LOCALLY CONNECTED PYRAMID REPRESENTATION FOR INSTANCE OPTIMIZATION IN 3D IMAGE REGISTRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose Locally Connected Pyramids for image registration (LCPReg), a light-weight neural displacement representation built specifically for pair-wise optimization. LCPReg explicitly optimizes only a coarse displacement field and carries it to image resolution through a cascade of learnable, locally connected upsampling layers. Each coarse site holds its own kernel, so the refinement can adapt to the local deformation, and each kernel is factorized into one-dimensional components, which keeps the parameter count to manageable numbers and encourages smooth transformations. We further introduce an approximate B-spline weight initialization scheme with warm start optimization that exactly reproduces a prior displacement estimate, allowing LCPReg to serve as a drop-in replacement in existing image registration pipelines. We integrate LCPReg into three frameworks (NODEO, convexAdam, and uniGradICON) and evaluate it on benchmark datasets (OASIS, CANDI and EMPIRE10). LCPReg consistently improves Dice scores over each framework's original representation with comparable regularity. It also substantially lowers optimization time for NODEO and uniGradICON.

1 INTRODUCTION

Many tasks in medical imaging depend on first bringing anatomical structures from different scans into correspondence. Deformable image registration (DIR) provides this alignment by estimating a dense spatial transformation that warps a moving image into the coordinate frame of a fixed one. The transformation must be flexible enough to capture large, non-linear anatomical differences, yet constrained enough to remain smooth and topology-preserving. Traditional frameworks define this as a variational problem, in which a measure of image dissimilarity is traded off against a regularization penalty on the deformation (Beg et al., 2005; Miller et al., 2006; Vercauteren et al., 2009). Learning-based registration takes advantage of recent advancements in deep learning by training on datasets of image pairs to predict deformation directly (Balakrishnan et al., 2019; Mok & Chung, 2020a; Tian et al., 2023). Although significantly faster at inference, these approaches are susceptible to domain shift (Hoffmann et al., 2021; Jena et al., 2024b), partially because of the limited availability of medical imaging training data.

Foundation models such as uniGradICON (Tian et al., 2024) and multiGradICON (Demir et al., 2024) aim to mitigate domain shift by training on large collections of heterogeneous datasets. Yet they too reach their best accuracy only after instance optimization (IO), a short round of pair-wise optimization that refines the network's prediction at test time (Heinrich & Hansen, 2022; Mok et al., 2023). Hybrid pipelines such as convexAdam (Siebert et al., 2024) follow the same pattern, pairing a fast coarse estimate from discrete optimization with iterative Adam-based refinement. Given the persistent prevalence of iterative optimization, numerous works have proposed advanced neural network-based representations of displacement fields (Wu et al., 2022; Wolterink et al., 2022; Sun et al., 2024; Hernandez, 2025; Han et al., 2023). This typically entails utilizing a small neural network to reconstruct the final deformation field which is independently optimized for a given image pair. However, the architectures that are employed for this purpose are usually Convolutional Neural Networks (CNNs), Multi-Layer Perceptrons (MLPs) or similar configurations that perform well in various deep learning applications, but are not specialized for large 3D volume inputs or cases where generalization is not an important requirement.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

