A Comparison of Reduced Order Models and Fourier Neural Operators for Geodesic Shooting in Diffeomorphic Registration

Carlos Paesa-Lia[1], Álvaro de la Asunción, Mónica Hernández[1]

[1] Affiliation: T64_23R COSMOS, Computer Science for Complex Systems Modelling
Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.
Tel. +34-976762707, e-mail: *cpaesa@unizar.es*

Abstract

Geodesic shooting in the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework is computationally expensive due to the complex dependence of the image similarity energy and the initial velocity field driving the shooting. This work compares two acceleration strategies: Proper Orthogonal Decomposition and Fourier Neural Operators for learned neural surrogates.

Introduction

Diffeomorphic image registration aims to find a smooth and invertible transformation that aligns a moving image with a fixed one. The LDDMM framework formulates this problem as a path on the diffeomorphism group $Diff(\Omega)$, in which the deformation is parameterized by a time-dependent velocity field $v(t)$ belonging to a reproducing kernel Hilbert space V. If we only consider minimum length paths (geodesics), the whole geodesic path is determined by an initial velocity v_0 via the Euler–Poincaré differential equation (EPDiff) [1].

The numerical cost of integrating the EPDiff equation and evaluating the gradient with respect to v_0 is dominated by the complex relationship between energy and initial velocity. We study two different approaches to mitigate it: classical model-order reduction by Proper Orthogonal Decomposition (POD) [2], which restricts the dynamics to a low-dimensional linear subspace built from data; and using neural surrogates such as NeurEPDiff [3], which learns the EPDiff flow offline with a Fourier Neural Operator (FNO) to then quickly produce its solution from the initial value.

Method

LDDMM and geodesic shooting

Given a moving image I_0 and a fixed image I_1, LDDMM seeks a smooth time-dependent velocity field $v(t) \in V$, $t \in [0,1]$, that minimizes the registration energy

$$E(v) = \int_0^1 ||v(t)||^2_v \, dt + (1/\sigma^2) \, ||I_0 \circ \varphi_1^{-1} - I_1||^2_{L2},$$

where the diffeomorphism φ_t is obtained from the velocity field by the transport equation $\partial_t \varphi_t = v_t \circ \varphi_t$, $\varphi_0 = id$, integrated from $t=0$ to $t=1$. Under geodesic shooting, the whole geodesic is determined by the initial velocity $v_0 \in V$, obeying the EPDiff equation

$$\partial v/\partial t = -K[(Dv)^T \cdot Lv + D(Lv) \cdot v + Lv \cdot div(v)] \quad \text{[EPDiff]}$$

where D is the Jacobian matrix, div is the divergence operator, $\cdot$ is the pointwise product, K is the V -smoothing kernel and L its inverse. Optimization is therefore carried out in the initial velocity v_0 at $t=0$; its gradient is obtained by integrating the adjoint transport equations backwards in time to transfer the energy gradient from $t=1$ to $t=0$ [1].

POD

A snapshot matrix $S = [v_0^{(1)}, \ldots, v_0^{(N)}]$ is built from converged initial velocities obtained on a training set from a few iterations of the baseline method on the same experiment. Its leading left singular vectors yield a low-rank projection matrix U whose columns span the dominant r directions of the deformation manifold. Galerkin projection of EPDiff and of the adjoint transport equations onto the column space of U produces reduced dynamics, and the optimization is reprojected in the reduced coordinate $\alpha_0 = U^T v_0$ with $v_0 = U\alpha_0$. The per-iteration cost drops to matrix multiplications (the projections and reprojections) and r-dimensional ODEs [2].

NeurEPDiff

NeurEPDiff [3] trains a FNO to learn the EPDiff flow. Given an initial velocity v_0, the network predicts its time evolution $v(t)$. Training samples are generated synthetically by integrating EPDiff with a classical solver on sampled initial velocities. At inference time the optimization loop over v_0 is unchanged, only the EPDiff integrator is replaced by a forward pass of the FNO, with gradients propagated through the network by backpropagation.

Results

We evaluate our methods on two 3D T1 brain MRI benchmarks. NIREP provides 16 MRI with manual segmentations of 32 grey-matter regions; the first subject is taken as source and registered to the remaining target samples. OASIS Learn2Reg consists of MRI scans annotated with 35 brain structures; we work with the 19-image validation

subset as our test set guaranteeing there is no data leakage.

To separate the dimensionality reduction from the implementation language gap, each method is compared against a language-matched Flash [1] baseline. For each pair we perform up to 50 iterations and report mean and standard deviation of the Dice similarity coefficient (DSC), the minimum and maximum Jacobian determinant of the deformation, and the iteration runtime. Across all configurations, every registration yielded a strictly positive Jacobian determinant. Tables 1 and 2 collect the quantitative results. Figure 1 shows the qualitative results on a representative NIREP experiment.

Conclusions

In POD, the linear subspace built from training snapshots is not expressive enough to capture the relevant deformation modes of either population: a richer, maybe non-linear reduction is needed if the saving has to come without a clear accuracy loss.

We also suspect that NeurEPDiff is no longer carried out in the true geodesic space since the FNO is trained on samples produced by a numerical integrator and does not achieve nearly zero error while training. This more permissive surrogate admits a lower effective regularization, which translates into a wider Jacobian range yet delivers the highest registration accuracy.

