphic transformations between 3D shapes and registering a shape A to a shape B, and inversely, can yield in different matching results.

1.3 Contributions and Paper Organization

In this work, we propose a novel *joint geometric-neural networks framework* for diffeomorphic registration of 3D shapes, and 3D point clouds, in general. Our formulation differs in several aspects from previous end-to-end deep learning approaches. Importantly, we introduce a Riemannian-like framework (that is, a length space), for comparing and deforming 3D shapes, by revisiting the original LDDMM framework using deep residual networks.

– Inspired by the SVF framework, most of end-to-end deep neural network diffeomorphic registration approaches tend to solve a stationary ODE; thus, they are *static*. They work on the group structure of the underlying Lie group of diffeomorphic transformations and compute stationary vector fields on a fixed tangent space. Consequently, they do not allow a Riemannian metric to be defined or geodesics (minimizing paths) to be computed in the shape space. In contrast, our ResNet-LDDMM is *dynamic* as it was designed to solve the non-stationary flow equation ODE Eq.(2) by integrating time-dependent velocity fields. ResNet-LDDMM offers more flexibility and is more suitable for handling large and, to some extent, complex deformations.

– Taking advantage of the functional space of deep neural networks, our ResNet-LDDMM predicts time-dependent, regular and smooth velocity fields (the endpoint of the flow is the desired diffeomorphic transformation). The network structure brings a natural regularization at a primary level, in addition to the regularization related to the time-integrated kinetic energy (or the geometric norm of the control). We demonstrate in Sec. 3 how ResNet-LDDMM predicts smooth and regular time-varying velocity fields. So, just like in LDDMM, ResNet-LDDMM does generate (essentially) a diffeomorphism, without defining a Gaussian kernel (with a fixed scale of deformation).

– Appropriately applying the ReLU activation function to predict velocity fields, we reveal how each time-step of our ResNet-LDDMM divides the space ($\mathbb{R}^3$) into multiple polytopes (i.e., unbounded polyhedra) in which optimal velocity fields are computed. This provides more flexibility in the registration while maintaining diffeomorphic properties. We describe these nice geometric interpretations (Sec. 3.2) on several examples involving hand shapes and complex anatomical organs.

– Our ResNet-LDDMM makes use of fully connected ReLU networks (operated by 1x1 convolutions on 3D points taken individually). This enables to operate on unstructured 3D data without explicitly considering neighborhood information. This is a fundamental difference from the end-to-end deep neural networks approaches described in Sec. 1.2.2, proposed mainly for medical image registration.

– On different internal anatomical organs, we study the performance of our ResNet-LDDMM in comparison with LDDMM [13]. We show its superiority in some cases. We compare to other non-rigid registration approaches as N-ICP [30], RPTS [31], SVR-ℓ_0 [32] and QNS [33] on body shapes. We highlight in particular its behavior in the presence of noise and missing data.

To sum up, we provide the initial and essential foundations for the analysis of shapes and their variability analysis under a Riemannian-like framework, called ResNet-LDDMM. The rest of the paper is organized as follows. Sec. 1 formulates the diffeomorphic registration problem. A comprehensive survey is then drawn on non-rigid (diffeomorphic) registration, before summarizing the contributions the present work. In Sec. 2 we describe our ResNet-LDDMM framework from both geometric and (unsupervised learning) neural networks perspectives. For pedagogical purposes, we also provide an algorithmic summary of ResNet-LDDMM (Algo. 1) and how compute minimizing geodesics (Algo. 2). They represent the main ingredients for conducting statistical analysis. How ResNet-LDDMM naturally builds diffeomorphic transformations between shapes is formally elaborated in Sec. 3. Furthermore, we provide a geometric description of the velocity fields accompanied by a qualitative interpretation on the registration methodology behind ResNet-LDDMM. Sec. 4 is dedicated to multiple experiments. It includes, in particular, an ablation study regarding the key features of ResNet-LDDMM. Some concluding remarks and perspectives are drawn in Sec. 5.

2 RESNET-LDDMM, OR LDDMM REVISITED

The central idea in our ResNet-LDDMM is to make use of a particular family of deep neural networks, i.e. residual deep networks (or ResNets), to solve the flow equation Eq. (3). In supervised learning, a residual network [35], [36]) has the following form:

$$h^{l+1} = h^l + f^l(h^l, \theta^l) \tag{4}$$

where h^l is the trainable hidden layer, $l \in \{1, \dots, L\}$, and $\theta^l \in \{\theta^1, \dots, \theta^L\}$ denotes the network parameters. Compared to a standard feed-forward network, where $h^{l+1} = f(h^l, \theta^l)$, at the heart of ResNet is the ultimate idea that every additional layer should contain the identity function as one of its elements. Thus, a residual $h^{l+1} - h^l$ is learned instead of transforming the output of the previous layer h^l to h^{l+1}. Recent works have pointed out a striking similarity between this important property in residual networks and the numerical solution of ordinary differential equations

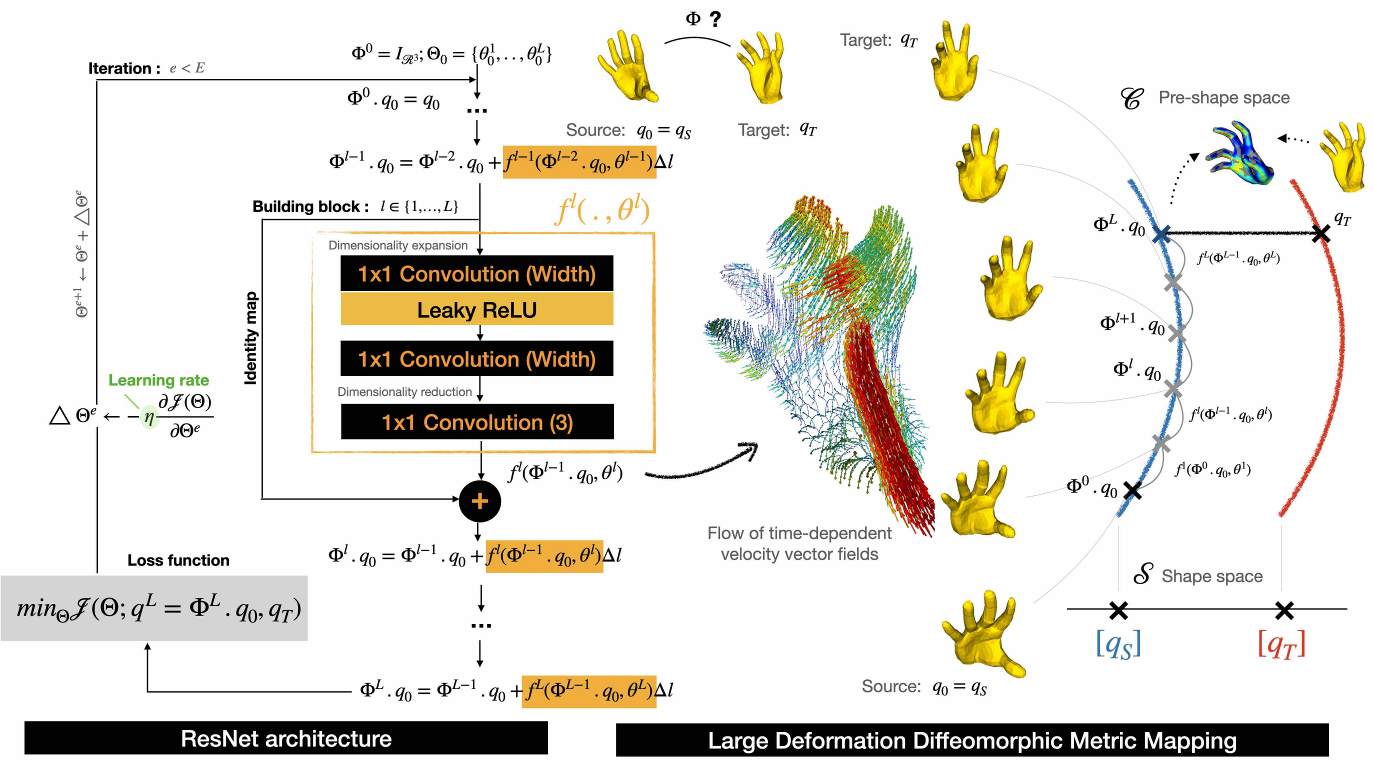

Fig. 1. Overview of our joint ResNet-LDDMM diffeomorphic registration framework – the ResNet architecture is shown on the left, comprising of an ensemble of successive two-layers ReLU networks followed by a dimensionality reduction layer (the ensemble is called a building block). Each predicts a time-dependent velocity field $f^l(.,\theta^l)$. On the right, we show the flow of obtaining velocity fields and diffeomorphisms Φ^l, including the end point Φ^L, achieved by integration. In theory (e.g. [34]), both orbits $[q_S]$ and $[q_T]$ coincide, so q_S can be deformed to a point in the neighborhood of q_T, for a given data fit error. Thus, the shape space is nothing else than the orbit of q_s.

(e.g. [37], [38], [39]) using the forward Euler method with a given initial value. In fact, training a deep residual network is viewed as a discretization of a dynamical training system governed by a first-order ODE, where the network layers are viewed as time-steps and the network parameters θ^l are interpreted as the *control* to optimize [40].