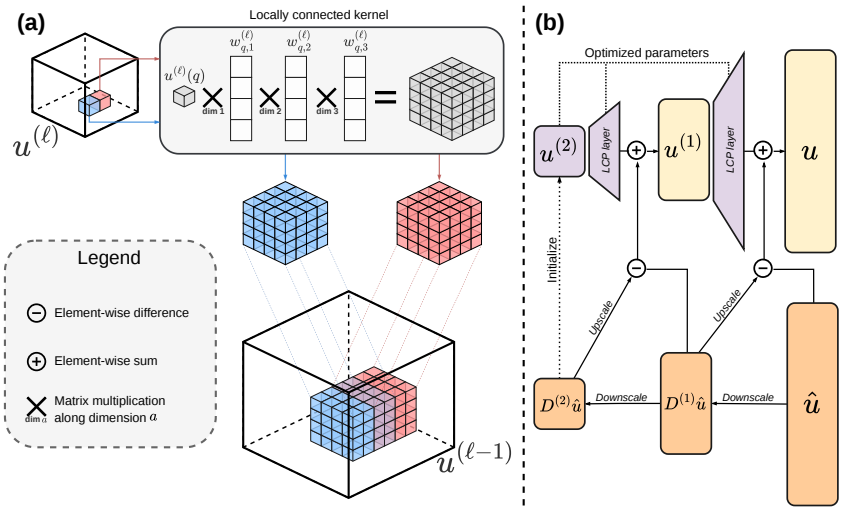

Figure 1: Diagram depicting a summary of our Locally Connected Pyramid (LCP). **(a)** Each LCP layer is composed of an upsampling operation with learnable weights. For every voxel in the coarse volume $u^{(\ell)}$ we define a set of learnable weights in the form of a locally connected kernel. Every voxel is multiplied by its corresponding one-dimensional weight vectors, and their products are assembled to form the upsampled volume. Overlapping multiplication results are combined by summation. **(b)** The pyramid is formed by an initial coarse volume and a set of LCP upsampling layers. In the depicted example, the learnable weights of the pyramid include both the coarsest vector field $u^{(L)}$ and the weights of the two LCP layers. Optionally, we define a warm-start initialization of our network using the Laplacian pyramid decomposition of an initial field $\hat{u}$.

To fill this gap, we propose the Locally Connected Pyramids (LCPs) for image registration (LCPReg), a displacement representation designed specifically for pair-wise optimization (Fig. 1). LCPReg explicitly optimizes only a coarse displacement field and carries it to image resolution through a cascade of learnable upsampling layers. Each layer is locally connected, such that every coarse site holds its own kernel rather than sharing one across the volume. We restrict each kernel to a product of one-dimensional factors to keep the total parameter count low and encourage smoother transformations. We integrate LCP into three frameworks (NODEO (Wu et al., 2022), convexAdam, and uniGradICON with IO) and evaluate it on several datasets. In every setting, LCP improves the Dice score over the framework's original representation while preserving transformation regularity and substantially lowering optimization times for both NODEO and uniGradICON. In summary, our contributions are: (1) LCP, a coarse-to-fine, locally connected displacement representation for pair-wise optimization that is spatially adaptive and inherently encourages smoother deformations; (2) an approximate B-spline initialization and a Laplacian-pyramid warm start that allow LCP to serve as a drop-in replacement in existing registration pipelines; and (3) an evaluation across three optimization frameworks and three datasets showing consistent accuracy gains over their original representations.

2 Related Work

Traditional image registration methods, such as Large Deformation Diffeomorphic Metric Mapping (LDDMM) by Beg et al. (2005); Miller et al. (2006) and Symmetric Normalization (SyN) by Avants et al. (2008), rely on computationally intensive iterative optimization algorithms. To overcome this computational hurdle, learning-based approaches utilize deep neural networks (DNNs) to predict deformation fields through a single forward pass. While initial works achieve this through supervised learning objectives (Cao et al., 2017; Krebs et al., 2017; Rohé et al., 2017; Sokooti et al., 2017), their reliance on labeled training datasets makes them impractical in most medical applications due to the lack of high quality annotations. The seminal work by Balakrishnan et al. (2018; 2019) first introduced an unsupervised training approach in the form of the VoxelMorph model, which incorporates

image similarity and regularization techniques from classical frameworks. Symmetric registration frameworks were also incorporated into deep learning by later works (Mok & Chung, 2020a; Kim et al., 2021; 2019; Tian et al., 2023; Zhao et al., 2019b; Greer et al., 2021). Additionally, coarse-to-fine refinement, another feature of classical registration algorithms, proved to be an especially effective strategy and has been the subject of recent research (Honkamaa & Marttinen, 2023; Zhao et al., 2019b; Joshi & Hong, 2022; De Vos et al., 2019; Mok & Chung, 2020b; Qiu et al., 2021; Zhang et al., 2021; Chen et al., 2022a; Guo et al., 2026). Despite their speed advantage, more recent research has shown that learning-based approaches fail to generalize outside of their initial training dataset (Hoffmann et al., 2021; Mok & Chung, 2022; Mok et al., 2023; Jena et al., 2024b; Jian et al., 2024).