Acknowledgements

This work has received partial funding from the projects PID2022-138703OB-I00, PID2023-148219OB-C22, the I3A (Gobierno de Aragón) Programa de Becas y Ayudas MCIN/AEI/10.13039/501100011033/FEDER and the inflammatory diseases network RD24/0007/0022 of the Instituto de Salud Carlos III.. Carlos Paesa-Lia is supported by a Ministerio de Ciencia, Innovación y Universidades FPU2024 predoctoral grant (2025-2029).

References

[1]. ZHANG, M. and FLETCHER, P.T. Finite-Dimensional Lie Algebras for Fast Diffeomorphic Image Registration. In: Information Processing in Medical Imaging: 24th International Conference, IPMI 2015, Isle of Skye, UK, June 28 - July 3, 2015, Proceedings. Cham: Springer, 2015, pp. 249-259. Available from doi: 10.1007/978-3-319-19992-4_19

[2]. WANG, J., XING, W., KIRBY, R.M., and ZHANG, M. Data-driven model order reduction for diffeomorphic image registration. In: Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2-7, 2019, Proceedings. Cham: Springer, 2019, pp. 694-705. Available from doi: 10.1007/978-3-030-20351-1_54

[3]. WU, N. and ZHANG, M. NeurEPDiff: Neural Operators to Predict Geodesics in Deformation Spaces. In: Information Processing in Medical Imaging: 28th International Conference, IPMI 2023, San Carlos de Bariloche, Argentina, June 18-23, 2023, Proceedings. Cham: Springer, 2023, pp. 588-600. Available from doi: 10.1007/978-3-031-34048-2_45

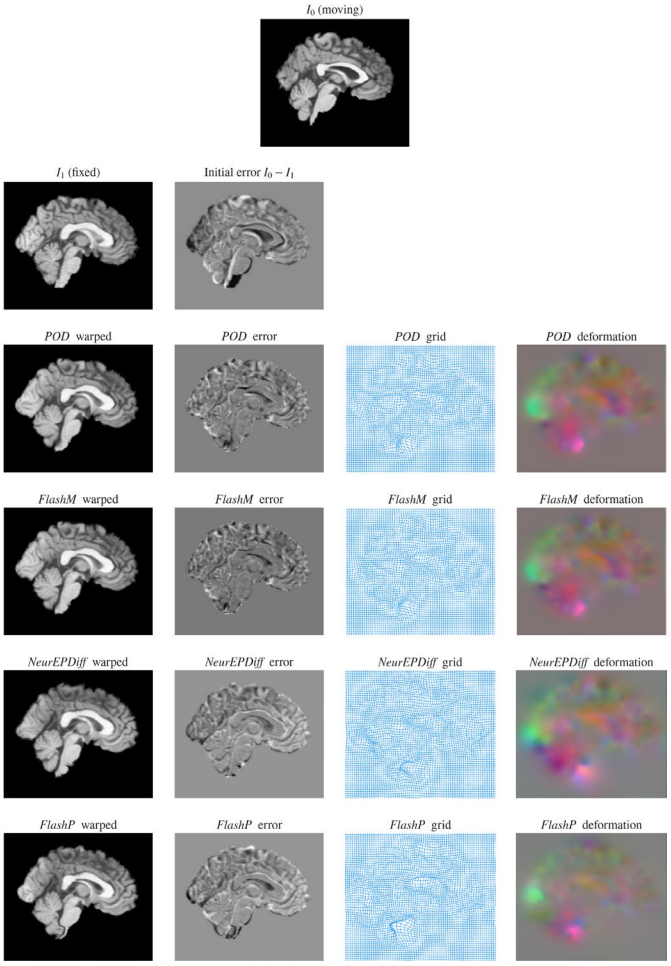

Figure 1. Registration results on the first NIREP pair, showing for each method's resulting central slices: the deformed image, the final error, the deformed grid and the 3D deformation seen as RGB. FlashM and FlashP are the Matlab and Python implementations of Flash [1], respectively.

Table 1. Comparison on NIREP (16 subjects, 32 labels).

Method	DSC (%) ↑	min |J| ↑	max |J| ↓	Time (s) ↓
POD (Matlab)	53.95±1.48	0.14±0.05	3.21±1.17	3.82±0.08
FlashM	55.56±1.57	0.14±0.04	3.13±0.28	4.96±0.01
NeurEPDiff (Python)	57.11±1.62	0.30±0.07	24.23±16.03	7.99±0.02
FlashP	54.27±1.90	0.29±0.12	7.61±3.43	11.85±0.70

Table 2. Comparison on OASIS (19 subjects, 35 labels).

Method	DSC (%) ↑	min |J| ↑	max |J| ↓	Time (s) ↓
POD(Matlab)	66.24±7.16	0.36±0.19	2.31±0.71	3.86±0.17
FlashM	69.02±5.86	0.24±0.09	3.75±2.50	4.84±0.01
NeurEPDiff (Python)	70.35±4.43	0.55±0.10	4.20±3.24	6.65±0.01
FlashP	68.67±6.76	0.42±0.15	4.00±2.05	8.95±0.77