Inspired by the general formulation and aiming to solve the flow equation Eq. (2), we build our ResNet-LDDMM framework. As shown in Fig. 1, our ResNet-LDDMM is an ensemble of L successive identical building blocks. Each building block is composed of three fully connected layers and a point-wise ReLU activation function separating the first two layers. From the LDDMM perspective (right panel of Fig. 1), each building block f^l represents a velocity vector field $\mathbb{R}^3 \to \mathbb{R}^3$ at a discrete time l. The f^l are parameterized by θ^l. If f^l are sufficiently smooth and spatially regular (we postpone this demonstration to Sec. 3), one can then build a diffeomorphic transformation $\Phi^f(1)$ (by integration of Eq. (5)) that moves the initial shape q_S to fit q_T, which is mathematically equivalent to $\Phi(1).q_S \sim q_T$ and $\Phi(1) = \Phi^L$:

$$\Phi^{l+1}.q_0 = \Phi^l.q_0 + \Delta^L f^l(\Phi^l.q_0, \theta^l), \text{ with } \Delta^L = \frac{1}{L}. \tag{5}$$

Compared to the conventional LDDMM formulation, here, the time-varying velocity vector fields are deep neural networks $f^l(.,\theta^l)$ that should be optimized with respect to a fit error. To this end and if assume a ResNet-LDDMM with number of building blocks $L \to \infty$ (i.e. $\frac{1}{L} \to 0$), we cast Eq. (3) as:

$$\begin{aligned}\Theta^*, \Phi^*(1) = \underset{\Theta=\{\theta^t\}_{t\in[0,1]}}{\operatorname{argmin}} \frac{1}{2\sigma^2}\mathcal{D}(\Phi^\Theta(1).q_S, q_T) \\ + \frac{1}{2}\int_0^1 \|f^t(\Phi^\Theta(t).q_S, \theta^t)\|_2^2 dt,\end{aligned} \tag{6}$$

subject to $\partial_t \Phi^\Theta(t,.) = f^t(\Phi^\Theta(t,.), \theta^t)$ for a.e. $t \in [0,1]$ and $\Phi(0,.)$ denotes the identity map (the neutral element of the group $\text{Diff}(\mathbb{R}^3)$). The family $f^t(.,\theta^t)$ is the ensemble of functions to be approximated by the network which coincides with the time-dependent velocity fields along the time interval $[0,1]$ and $\Theta = (\theta^t)_{t\in[0,1]}$ are the whole ResNet-LDDMM parameters. $\Phi^\Theta(1)$ is the final transformation and $\Phi^\Theta(t)$ are the intermediate transformations. The choice of $\mathcal{D}(.,.)$ will be detailed in the next section (Sec. 2.1). The second term is a regularizer that presents the length of the path connecting q_S to $\Phi^\Theta(t).q_S$, on the orbit of q_S, which ensures that the optimal vector field stays regular enough to ensure topology preserving transformations.

In the right panel of Fig. 1 we illustrate how ResNet-LDDMM enables a minimizing path (a geodesic) on the orbit of q_s. Intermediate shapes are discrete elements of the geodesic outputs of the action of the intermediate diffeomorphisms Φ^l on the source shape q_S. Φ^l are deduced from the optimal vector fields $f^l(.,\theta^l)$ sitting in between the panels of Fig. 1 (colors (cold → hot) reflect the absolute amount of displacement at each time-step and at a vertex-level).

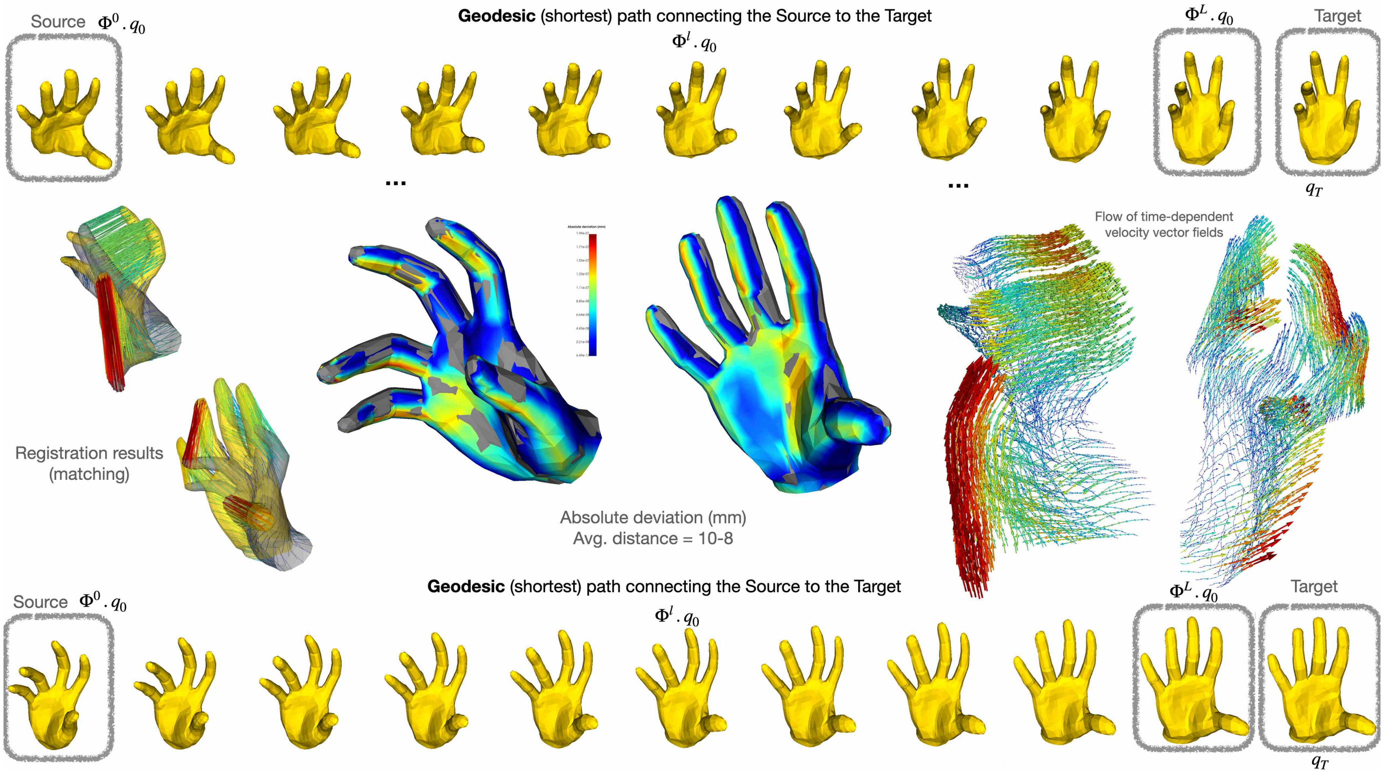

Fig. 2. A pictorial summary of our diffeomorphic registration results. Top: a geodesic path connecting a source hand shape to a target shape; Bottom: the inverse path (source and target change roles); Center-right: both flows of the time-dependent velocity fields $f^l(.,\theta^l)$, building blocks of our ResNet-LDDMM. Center-left: final Euclidean displacements and registration results.

2.1 ResNet-LDDMM Network Architecture

Let us now come back to the architecture of our ResNet-LDDMM. As illustrated in the left panel of Fig. 1, our network takes as inputs (1) the source shape q_S, or to be more accurate $\Phi^0.q_S$, where Φ^0 is the identity diffeomorphism of $\mathbb{R}^3$, and (2) an initial guess $\Theta_0 = (\theta_0^1, \dots, \theta_0^l)$, initial parameters of the network and (3) the target shape q_T to evaluate the loss function Eq. (6). The *velocity fields* $f^l(\cdot,\theta^l)$ and the corresponding diffeomorphisms Φ^l are predicted by the l-indexed successive building blocks of the network (orange box in Fig. 1). The number of building blocks L in the ResNet-LDDMM coincides with the total number of time-steps in the conventional LDDMM, once discretized.

– Each **building block** (denoted by $f^l(.,\theta^l)$ and generating the l-th velocity field on $\mathbb{R}^3$) consists of three successive fully connected weight layers. Weight matrices W_i and bias vectors b_i (limited to the first and second weight layers) define point-wise convolution operations, with kernel filters of size $1 \times 1(\times m)$, performed separately on all points. This is equivalent to designing a fully connected layer as described in the Network-in-Network [41]. We will denote m as the width of individual building blocks. Subsequently, a ReLU activation function is applied to the output of the first weight layer within each building block. The last weight layer reduces the dimensionality of the output of the previous layer to get velocity vectors in $\mathbb{R}^3$ (so, the width of this layer is exactly 3). While the weight layers conduct affine transformations of the input (output of the previous building block), ReLU reduces to zeros all negative outputs of the first layer, thus dividing the space $\mathbb{R}^3$ into m-half spaces for more flexible prediction of the velocity fields (this point will be detailed in Sec. 3.2).

– The **identity map**, which connects successive building blocks of our ResNet-LDDMM allows the network to focus predicting the difference $\Phi^l - \Phi^{l-1}$, and thus time-dependent velocity fields. This specific architecture allows a forward Euler scheme to solve the non-stationary ODE (flow equation Eq. (2)). Intuitively, an individual building block f^l will move each point $x^l \in q^l$ into $x^{l+1} \in q^{l+1}$ such that passing the three successive connected layers results in

$$\begin{aligned} x^{l+1} - x^l &= f^l(x^l, \theta^l).\Delta^L \\ &= W_3^l(W_2^l(ReLU(W_1^l x^l + b_1^l) + b_2^l).\Delta^L \end{aligned} \tag{7}$$

Here W_i^l are the set of filters and b_i^l biases which compose the weight layers and ReLU is an element-wise activation function.

– The **loss function** $\mathcal{J}(\Theta)$ given in Eq. (6) to be minimized consists of a data term and the integration of instantaneous kinetic energy terms as a regularization term. While the data term pushes the deformed template $\Phi^L.q_S$ closer to the target q_T by minimizing an appropriate distance between point clouds, the regularization controls activities of the building blocks, i.e. their outputs. These terms are balanced with the $1/2\sigma^2$ parameter defined in Eq. (6).

– As a **data attachment** term, we use either a point-wise distance called the *Chamfer's distance (CD)* (Eq. (8)) or a global distance measure referred to us the *Earth mover's distance (EMD)*, the *Wasserstein distance* or the *Optimal Transport (OT) costs* (Eq. (9)). The former measures the squared distance between each point in one set to its nearest neighbor in the other set and vice verse. Mathematically, it is formulated as follows: Eq. (8),

$$\mathcal{D}^{\mathcal{CD}}(q_1, q_2) = \sum_{x \in q_1} \min_{y \in q_2} \|x - y\|^2 + \sum_{x \in q_2} \min_{y \in q_1} \|x - y\|^2. \quad (8)$$

The second distance introduced in [42], is measures the discrepancy between distributions when accounting for their respective geometries [43]:

$$\mathcal{D}^{\mathcal{MED}}(q_1, q_2) = \min_{\xi} \sum_{x \in q_1} \min_{x \in q_1, y \in q_2} \|x - \xi(y)\|^2, \quad (9)$$

where ξ denotes an arbitrary permutation over all $y \in q_2$ while $x \in q_1$ remain fixed and ξ^* is the optimal labeling minimizing Eq. (9). In practice, the iterative *Sinkhorn's Algorithm*[1] is used to approximate the EMD distance, i.e. an Optimal Transport with Entropic Constraints (the reader is directed to [44] and [45] for more details). Importantly, we note that both distances are differentiable almost everywhere [43]. The choice of distance used in the data term depends on the data itself and the nature of the deformations (also the application), as we will discuss in Sec. 4.