Instance optimization (IO) (also referred to as instance adaptation) is an adaptation step utilized at inference time of learning-based methods. To overcome the domain shift issue, many works use instance specific fine-tuning of the trained deformation predictor on a single image pair (Heinrich & Hansen, 2022; Mok et al., 2023; Balakrishnan et al., 2019; Chen et al., 2022b). Most recently, foundation models in the form of uniGradICON (Tian et al., 2024) and its multimodal extension, multiGradICON (Demir et al., 2024), achieve state-of-the-art generalization with the help of IO. However, IO often introduces high computational costs, as it requires performing backpropagation through large cascades of neural networks used by these foundational models. Another line of work integrates test-time prediction refinement through various recurrent inference approaches (Zhao et al., 2019a; Zhang et al., 2021; Qiu et al., 2022; Sivan et al., 2026; Xu et al., 2021; Matinkia & Ray, 2026a;b). Other approaches fully abandon generalized learning over large datasets and return to iterative optimization by introducing different parameterizations of deformation fields using lightweight neural networks (Chen et al., 2026; Wolterink et al., 2022; Sun et al., 2024; Hu et al., 2024; Min et al., 2023; Chen et al., 2018). Finally, one of the best-performing image registration methods, convexAdam (Siebert et al., 2021; 2024; Heinrich et al., 2014), uses an initial displacement field to warm start direct optimization over a dense vector field, achieving state-of-the-art levels in both accuracy and generalization. Our work introduces a novel representation for the displacement field in the form of a lightweight locally connected neural network specifically designed to represent 3D spatial information. It provides implicit regularization while avoiding the prohibitive computational costs of larger neural network representations.

3 METHOD

3.1 PROBLEM SETTING

Given a fixed image I_F and a moving image I_M defined on a domain $\Omega \subset \mathbb{R}^3$, the goal of deformable image registration is to determine the transformation $\varphi = \mathrm{Id} + u$, with $u : \Omega \to \mathbb{R}^3$ being a dense displacement field, that brings the moving image I_M into alignment with I_F. In traditional pair-wise optimization and instance optimization frameworks u is determined by minimizing the registration objective for a single image pair, over the parameters θ of a chosen representation u_θ of the field,

$$\theta^\star = \arg\min_\theta \; \mathcal{L}_{sim}\big(I_F,\, I_M \circ \varphi_\theta\big) + \lambda_{reg}\, \mathcal{L}_{reg}(\varphi_\theta), \qquad \varphi_\theta = \mathrm{Id} + u_\theta, \tag{1}$$

where $\mathcal{L}_{sim}$ is a similarity measure between images, $\mathcal{L}_{reg}$ is a regularization applied to the deformation field that penalizes irregular deformations, and λ_{reg} is a scalar coefficient for balancing the two terms. The composition of θ is determined by the chosen representation of u, which consequently dictates the set of fields reachable by the optimization. Existing choices range between representing u directly as a vector field, in which case θ takes the value of the displacements at each voxel, or as a neural network, where the network's weights are optimized to emit a desirable deformation.

We represent the displacement field as Locally Connected Pyramid (LCP), which is a coarse-to-fine cascade of optimizable upsampling layers. Let $\Omega^{(0)}, \ldots, \Omega^{(L)}$ be regular grids over Ω, where $\Omega^{(0)}$ is the full resolution image grid and each $\Omega^{(\ell)}$ refines $\Omega^{(\ell+1)}$ by a factor s along every axis, so that $\Omega^{(\ell)}$ has spacing $h^{(\ell)} = h^{(0)} s^\ell$ and $|\Omega^{(0)}| = s^{3L} |\Omega^{(L)}|$. At the coarsest resolution, we place a vector field $u^{(L)} : \Omega^{(L)} \to \mathbb{R}^3$ whose values are included in θ, and thus, directly optimized. Each finer level is obtained from the level below by an upsampling operator $U_{\theta^{(\ell)}}$ as

$$u^{(\ell-1)} = U_{\theta^{(\ell)}}\big[u^{(\ell)}\big], \qquad \ell = 1, \ldots, L, \qquad u_\theta \equiv u^{(0)}. \tag{2}$$

The parameters of the representation are the coarsest field, together with the parameters of the subsequent upsampling operators:

$$\theta = \{\, u^{(L)},\ \theta^{(L)},\ \ldots\ \theta^{(1)} \,\}. \tag{3}$$

No field variables are directly optimized except at the coarsest level $\Omega^{(L)}$, the values on $\Omega^{(0)}, \ldots, \Omega^{(L-1)}$ are instead determined by $u^{(L)}$ and the weights of each upsample operator up to the current level: $\theta^{(L)},\ \ldots\ \theta^{(\ell+1)}$. The coarse spacing $h^{(L)}$ therefore sets the scale at which the deformation is described explicitly, while the $\theta^{(\ell)}$ govern how that description is carried to the image resolution. Building the field as a cascade of L small-factor refinements rather than a single s^L one lets this transport be adapted separately at each scale, with the operator at level ℓ acting at spacing $h^{(\ell)}$.

3.2 Locally Connected Upsampling Layers

The operators in equation 2 are linear maps from a coarse field to a finer one, and how they are defined determines both the expressiveness of the displacement representation and its computational complexity. A fully connected layer would let every voxel in the finer field draw on every voxel from the coarse one. But this is at a cost that scales with the product of the grid sizes and has no notion of locality. A convolutional (or de-convolutional) layer restricts each fine site to a neighborhood and shares one kernel across every position. This is ideal for learning universal features across a dataset, making convolutional neural networks a great choice for generalized learning-based methods. However, in an instance optimization setting, optimization is conducted on a single image pair, making generalization neither possible nor necessary. Additionally, by setting a single reconstruction rule for the entire volume, any spatial variation in how the field should be refined is disregarded.

By utilizing locally connected layers, we can preserve the locality of convolution while still keeping the space of optimizable parameters at a reasonable scale. Each coarse site q carries its own kernel $\kappa_q^{(\ell)}$, and the fine field is assembled as:

$$u^{(\ell-1)}(x) = \sum_{q \in \Omega^{(\ell)}} \kappa_q^{(\ell)}\left(\frac{x-q}{h^{(\ell)}}\right) u^{(\ell)}(q), \qquad x \in \Omega^{(\ell-1)}, \tag{4}$$

where each $\kappa_q^{(\ell)} : \mathbb{R}^3 \to \mathbb{R}$ is supported on $(-1,1)^3$, and $\theta^{(\ell)} = \{\kappa_q^{(\ell)} : q \in \Omega^{(\ell)}\}$. The argument of $\kappa_q^{(\ell)}$ is the offset from the coarse site q to the fine site x in units of the coarse spacing. We write $\xi = (x-q)/h^{(\ell)} \in \mathbb{R}^3$, with components ξ_1, ξ_2, ξ_3 along the three axes. Since $\kappa_q^{(\ell)}$ is evaluated only at fixed offsets, ξ takes a set of possible values that is dependent on the scaling factor s,

$$\Xi_s = \left\{ \pm \tfrac{2i+1}{2s} \ :\ i = 0, \ldots, s-1 \right\} \subset (-1,1), \tag{5}$$

so the offsets realized in equation 4 form $\Xi_s^3 \subset (-1,1)^3$. An unconstrained kernel, applied on a single channel of the displacement volume, is defined by $(2s)^3$ entries, and each upsampling operator $U_{\theta^{(\ell)}}$ would then require a total of $(2s)^3|\Omega^{(\ell)}|$ parameters. Although this is an improvement over fully connected networks, having every entry as a parameter is computationally too expensive at finer layers. For this reason, we restrict each kernel to a product of one-dimensional factors,

$$\kappa_q^{(\ell)}(\xi) = \prod_{a=1}^{3} w_{q,a}^{(\ell)}[\xi_a], \qquad \xi \in \Xi_s^3, \tag{6}$$

where each $w_{q,a}^{(\ell)}$ represents a vector of $2s$ entries indexed by Ξ_s. This reduces a kernel to only $3 \times 2s$ optimizable parameters while also constraining the upsampling operations to slightly smoother, more uniform, transformations.