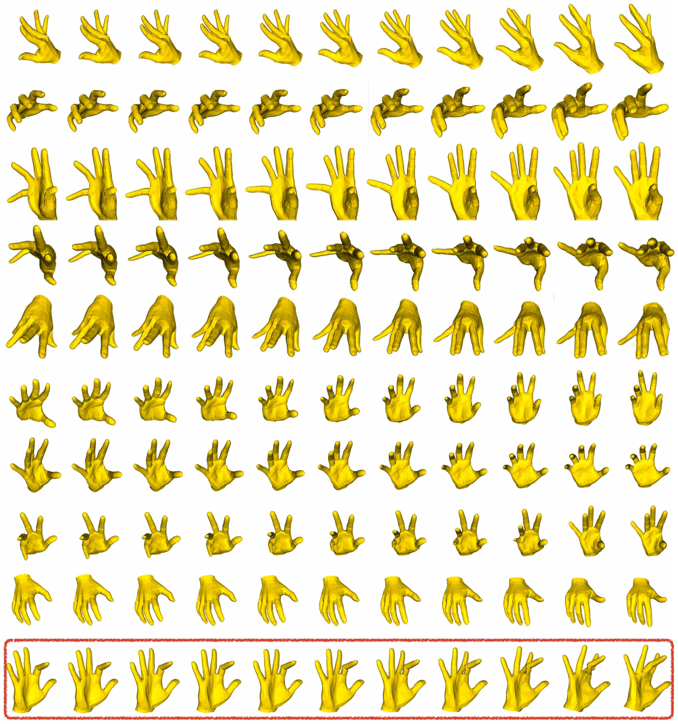

Fig. 3. Geodesic paths connecting source and target hand shapes modulo deformations (last row - index and middle fingers interchanged).

– Following LDDMM algorithms (e.g. [11] and [46]), our **regularizer** is a summation over all kinetic energies of the system at all time-steps $\mathcal{S}(f) = \frac{1}{2}\int_0^1 \mathcal{S}_t(f^t)dt$, where, $\mathcal{S}_t(f) = \|f^t(\Phi^{\Theta}(t,.), \theta^t)\|_{\ell_2}^2$. The main difference is that while LDDMM makes use of the $\|.\|_{\mathcal{A}}$ of the RKHS induced by Gaussian Kernel K with a fixed deformation scale, our ResNet-LDDMM uses the $\|.\|_{\ell_2}$ on the neural functions $f^l(.,\theta^l)$. Similar to LDDMM, ResNet-LDDMM generates geodesics by minimizing the amount of energy spent to get close to q_T from q_S, when traveling along the Orbit of q_S. We summarize the steps of our ResNet-LDDMM in Algo. 1.

Algorithm 1: ResNet-LDDMM.

```
Require: Source shape $q_S$, Target shape $q_T$, $L$: # of building
  blocks, $\eta$: learning rate, $\Delta^L = 1/L$: time-step, E:# of iter.,
  $m$: ResNet's width (# of filters), $\sigma$ (see Eq. (6)) .
  $l \leftarrow 1$, $e \leftarrow 0$, $\Phi^0 = I_{\mathbb{R}^3}$
  Set $\Theta = \{\theta^l\}_{l \in 1..L} \leftarrow \Theta_0$ (initial guess)
  Set $q^0 \leftarrow \Phi^0.q_S$ (initial state)
  while $e < E$ (epoch) do
    while $l < L$ (building block) do
      Compute $v^l \leftarrow f^l(\Phi^{l-1}.q^{l-1}, \theta^l)\Delta^L$
      Update $\Phi^l \leftarrow \Phi^{l-1} + f(\Phi^{l-1}.q^{l-1}, \theta^l)\Delta^L$
    end while
    Compute $q^L \leftarrow q^0 + \sum_{l=1}^{L} v^l$ ($\Phi^L \leftarrow \Phi^0 + \sum_{l=1}^{L} v^l$)
    Evaluate the loss function $\mathcal{J}(\Theta^e)$ (Eq. (6)) at epoch $e$
    Update $\Theta_{e+1} \leftarrow \Theta_e - \eta\nabla_{\Theta}\mathcal{J}(\Theta)$
  end while
  Compute $\Theta^*$ minimizer of Eq. (6) using ADAM.
Ensure: $\Phi^*(.,1) \leftarrow \Phi^L$ parameterized by $\Theta^*$: final
  transformation; $\{f^l\}_l$: flow of velocity fields.
```

We show in Fig. 2 both paths from a source shape (on the left) and a target shape (on the right). We illustrate the flows of velocity fields obtained for $L = 10$ intermediate time-steps. We can see also how close are the deformed sources to the target shapes (from the absolute spatial deviations). The examples involving hands, shown in Fig. 3, are particularly interesting (because of the nature of deformations possible for the hand) and challenging at the same time (as different neighboring fingers can exhibit opposite movements). Note that here we make use of $\mathcal{D}^{\mathcal{CD}}$ as the data attachment term. The last example, depicted in a red box, shows a case where ResNet-LDDMM fails at matching corresponding fingers. In fact, while the transformation is a diffeomorphism (by swapping the index and middle fingers), an incorrect registration result is obtained. Algo. 2 summarizes the steps for building a geodesic path connecting a source shape to a target shape using the output of Algo. 1.

Algorithm 2: Optimal Trajectory on the *Orbit* of q_S.

```
Require: Source shape $q_S$; Target shape $q_T$; $\sigma$ (Eq. (6)).
  $l \leftarrow 1$
  Compute $\Phi(1)$, $\{f^l\}_l$ using Algo. 1.
  while $l \leq L$ do
    Compute $\dot{q}^l \leftarrow f^l(., \theta^l)$ ($\dot{q}^l \in T_{q^l}([q_S])$: the
    instantaneous shape's speed, $[q_S]$: Shape Space)
    Update the shape $q^{l+1} \leftarrow q^l + \dot{q}^l$ (on $\mathcal{S}$)
  end while
Ensure: Discrete steps $q^l$, $l \in \{1, \ldots, L\}$ of an optimal
  trajectory connecting $q^0 \leftarrow q_S$ to $q^L \sim q_T$.
```

Now, to complete the picture and demonstrate the performance of ResNet-LDDMM in registering anatomical parts of the human body, we provide in Fig. 4 several examples of optimal deformations. Shapes are 3D triangulated

---

[1] We used the implementation available here.

surfaces obtained from manual segmentation of MRI images (of different subjects). These include the whole heart, the liver, the femur, a heart's valve, the brain cortex and left and right ventricles (LV+RV) of the heart. In each row, we show the source shape on the left, the target shape on the right and discrete steps from the geodesic path connecting them generated via ResNet-LDDMM and Algo. 2.

3 DYNAMIC AND GEOMETRIC ANALYSIS OF RESNET-LDDMM, AND HOW IT BUILDS DIFFEOMORPHISMS

3.1 Regularity of the Deformations

Each building block $f(\cdot, \theta^l)$ of ResNet-LDDMM can be seen as a vector field on $\mathbb{R}^3$, that is, a mapping $\mathbb{R}^3 \to \mathbb{R}^3$. This yields the transformations $\Phi^l : \mathbb{R}^3 \to \mathbb{R}^3$

$$\begin{aligned} &\forall x \in \mathbb{R}^3,\ l \in \{1, \ldots, L\}, \\ &\Phi^0(x) = x, \\ &\Phi^l(x) = \Phi^{l-1}(x) + f(\Phi^{l-1}(x), \theta^l).\Delta^L. \end{aligned} \tag{10}$$

Then, applying Φ^l separately to each point of the source shape q_0, we get the deformed shapes $q^l = \Phi^l.q_0$. Moreover, for a given $x \in q_0$, each f^l is actually computed explicitly (using Eq. (11)) by,

$$f(x, \theta^l) = w_3^l(w_2^l((w_1^l x + b_1^l)^+ + b_2^l)), \tag{11}$$

with w_i^l matrices whose entries are made up of the parameters θ^l, along with the real numbers b_j^l, and $r^+ = \max(r, 0) = ReLU(r)$, for every real number r, is the ReLU function. Hence, each $f(\cdot, \theta^l)$ is Lipshitz, and for every x, y in $\mathbb{R}^3$, we get

$$\begin{aligned} &\|f(x, \theta^l) - f(y, \theta^l)\|_2 \\ =&\left\| w_3^l w_2^l((w_1^l x + b_1^l)^+ - (w_1^l y + b_1^l)^+)\right\|_2 \\ \leq&\|w_3^l\|\|w_2^l\|\left\|((w_1^l x + b_1^l, 0)^+ - (w_1^l y + b_1^l, 0)^+)\right\|_2, \end{aligned} \tag{12}$$

where $\|.\|$ is the matrix operator norm. Then, since the ReLU function is 1-Lipshitz, that is, $\|a^+ - b^+\|_2 \leq \|a - b\|_2$ for every a, b, we get

$$\begin{aligned} &\|f(x, \theta^l) - f(y, \theta^l)\|_2 \\ \leq&\|w_3^l\|\|w_2^l\|\|w_1^l x + b_1^l - w_1^l y + b_1^l\|_2 \\ \leq&\|w_3^l\|\|w_2^l\|\|w_1^l\|\|x - y\|_2. \end{aligned} \tag{13}$$

Now Eq. (10) is clearly a Euler scheme at the times $t = \Delta^L$ for the flow of a differential equation of the form

$$\begin{aligned} &\forall x \in \mathbb{R}^3,\ t \in [0, 1], \\ &\Phi^0(x) = x \\ &\partial_t \Phi(t, x) = w_3(t) w_2(t)(\sigma(w_1(t)x + b_1(t)) + b_2(t)), \end{aligned} \tag{14}$$

with each coefficient of w_i and each b_j bounded in t. If we denote $f(t, x) = w_3(t) w_2(t)(\sigma(w_1(t)x + b_1(t)) + b_2(t))$ the time-dependent vector field whose integration yields $\Phi(t, x)$, we immediately see that for each t, $f : x \mapsto f(t, x)$ is Lipshitz with, for every t, x and y,

$$\begin{aligned} &\|f(t, x) - f(t, y)\|_2 \\ \leq&\|w_3(t)\|\|w_2(t)\|\|w_1(t)\|\|x - y\|_2 \\ \leq&\max_{t \in [0,1]} \|w_3(t)\|\|w_2(t)\|\|w_1(t)\|\|x - y\|_2, \end{aligned} \tag{15}$$

where $\|\cdot\|$ denotes the operator norm for linear operations, with the same reasoning as in Equation (13). The Cauchy-Lipshitz theorem ensures that Φ exists for every (t, x) and is (essentially) a diffeomorphism, or more precisely a bilipshitz transformation. In other words, ResNet-LDDMM does generate an approximation Φ^L of (essentially) a diffeomorphism $\Phi(1, \cdot)$ of $\mathbb{R}^3$, just like in LDDMM. As an additional remark, Gronwall's lemma shows that if we denote $C = \max_{t \in [0,1]} \|w_3(t)\|\|w_2(t)\|\|w_1(t)\|$, we have

$$\begin{aligned} &\|\Phi(t, x) - \Phi(t, y)\|_2 \leq \exp(tC)\|x - y\|_2, \\ &t \in [0, 1],\ x, y \in \mathbb{R}^3. \end{aligned} \tag{16}$$

Going back to ResNet-LDDMM, a quick induction can be used to show that, denoting $C(\Theta) = \max_{l \in \{1,\ldots,L\}}(\|w_3^l\|\|w_2^l\|\|w_1^l\|)$ with $\Theta = (\theta^1, \ldots, \theta^L)$ an instance of ResNet-LDDMM, we have

$$\begin{aligned} &\|f(x, \theta^l) - f(y, \theta^l)\|_2 \leq C(\Theta)\|x - y\|_2, \text{ and} \\ &\|\Phi^l(x) - \Phi^l(y)\|_2 \leq \exp(l\Delta^L C(\Theta))\|x - y\|_2, \\ &l \in \{1, \ldots, L\},\ x, y \in \mathbb{R}^3. \end{aligned} \tag{17}$$

The regularizer term in the cost function (Eq. (6)) then ensures that during the optimization, the constant $C(\Theta)$ does not go to infinity. As a final note, should smoothness be required,the ReLu can be replaced by a smooth approximation g with bounded differential (e.g. LeakyReLU). In this case, the velocity fields will be as smooth as g and the transformation Φ will then truly be a diffeomorphism. However, as shown in our various experiments, this does not appear necessary, and doing so intrinsically causes the loss of one of the strong points of our approach compared to classical LDDMM: the absence of a scale (see Sec. 4.2).

3.2 Geometric Description of the Velocity Fields and Interpretation

For fixed l, the vector fields $f^l(\cdot, \theta^l)$ given by the l-th building block have a nice geometric interpretation, which helps understand the way ResNet-LDDMM functions work and the areas in which it improves upon some of the classical LDDMM's weaknesses. For readability, we will omit the index l for now. Recall that m denotes the width of the network. For a fixed parameter θ of a building block $f(\cdot, \theta)$ of our network (i.e., a velocity field $x \mapsto f(x, \theta)$), the first layer is the affine transformation $L_1 : x \in \mathbb{R}^3 \mapsto w_1 x + b_1 \in \mathbb{R}^m$, with w_1 being an $m \times 3$ matrix and $b_1 = (b_{1,1}, \ldots, b_{1,m})$ in $\mathbb{R}^m$. Let us now note $n_1, \ldots, n_m \in \mathbb{R}^3$ the vectors whose transposes are the lines of w_1; that is, $w_1 = (n_1, \ldots, n_m)^T$. Then $L_1(x)^T = (n_1^T x + b_{1,1}, \ldots, n_m^T x + b_{1,m})$.

The second layer $L_2 : \mathbb{R}^m \to \mathbb{R}^m$ is a ReLU function, so that

$$\begin{aligned} L_2 \circ L_1(x)^T =&((L_2 \circ L_1)_1(x), \ldots, (L_2 \circ L_1)_m(x)) \\ =&((n_1^T x + b_{1,1})^+, \ldots, (n_m^T x + b_{1,m})^+) \end{aligned} \tag{18}$$

Geometrically, a vector $n \in \mathbb{R}^3$ and a number $b \in \mathbb{R}$ define an affine plane P of $\mathbb{R}^3$ through the equation $n^T x + b = 0,\ x \in \mathbb{R}^3$. This separates $\mathbb{R}^3$ into two half spaces: $E^+ = \{x \in \mathbb{R}^3,\ n^T x + b \geq 0\}$ on the side of P towards which n is pointing, and $E^- = \{x \in \mathbb{R}^3,\ n^T x + b < 0\}$, on the other side of P. Then, for any x in $\mathbb{R}^3$, we have that the function

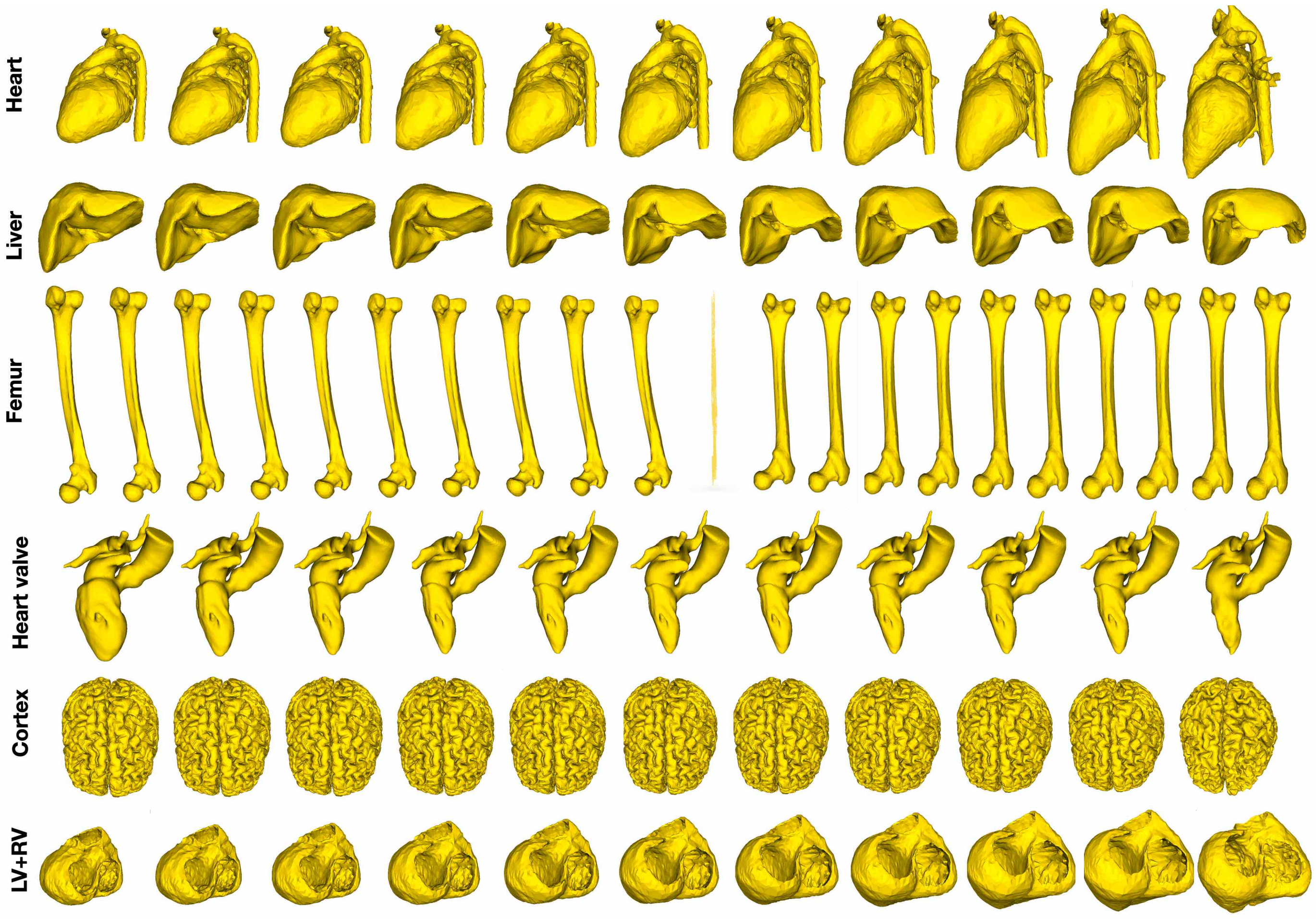

Fig. 4. Geodesic paths connecting source (first image) and target anatomical shapes (last image) of different subjects. From top to bottom: whole heart, liver, femur 1-2 (left) followed by femur 2-1 (right), heart valve and LV+RV (left and right ventricles of pathological hearts).

$$\begin{aligned} D_{n,b}(x) =& (n^T x + b)^+ \\ =& \begin{cases} n^T x + b \text{ if } x \in E^+, \\ \quad 0 \text{ if } x \in E^-. \end{cases} \end{aligned} \tag{19}$$

Then we notice that for $k = 1, \dots, m$, the k-th component of $L_2 \circ L_1$ is just $(L_2 \circ L_1)_k = D_{n_k, b_k}$.

Hence, the choice of parameters for the first two layers is equivalent to choosing m oriented affine planes $P_1, \dots, P_m$ with corresponding Cartesian equations $n_k^T x + b_k = 0$, $k = 1, \dots, m$, and half-spaces $E_k^{\pm}$, obtaining the corresponding components $D_k = D_{n_k, b_k}$. The third and fourth layers of the building block simply combine into an affine transformation $\mathbb{R}^m \to \mathbb{R}^3$. Therefore, each of the three components of the entire building block $f(., \theta) : \mathbb{R}^3 \to \mathbb{R}^3$ is just an affine combination of the functions D_k. In other words, for some vectors $a_1, \dots, a_m$ and c in $\mathbb{R}^3$, we have

$$f(x, \theta) = a_1 D_1(x) + \cdots + a_m D_m(x) + c. \tag{20}$$

Now, take a finite sequence of symbols $\epsilon = (\epsilon_1, \dots, \epsilon_m)$ with each ϵ_k being either $+$ or $-$, and consider the subset $T_\epsilon = \cap_{k=1,\dots,m} E_k^{\epsilon_k}$. The domain T_ϵ is a polytope, that is, a possibly unbounded polyhedron. Then, for every k such that $\epsilon_k = -$, $D_k(x) = 0$, and for every other k, $D_k(x) = n_k^T x + b_k$. Plugging this formula into Eq.(20), we get

$$\begin{aligned} f(x, \theta) =& \sum_{k, \epsilon_k = +} (\underbrace{a_k n_k^T}_{\text{3 by 3 matrices}} x + b_k) + c \\ =& \left(\sum_{k, \epsilon_k = +} a_k n_k^T \right) x + \sum_{k, \epsilon_k = +} b_k + c \\ =& A_\epsilon x + c_\epsilon. \end{aligned} \tag{21}$$

In other words, on each T_ϵ, $f(\cdot, \theta)$ is just an affine vector field. In conclusion, each building block of our network, that is, each velocity field we use to construct our final transformation, is just a globally continuous (even globally lipshitz), piece-wise affine mapping, defined on a partition by 2^m disjoint polytopes of $\mathbb{R}^3$.

The optimization of the final network, ResNet-LDDMM, can therefore be thought of as searching for:

1) The optimal partition of $\mathbb{R}^3$ into polytopes by m planes at each step
2) The correct affine transformation on each of these polytopes.

While there are obviously some additional constraints (for example, the transformations need to be globally continuous), this description gives a good representation of what

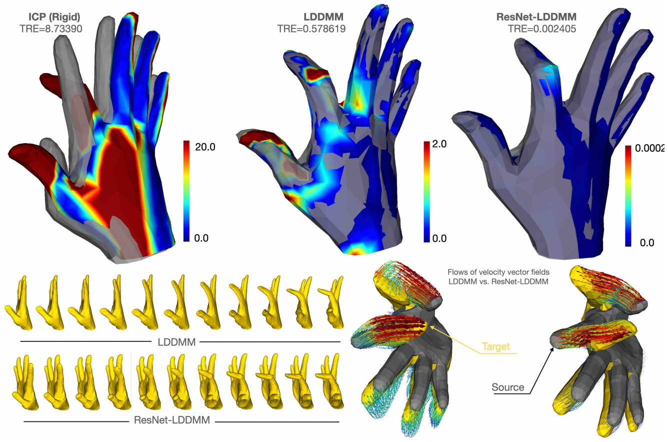

Fig. 5. LDDMM vs. ResNet-LDDMM – registration results of hand shapes with fingers moving in opposite directions.

ResNet-LDDMM does. As a result, different parts of the shape, even if they are close in $\mathbb{R}^3$ (such as two fingers on a hand) can be made to belong to different polytopes, and therefore be moved in opposite directions without costing too much energy (see Fig. 5).

4 IMPLEMENTATION AND EXPERIMENTS

In this section, we discuss both qualitative and quantitative registration results of our ResNet-LDDMM on different kinds of datasets. We compare some of our results to those obtained using a Varifold-type LDDMM implementation [13] (described in Sec. 1.2.1). As an ablation study, we will discuss the influence of the regularization term, the impact of the network width, i.e. the parameter m and the key role that the *ReLU* activation function plays inside the building blocks f^l, in Sec. 4.3. Next, we will study the robustness of ResNet-LDDMM to noise and missing data. Herein, we compare our results to state-of-the-art non-rigid registration techniques – N-ICP [30], RPTS [31], SVR-ℓ_0 [32] and QNS [33], previously reviewed in Sec. 1.2.3. First, we provide the implementation details of our framework.

4.1 Implementation Details

Our basic ResNet-LDDMM architecture consists of an ensemble of L=10 building blocks, thus, we consider 10 time-steps to solve the Flow Equation. Each generates a time (or block)-dependent field of velocity vector f^l. As discussed previously, each building block consists of three weight layers and only one ReLU layer transforming the output of the first weight layer. The width m of the first two layers is fixed to 900 (the width of the last layer is exactly 3 to ensure $f^l(.,\theta^l) : \mathbb{R}^3 \to \mathbb{R}^3$). The third weight layer could be viewed as a dimensionality reduction layer from $\mathbb{R}^m \to \mathbb{R}^3$ where our velocity fields naturally reside. Our initial guess (network parameters) $\Theta_0 = \{\theta_0^1, \ldots, \theta_0^L\}$ are obtained using *Xavier initialization*. Prior to our non-rigid registration method, a rigid alignment (which finds the best translation and rotation between q_S and q_T) using the well-known Iterative Closest Point algorithm is performed. All hand experiments reported in the paper are produced using the *Wasserstein distance* $\mathcal{D}^{\mathcal{MED}}$ as part of the loss function. This distance accounts for the geometry of the data and is thus suitable for large deformations. It is approximated using *Sinkhorn's algorithm*[2] (i.e. optimal transport with entropic constraints), as described in [44]. The main difficulty of approximating $\mathcal{D}^{\mathcal{MED}}$ is to find the best regularization parameter ($\epsilon^* = 8.10^{-6}$ is the best in the hands experiments) and we set the minimum steps in Sinkhorn's loop to 200. For remaining experiments, we used the *Chamfer's distance* $\mathcal{D}^{\mathcal{CD}}$ to evaluate the data term[3]. Finally, optimal parameters of our ResNet-LDDMM are computed using the ADAM optimizer with a learning rate $\eta = 10^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\beta_1 = 10^{-8}$. We consider $\sigma = 0.1$ in all experiments.

4.2 Qualitative Comparison to LDDMM

Both LDDMM and ResNet-LDDMM produce a topology-preserving transformation (i.e. a diffeomorphism) of the whole ambient space $\mathbb{R}^3$. However, there are several differences between the two make the ResNet approach both easier to run and more adapted to certain problems, particularly those involving very distinct motions in different areas of the considered shape. This is because LDDMM methods require choosing a certain "scale". For a discrete surface, when using a Gaussian kernel $K(x, y) = \exp(\frac{\|x-y\|_2}{2\sigma^2})$, the scale is σ, and every vector field generated will be a sum of Gaussian vector fields with fixed variance σ, each centered at a point on the shape (see [9] for example). As a result, moving points within distances less than σ in different directions requires a lot of energy, so such movements generally do not appear when minimizing the functional Eq. (3). However, the scale cannot be too small, or the motion will no longer need to preserve the topology of the triangulated surface. As a result, two main difficulties appear in LDDMM that the ResNet version improves upon.

– First, in LDDMM, one must find a "good" scale σ, which cannot be too small or too large, in order to allow a wide enough range of motions while still preserving the smoothness of the deformation. That is much less of a problem with ResNet-LDDMM: the size of the polytopes on which the velocity fields are computed not fixed, and so the algorithm automatically computes polytopes and polyhedrons of appropriate sizes. The role of scale falls instead on the width m of the network, However, we will see that the results are not particularly sensitive to a change of width less than an order of magnitude, and simply choosing a width of a few hundred yielded good results on every case we tested (see Sec. 4.3). We show in Fig. 7 both results of LDDMM and ResNet-LDDMM operating on pathological hearts (last row of Tab. 2). We can clearly see the qualitative superiority of our method in terms of deformations of the source shape to fit the target. It should be noted that this may simply be because we did not find the ideal scale for the LDDMM-matching, but that leads us to another, very different aspect in which ResNet-LDDMM seems superior: the computational time. On a CPU, each simulation of LDDMM takes a couple hours, while the ResNet version converges in minutes. Further there is no automated way to find the best scale, so finding the correct one for LDDMM simply requires testing several values (with a bit of intuition to avoid bad

[2] We use the CPU implementation available here.

[3] We use the GPU implementation available here.

TABLE 1
Key differences between the original LDDMM framework and our ResNet-LDDMM.

Algorithm	**LDDMM** (Large Deformation Diffeomorphic Metric Mapping)	**ResNet-LDDMM** (or LDDMM Revisited)
Problem	Solve the **Flow Equation** (ODE) $\partial_t \phi(t,.) = v(t, \phi(t,.))$ s.t. at $t = 0$, $\phi(0,.) = q_S$ and $\phi(1,.)$ is a **Diffeomorphism**.	
Paradigm	optimize-then-discretize (Hamiltonian formulation)	discretize-then-optimize (Euler discretization)
Admissible space	$(\mathcal{A}, \|.\|_{\mathcal{A}})$: predefined RKHS induced by the chosen scale	functional space of NNs $f^l(., \theta^l)$
Kinetic energy	$\mathcal{S}(v) = \frac{1}{2}\int_0^1 \langle \mathcal{A}v, v\rangle_{\mathcal{A}} dt = \frac{1}{2}\int_0^1 \|v\|^2_{\mathcal{A}} dt$	$\mathcal{S}(f) = \frac{1}{2}\sum_{l,i} \|f^l(x_i, \theta^t)\|^2$
Optimization	Pontryagin's Maximum Principle (PMP) then GD	Backpropagation (BP) with ADAM
Velocity vectors	Generated by convolution with the kernel	Piece-wise affine in multiple polytopes
Parameter of the model	Positive parameter σ (not robust)	Width m (robust)
Best on	Shapes with important features on a single scale	Shapes can be of varying scale
Computational cost	Large computational time	Efficient computational time

ones). Hence, finding the best scale can sometimes take several days, and so we often settle one that is simply "good enough". Whether the results of Tab. 2 are due to not finding the correct scale, or to our ResNet beating the best scale is up to debate, but regardless the ResNet results were obtained on the first try, which for practical applications is a clear improvement. For synthetic heart, we report the results of rigid ICP instead of LDDMM as the high data resolution do not allow such simulations.

– Second, there may not even be a good scale in the LDDMM framework to compare certain shapes. This is generally caused by points belonging to very different parts of the shape that also happen to be very close, such as two fingers on a hand. Such matching problems are generally much harder for the usual LDDMM, because one needs to choose big enough scales of deformation in the reproducing kernel to ensure regularity of the shape. This can either prevent necessary deformations from occurring, or force the transformation to first separate and unnaturally expand the fingers during the first half of the transformation. This is also a well-known problem when tackling multiple shapes (see [9] for example). This problem is particularly obvious in Fig. 6 when matching various hand positions, especially (but not only) in the intermediate deformation steps. On the other hand, thanks in part to only requiring bi-lipshitz regularity, ResNet-LDDMM can move two parts of the shape (such as two fingers) in opposite directions while keeping the regularization term small by encasing each part within distinct polytopes. Fig. 5 illustrates how ResNet-LDDMM performs better than LDDMM for the case of a hand. That is, the True Registration Error (see Sec. 4 for definition) is much lower, and the geodesic much more plausible in the case of ResNet-LDDMM compared to LDDMM.

– As a final difference between the methods, LDDMM is quadratic with respect to the resolution (i.e., the number of points), and is notoriously time-consuming to run. While matching two shapes usually takes a acceptable amount of time, matching hundreds (or even thousands) for adequate statistical analysis can quickly become impossible. Significant advances have been made using GPUs and the free library KeOps[4] on this front [47], but it is still time-consuming to perform such an analysis, especially when one needs several retries to find the correct scales for the reproducing kernel. On the other hand, ResNet-LDDMM is much faster, especially at high resolution; the network itself actually has linear complexity with respect to the resolution. Only the data attachment terms have quadratic complexity. Combined with the lack of needing compute a correct scale, this new method seems like a clear winner in this area (at least for more than a few hundred points). Tab. 1 summarizes the key differences between LDDMM and ResNet-LDDMM.

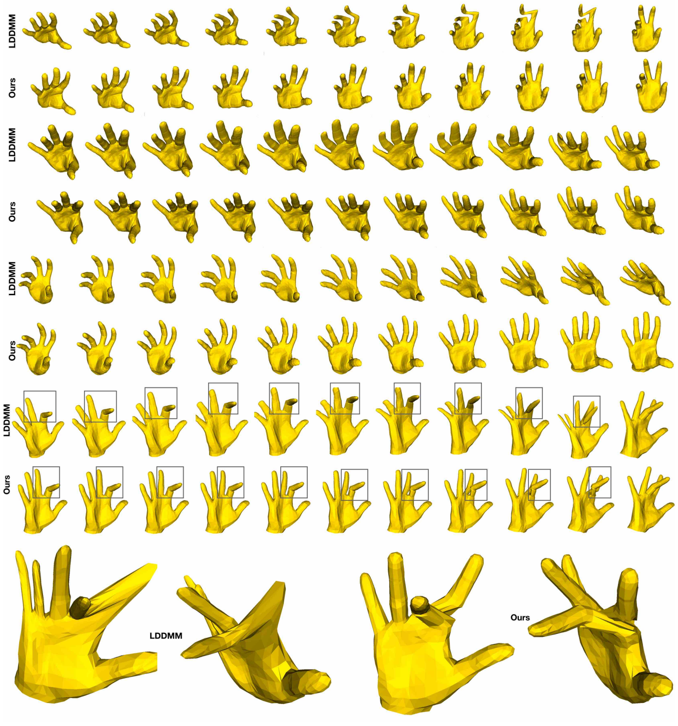

Fig. 6. Four examples to compare ResNet-LDDMM (ours) to LDDMM.

4.3 Ablation Study

Here, we study some important features of ResNet-LDDMM. These are the width m of the building blocks (which also defines the number of network parameters), the key role that the ReLU activation function play and the influence of the regularizer on the registration. In the following experiments, we evaluate different parameters using the *Average Target Registration Error (TRE)*, reported in [33] as the RMSE (for Root Mean Square Error). It is given by Eq. (22):

$$\mathrm{TRE} = \left(\frac{\sum_{x_i \in Q_s, \xi(x_i) \in Q_t} \|x_i - \xi(x_i)\|^2}{|Q_s|} \right)^{1/2}, \qquad (22)$$

---

[4] www.kernel-operations.io

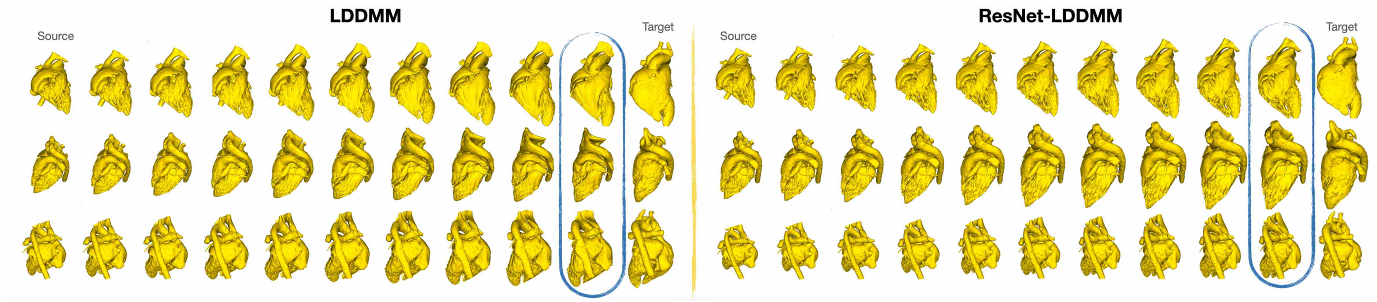

Fig. 7. Three viewpoints of geodesic paths generated by LDDMM (left) and ResNet-LDDMM (right) between pathological hearts (last row of Tab. 2).

where $\xi(.)$ is the ground truth registration of each point from the point set Q_t to physically match a point in Q_s.

Impact of the Network's Width m

We study in Fig. 8 registration results on hand shapes with large deformations separating them when varying our Network's width m (m is also the width of each building block). It is clear from the figure that the higher is the width m, the lower is the TRE, which indicates accurate registration. Indeed, as the width m of each building block is taken to be large, the distribution over functions f^l tends to converge towards a Gaussian process. On can appreciate the smoothness of generating velocity fields from the second row of Fig. 8. However, due to the freedom given by the width m, some irregular (with respect to the neighborhood) deformations appear in the deformed source. In fact, the number of polytopes, in which our velocity fields are predicted, depends on the parameter m. Thus, a wide-network isn't necessarily the best choice. Accordingly, in remaining experiments, we set $m^* = 900$ the number of neurons in the first two layers, which provides an acceptable trade-off to obtain smooth flows of velocity fields.

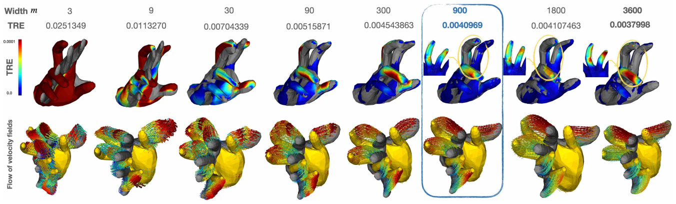

Fig. 8. Impact of the network width m. We vary the value of m from 3 to 3600, corresponding TREs and generated flows of velocity fields.

Influence of the ReLU Activation Function

As we have stated before, the ReLU activation function plays a key role in achieving correct transformations. This is illustrated in the three panels of Fig. 9, where different registration results are reported for ReLU, LeakyRelu and Tanh activation functions. We recall that the activation function is element-wisely applied to the output of the first layer of each building block. It is clear that while the ReLU family allows computing a good transformation and achieve accurate registration, Tanh doesn't (It is also the case the in absence of activation function). ReLU activation function not only dictates how the space needs to be divided to compute the velocity fields (for a different phalanx of the fingers, in this case), but it also improves the optimization (by speeding up the training) as it forces several outputs (of the first layer) to be zero. This makes the ReLU activation function a key ingredient in our ResNet-LDDMM architecture. In our experiments, we make use of the LeakyReLU version.

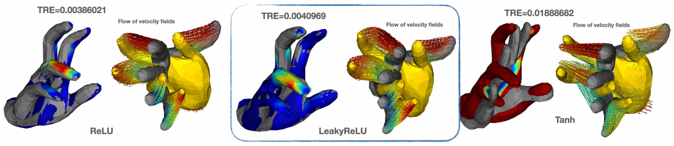

Fig. 9. Influence of the ReLU activation function on registration results: From left to right: ReLU, LeakyReLU and Tanh (hyperbolic tangent).

Role of the Regularizer

We end our ablative study by revealing the role of the regularizer in our joint geometric-neural network framework, ResNet-LDDMM. We recall that following the LDDMM formulation, our *regularizer* is also the summation of all the kinetic energy of the system at all time-steps, i.e. $\mathcal{S}(f) = \frac{1}{2}\int_0^1 \mathcal{S}_t(f^t)dt$, where $\mathcal{S}_t(f) = \|f^t(\phi(t,.),\theta^t)\|_2^2$. To make this clear, we vary the parameter σ in Eq. (6), where $1/2\sigma^2$ is the weight of the data term, and we report registration results as shown in Fig. 10.

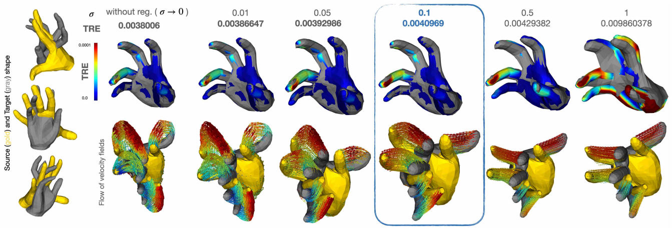

Fig. 10. Role of the Regularizer (or parameter σ) in ResNet-LDDMM.

We recall that σ and the weight given to the data term vary inversely (thus, the higher σ, the higher the regularizer contribution). It is clear in Fig. 10 that our regularizer reduces the *kinetic energy* of the whole system while trying to connect the source to the target. That is, for all values of σ, the final transformation is plausible with quite regular and smooth time-dependent velocity fields. However, an optimal choice σ^* allows a minimizing path to be found and enables $\Phi(1).q_S$ to be in the neighborhood of q_T. Accordingly, we set $\sigma^* = 0.1$ as it guarantees, in general, a

TABLE 2
Liver and heart registration – TRE (in millimeters) on the *BPSM in silicon liver dataset* [48] and *Synthetic Cardiac dataset* from [49].

Deformed Liver	LDDMM	**ours** ($\mathcal{D}^{\mathcal{CD}}$)	**ours** ($\mathcal{D}^{\mathcal{MED}}$)
#1 → #1-Def.	**0.35327**	0.386471	1.322229
#2 → #2-Def.	**0.28004**	0.374769	2.322149
#3 → #3-Def.	**0.25295**	0.396076	3.737282
Synthetic 4D Heart	Rigid (ICP)	**ours** ($\mathcal{D}^{\mathcal{CD}}$)	**ours** ($\mathcal{D}^{\mathcal{MED}}$)
F#8 → F#0	9.372447	**0.826361**	2.181144
F#16 → F#0	6.747137	**0.986380**	2.566208
F#24 → F#0	2.942818	**0.662351**	1.972288
F#33 → F#0	1.320668	**0.469837**	1.868704
Segmented Heart	LDDMM	**ours** ($\mathcal{D}^{\mathcal{CD}}$)	**ours** ($\mathcal{D}^{\mathcal{MED}}$)
S#5 → S#8	3.10755	**2.38700**	3.004888

good balance between a low kinetic energy and an accurate registration.

4.4 Anatomical Registration Experiments

We mainly consider the datasets released in [48] and [49]. The first one, contains three liver models that have been deformed by means of a non-linear bio-mechanical model. For each model, the original mesh as well as the deformed mesh are provided. From the second dataset, we select the sequence of a deforming healthy heart (generated from the STACOM2011 Challenge data[5] using the model described in [49]). In Tab. 2, we report the TREs for the three pairs of livers (original and deformed counterparts, i.e. #i → #iDef). We also report the TREs of registering frames F#8, F#16, F#24 and F#33 to F#0, taken from the a whole pumping cycle of a healthy heart. We notice similar results achieved by ResNet-LDDMM and LDDMM on these particular examples. The LDDMM implementation applied here consists of a Varifold-type data term [13] with scale 4, with $\sigma = 0.25$ and a scale in the Gaussian kernel equal to 8. One can clearly see that on anatomical shapes, $\mathcal{D}^{\mathcal{CD}}$ in the data term achieves better results than $\mathcal{D}^{\mathcal{MED}}$. We report TREs of three methods on segmented hearts (Fig. 7).

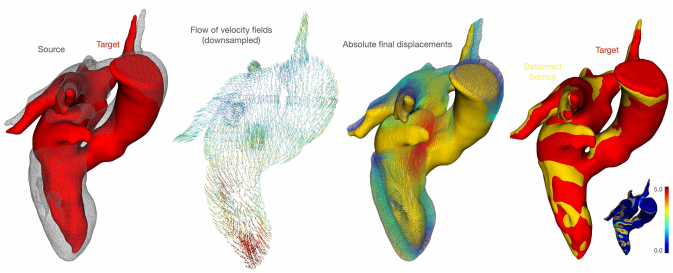

Fig. 11. Registration of heart valves (data from *fshapes* Toolkit).

Through Fig. 11 and Fig. 12 we provide some qualitative evaluations of our ResNet-LDDMM framework operating on complex anatomical shapes (a heart valve and a cortex). The outputs of our algorithm exhibit smooth and regular velocity vector fields. Final absolute displacements to reach the target from the source are also reported (arrow colors indicate the amount of displacement in $\mathbb{R}^3$). The very challenging problem of registering cortex shapes can be noted from Fig. 12.

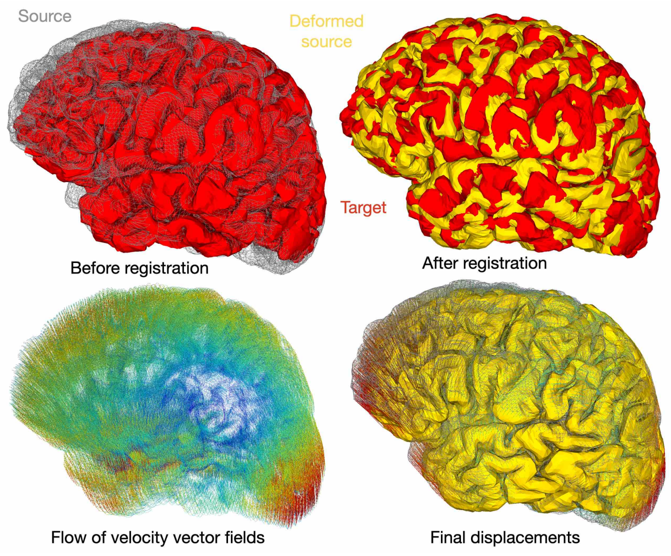

Fig. 12. Registration of Cortex shapes (data from the *fshapes* Toolkit).

4.5 Human Body Registration Experiments

We come now to a second kind of comparison involving human body shapes and a set of state-of-the-art non-rigid registration techniques, N-ICP [30], RPTS [31] and QNS [33] operating on 3D point clouds. We point out that, while our ResNet-LDDMM does not require any correspondence initialization, these techniques **need a good initialization to work properly**. In the interesting study presented in [33], initialization is often achieved by providing several pairs of corresponding feature points, determined either using the SHOT feature, or through manual labeling. For fair comparison, we use the same data from [50]. Due to the high resolution of the meshes, we make use of the *Chamfer's distance* $\mathcal{D}^{\mathcal{CD}}$ (Eq. (8)) to evaluate the data term. We notice that, for our LDDMM formulation, the transformed source will fall to a point in the near neighborhood of the target shape for a given error tolerance defined by the data attachment term. This is important as we do not expect better accuracy, i.e. lower TREs than those achieved by non-rigid registration techniques. Accordingly, in the following tables, our TREs are highlighted in bold when they are found to be lower than of those of the other approaches.

Registration Modulo Small and Large Deformations

First, we report the registration results of our ResNet-LDDMM in the presence of both small and large deformations between the source shape and the target shape. A nearest neighbor initialization was provided for N-ICP, RPTS, SVR-ℓ_0 and QNS to solve the small deformation problem. In contrast, large deformations use SHOT and diffusion pruning to get initial correspondence (we refer the reader to [33] for the optimal parameters of these algorithms). In both experiments ResNet-LDDMM achieves competitive results without requiring any initial corresponding pairs. Another

---

[5] http://www-sop.inria.fr/asclepios/data/STRAUS/meshes/

important difference is that, while ResNet-LDDMM is a Riemannian-like framework with a propor distance to compare shapes and algorithms to compute geodesics, non-rigid registration techniques cited above does not. They do not even guarantee diffeomorphic transformation.

TABLE 3
Small deformations – Comparison to N-ICP [30], RPTS [31], SVR-ℓ_0 [32] and QNS [33] - TREs (in meters) on examples taken from [50].

Pair	N-ICP	RPTS	SVR-ℓ_0	QNS	**ours** ($\mathcal{D}^{\mathcal{CD}}$)
#1 (crane)	0.0083	0.0113	**0.0015**	0.0020	**0.00599**
#2 (march)	0.0160	**0.0022**	0.0090	0.0117	**0.00989**
#3 (samba)	**0.0008**	0.0031	0.0022	0.0021	**0.00257**
#4 (squat)	0.0044	0.0031	0.0044	**0.0019**	**0.00311**
#5 (swing)	0.0093	0.0095	0.0078	**0.0076**	0.01091

TABLE 4
Large deformations – Comparison to N-ICP [30], RPTS [31] and QNS [33] - TREs (in meters) on examples taken from [50].

Pair	N-ICP	RPTS	SVR-ℓ_0	QNS	**ours** ($\mathcal{D}^{\mathcal{CD}}$)
#1 (crane)	0.0105	0.0124	–	**0.0041**	0.045134
#2 (crane)	0.0196	0.0204	–	**0.0093**	0.049671
#3 (swing)	0.0107	0.0108	–	**0.0029**	0.06033
#4 (swing)	0.0134	0.0117	–	**0.0058**	0.05118

The behavior of non-rigid registration techniques under the presence of noise and outliers in the target shape as well as their ability to recover missing data are important considerations. In the next paragraphs we report some results regarding these aspects and compare with our framework.

Robustness to Noise in the Target

Following the interesting experiments reported in [51] where a Gaussian noise is added, either sparsely or densely, to the target surface, taken from the "jumping" subset of [50], we report our registration results compared with N-ICP, RPTS and QNR [51]. In Tab. 5, we consider both experiments with dense noise (with two different variance [51]) and sparse noise (5% or 50% of vertices affected).

TABLE 5
Robustness to noise – Comparison to N-ICP [30], RPTS [31], and QNS [33] in terms of RMSE (in meters) on examples from [50].

Exp.	N-ICP	RPTS	QNS	**ours** ($\mathcal{D}^{\mathcal{CD}}$)
5% (sparse)	0.0785	0.0076	0.0079	**0.0531**
50% (sparse)	0.0954	0.0269	**0.0258**	**0.0636**
100% (dense)	0.0473	0.0184	**0.0159**	**0.0354**
100% (dense)	0.0680	**0.0178**	0.0211	**0.0288**

As a mandatory initialization step, N-ICP, RPTS and QNS use SHOT and diffusion pruning for the first and third rows and 60 manually selected labeled landmarks for reaming experiments. Again, these experiments confirm the competitiveness of ResNet-LDDMM. Fig. 13 zooms on our result reported in the second row of Tab.5. It shows the geodesic path and the deformed shape (end point). One can appreciate how smooth and noiseless is the deformed source, while the target shape is very noisy.

Robustness to missing data in the Target

Here, we study the behavior of our approach to missing data in the target. We report in Tab. 6 ResNet-LDDMM results compared to N-ICP, RPTS, SVR-ℓ_0 and QNR. The original models are taken from the "bouncing" dataset of [50]. Target models have been built by removing some parts from the original model (an example is provided in Fig. 14, right bottom corner).

TABLE 6
Recovering missing data – Comparison to N-ICP [30], RPTS [31], SVR-ℓ_0 [32] and QNS [33] - RMSE (in meters) on examples from [50].

Pair	N-ICP	RPTS	SVR-ℓ_0	QNS	**ours** ($\mathcal{D}^{\mathcal{CD}}$)
#1	0.0197	0.0255	0.0307	**0.0116**	**0.0186**
#2	0.0181	0.0226	0.0214	**0.0107**	**0.0194**
#3	**0.0958**	0.0982	0.1026	0.1009	0.1019
#4	0.0203	0.0291	0.0262	**0.0165**	**0.0198**
#5	0.0430	0.0441	0.0501	**0.0306**	**0.0474**
#6	0.0721	0.0663	0.0776	0.0615	**0.0389***
#7	0.0478	0.0448	0.0460	0.0322	**0.0262***

Fig. 14 focuses on the pair #6 given in Tab. 6. It shows on the top panels different TREs of state-of-the-art registration techniques (N-ICP, RPTS, SVR-ℓ_0, and QNS) in addition to our ResNet-LDDMM. At the bottom, we show the geodesic path generated using Algo. 2. The superiority of ResNet-LDDMM in the examples with (*) demonstrates its ability to deal with missing data in the target shape thanks to the shape prior preserved along the geodesic path when fitting to the target. This is visible on the right leg where ResNet-LDDMM achieves less errors. Despite the obscured data located in the higher part of the target shape, errors for all methods, except ResNet-LDDMM, are mainly located on the right leg. This is mainly due the diffeomorphic registration guaranteed by our method.

5 CONCLUSION AND FUTURE DIRECTIONS

Grounding on the elegant LDDMM framework for shape registration, combined with powerful deep residual neural networks, we have proposed a novel joint geometric-neural network Riemannian-like framework for diffeomorphic registration of 3D shapes. Our registration schema is completely unsupervised, i.e. only source and target shapes are needed. A two-level regularization process was introduced, first by the network's structure and induced high-dimensional functional spaces and second by minimizing the time-integrated kinetic energy. This allows the deformed shape to move along its orbit while tolerating an error defined by a data attachment term. Our ResNet-LDDMM geometrically builds time-dependent vector fields by first finding the optimal partition of the space into polytopes, and then predict the correct affine transformation on each of these polytopes. Several experimental illustrations involving both internal anatomical shapes and external human body shapes illustrate the ability of our ResNet-LDDMM framework computing diffeomorphic registrations. We believe that the present research builds an interesting first bridge between the purely geometric, well founded, LDDMM Riemannian framework and deep neural networks. Several aspects of our ResNet-LDDMM have nice geometric interpretations and, inversely, suitable geometric operations

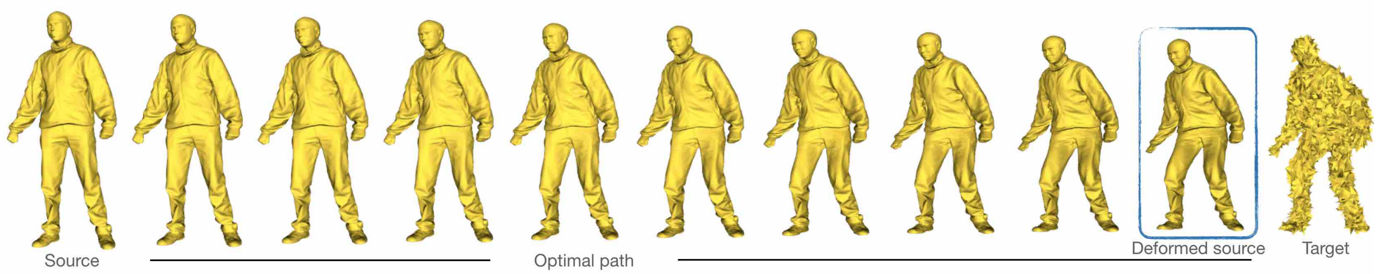

Fig. 13. Geodesic path connecting a source human shape to a deformed shape with a noisy human body shape as a target (last column).

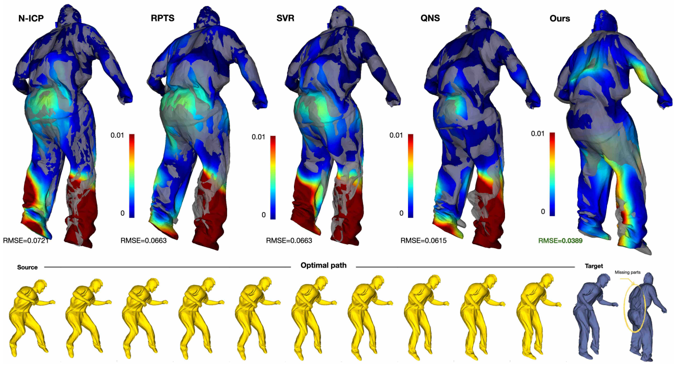

Fig. 14. Robustness to missing data in the target shape. Comparison to N-ICP, RPTS, SVR-ℓ_0, QNS and a geodesic connecting source and target generated by our ResNet-LDDMM (Algo. 2).

are achieved by our network. Our framework also opens the door to designing more efficient solutions in computational anatomy, statistical shape analysis and medical image registration. We leave for future investigation the connection of our ResNet-LDDMM with Kernel Methods through the recent theoretical study of networks with infinite-width.

REFERENCES

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

[2] 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*, vol. 61, no. 2, pp. 139–157, 2005.

[3] F.-X. Vialard, L. Risser, D. Rueckert, and C. J. Cotter, "Diffeomorphic 3d image registration via geodesic shooting using an efficient adjoint calculation," *International Journal of Computer Vision*, vol. 97, no. 2, pp. 229–241, 2012.

[4] P. Dupuis, U. Grenander, and M. I. Miller, "Variational problems on flows of diffeomorphisms for image matching," *Quarterly of applied mathematics*, pp. 587–600, 1998.

[5] L. Younes, *Shapes and diffeomorphisms*, vol. 171. Springer, 2010.

[6] C. BROIT, "Optimal registration of deformed images," *Doctoral Dissertation, University of Pennsylvania*, 1981.

[7] A. Trouvé, "An infinite dimensional group approach for physics based models in pattern recognition," *preprint*, 1995.

[8] T. Polzin, M. Niethammer, F.-X. Vialard, and J. Modersitzki, "A discretize–optimize approach for lddmm registration," in *Riemannian Geometric Statistics in Medical Image Analysis*, pp. 479–532, Elsevier, 2020.

[9] S. Arguillère, E. Trélat, A. Trouvé, and L. Younes, "Registration of multiple shapes using constrained optimal control," *SIAM J. Imaging Sci.*, vol. 9, no. 1, pp. 344–385, 2016.

[10] S. Arguillere, M. I. Miller, and L. Younes, "Diffeomorphic surface registration with atrophy constraints," *SIAM Journal on Imaging Sciences*, vol. 9, no. 3, pp. 975–1003, 2016.

[11] L. Younes, F. Arrate, and M. I. Miller, "Evolutions equations in computational anatomy," *NeuroImage*, vol. 45, no. 1, pp. S40–S50, 2009.

[12] B. Gris, S. Durrleman, and A. Trouvé, "A sub-riemannian modular framework for diffeomorphism-based analysis of shape ensembles," *SIAM Journal on Imaging Sciences*, vol. 11, no. 1, pp. 802–833, 2018.

[13] N. Charon and A. Trouvé, "The varifold representation of nonoriented shapes for diffeomorphic registration," *SIAM Journal on Imaging Sciences*, vol. 6, no. 4, pp. 2547–2580, 2013.

[14] M. I. Miller, A. Trouvé, and L. Younes, "Hamiltonian systems and optimal control in computational anatomy: 100 years since d'arcy thompson," *Annual review of biomedical engineering*, vol. 17, pp. 447–509, 2015.

[15] M. I. Miller, D. J. Tward, and A. Trouve, "Coarse-to-fine hamiltonian dynamics of hierarchical flows in computational anatomy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 860–861, 2020.

[16] J. Modersitzki, *FAIR: flexible algorithms for image registration*. SIAM, 2009.

[17] A. Mang and L. Ruthotto, "A lagrangian gauss–newton–krylov solver for mass-and intensity-preserving diffeomorphic image registration," *SIAM Journal on Scientific Computing*, vol. 39, no. 5, pp. B860–B885, 2017.

[18] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache, "A log-euclidean framework for statistics on diffeomorphisms," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 924–931, Springer, 2006.

[19] H. Kobatake and Y. Masutani, *Computational anatomy based on whole body imaging*. Springer, 2017.

[20] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *arXiv preprint arXiv:1506.02025*, 2015.

[21] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning for fast probabilistic diffeomorphic registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 729–738, Springer, 2018.

[22] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning of probabilistic diffeomorphic registration for images and surfaces," *Medical image analysis*, vol. 57, pp. 226–236, 2019.

[23] J. Krebs, H. Delingette, B. Mailhé, N. Ayache, and T. Mansi, "Learning a probabilistic model for diffeomorphic registration," *IEEE transactions on medical imaging*, vol. 38, no. 9, pp. 2165–2176, 2019.

[24] T. C. Mok and A. Chung, "Fast symmetric diffeomorphic image registration with convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4644–4653, 2020.

[25] A. Bône, M. Louis, O. Colliot, S. Durrleman, A. D. N. Initiative, *et al.*, "Learning low-dimensional representations of shape data sets with diffeomorphic autoencoders," in *International conference on information processing in medical imaging*, pp. 195–207, Springer, 2019.

[26] Z. Shen, X. Han, Z. Xu, and M. Niethammer, "Networks for joint affine and non-parametric image registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4224–4233, 2019.

[27] M.-M. Rohé, M. Datar, T. Heimann, M. Sermesant, and X. Pennec, "Svf-net: Learning deformable image registration using shape

matching," in *International conference on medical image computing and computer-assisted intervention*, pp. 266–274, Springer, 2017.

[28] M. Niethammer, R. Kwitt, and F.-X. Vialard, "Metric learning for image registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8463–8472, 2019.

[29] Z. Shen, F.-X. Vialard, and M. Niethammer, "Region-specific diffeomorphic metric mapping," *arXiv preprint arXiv:1906.00139*, 2019.

[30] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid icp algorithms for surface registration," in *2007 IEEE conference on computer vision and pattern recognition*, pp. 1–8, IEEE, 2007.

[31] K. Li, J. Yang, Y.-K. Lai, and D. Guo, "Robust non-rigid registration with reweighted position and transformation sparsity," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 6, pp. 2255–2269, 2018.

[32] K. Guo, F. Xu, Y. Wang, Y. Liu, and Q. Dai, "Robust non-rigid motion tracking and surface reconstruction using l0 regularization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3083–3091, 2015.

[33] Y. Yao, B. Deng, W. Xu, and J. Zhang, "Quasi-newton solver for robust non-rigid registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[34] M. I. Miller, "Computational anatomy: shape, growth, and atrophy comparison via diffeomorphisms," *NeuroImage*, vol. 23, pp. S19–S33, 2004.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, IEEE Computer Society, 2016.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, pp. 630–645, Springer, 2016.

[37] E. Weinan, "A proposal on machine learning via dynamical systems," *Communications in Mathematics and Statistics*, vol. 5, no. 1, pp. 1–11, 2017.

[38] E. Haber and L. Ruthotto, "Stable architectures for deep neural networks," *Inverse Problems*, vol. 34, no. 1, p. 014004, 2017.

[39] L. Ruthotto, S. J. Osher, W. Li, L. Nurbekyan, and S. W. Fung, "A machine learning framework for solving high-dimensional mean field game and mean field control problems," *Proceedings of the National Academy of Sciences*, vol. 117, no. 17, pp. 9183–9193, 2020.

[40] G.-H. Liu and E. A. Theodorou, "Deep learning theory review: An optimal control and dynamical systems perspective," 2019.

[41] M. Lin, Q. Chen, and S. Yan, "Network in network," 2014.

[42] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000.

[43] J. Feydy, B. Charlier, F.-X. Vialard, and G. Peyré, "Optimal transport for diffeomorphic registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 291–299, Springer, 2017.

[44] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), vol. 26, pp. 2292–2300, Curran Associates, Inc., 2013.

[45] L. Chizat, P. Roussillon, F. Léger, F.-X. Vialard, and G. Peyré, "Faster wasserstein distance estimation with the sinkhorn divergence," 2020.

[46] M. I. Miller, A. Trouvé, and L. Younes, "Geodesic shooting for computational anatomy," *Journal of mathematical imaging and vision*, vol. 24, no. 2, pp. 209–228, 2006.

[47] B. Charlier, J. Feydy, J. A. Glaunès, F.-D. Collin, and G. Durif, "Kernel operations on the gpu, with autodiff, without memory overflows," 2020.

[48] S. Suwelack, S. Röhl, S. Bodenstedt, D. Reichard, R. Dillmann, T. dos Santos, L. Maier-Hein, M. Wagner, J. Wünscher, H. Kenngott, *et al.*, "Physics-based shape matching for intraoperative image guidance," *Medical physics*, vol. 41, no. 11, p. 111901, 2014.

[49] N. Duchateau, M. Sermesant, H. Delingette, and N. Ayache, "Model-based generation of large databases of cardiac images: synthesis of pathological cine mr sequences from real healthy cases," *IEEE transactions on medical imaging*, vol. 37, no. 3, pp. 755–766, 2017.

[50] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," in *ACM SIGGRAPH 2008 papers*, pp. 1–9, 2008.

[51] Y. Yao, B. Deng, W. Xu, and J. Zhang, "Quasi-newton solver for robust non-rigid registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7600–7609, 2020.