In equation 4, a single scalar kernel $\kappa_q^{(\ell)}$ is applied to all three components of $u^{(\ell)}(q)$, so every displacement component is refined by the same spatially varying rule. This can be relaxed by assigning each component $c \in \{1,2,3\}$ its own kernel $\kappa_{q,c}^{(\ell)}$, with separable factors $w_{q,a,c}^{(\ell)}$, such that

$$u_c^{(\ell-1)}(x) = \sum_{q \in \Omega^{(\ell)}} \kappa_{q,c}^{(\ell)}\left(\frac{x-q}{h^{(\ell)}}\right) u_c^{(\ell)}(q), \qquad x \in \Omega^{(\ell-1)}. \tag{7}$$

Channel-wise kernels allow the refinement to differ between displacement components at the same location, for instance where the field varies sharply in one direction but smoothly in another. The parameter cost is a threefold increase to $18s\,|\Omega^{(\ell)}|$ parameters per level, exceeding a dense voxel field. However, because the parameter count is dominated by the finest levels, the choice can be made independently for each $U_{\theta^{(\ell)}}$. Channel-wise kernels at the coarse levels add comparatively few parameters, while shared kernels at the finest levels keep the total close to that of the shared configuration.

3.3 Approximate B-Spline Weight Initialization

Neural representations of the displacement field are commonly initialized so that their first prediction is the zero field, i.e., the identity transformation. A network with randomly initialized weights instead emits an irregular field with large, spatially incoherent displacements, from which the optimization converges slowly or diverges within the first iterations. The usual remedy is to initialize the weights of the final layer to zero, or to values close to it. This achieves the identity start, but it suppresses the gradient signal to the rest of the network: the gradient with respect to any earlier layer passes through the final layer's weights, so the first update reaches only the final layer, and the preceding layers receive gradients scaled by near-zero weights until these have grown. The early iterations therefore adapt a single layer on top of features that have not yet been shaped by the registration objective.

We instead obtain the identity start without zeroing any operator in the cascade. We set the coarse field $u^{(L)} = 0$ and initialize every factor to the linear tent

$$w^{(\ell)}_{q,a}[\xi_a] = 1 - |\xi_a|, \qquad \xi_a \in \Xi_s, \tag{8}$$

identically at every coarse site and every level. The product equation 6 is then the trilinear reconstruction kernel, and each operator performs trilinear interpolation of the field below it: every fine site draws on the eight surrounding coarse sites with non-negative weights that sum to one. At $s = 2$, the factors take the values $\frac{3}{4}$ and $\frac{1}{4}$ at the offsets $\pm\frac{1}{4}$ and $\pm\frac{3}{4}$. Composing L such levels does not reproduce a single trilinear interpolation from $\Omega^{(L)}$ to $\Omega^{(0)}$. Along each axis, every level applies the mask $(\frac{1}{4}, \frac{3}{4}, \frac{3}{4}, \frac{1}{4})$, which is the refinement rule of Chaikin's subdivision scheme, whose limit is the uniform quadratic B-spline. The initialized cascade therefore approximates a tensor-product quadratic B-spline with control points $u^{(L)}$. Because no operator is degenerate at initialization, the gradient of the objective reaches the coarse field from the first iteration. The kernels depart from equation 8 only where the objective favors it, and they can remain at the fixed interpolant wherever local adaptation does not help.

3.4 Warm Start Optimization

In many pipelines, instance optimization is not the first stage of alignment but a refinement of a displacement field $\hat{u}$ on $\Omega^{(0)}$ obtained in a prior step, for instance the prediction of a learning-based registration network or the output of a discrete optimization method. Let $D^{(\ell)}$ denote resampling onto $\Omega^{(\ell)}$. Seeding only the coarse field with $u^{(L)} = D^{(L)}\hat{u}$ does not make the cascade reproduce $\hat{u}$. Detail below the coarse spacing $h^{(L)}$ is lost in the restriction, and the initialized cascade reconstructs the remainder with the approximate quadratic B-spline described in the previous subsection, which is smoother than $\hat{u}$ itself. The optimization would therefore begin from a degraded version of the given field and spend its first iterations recovering detail that was already available.

We avoid this by seeding every level with Burt & Adelson (1983)'s Laplacian pyramid decomposition of $\hat{u}$, with $D^{(\ell)}$ and the initialized upsampling operators taking the roles of reduction and expansion. Let θ_{init} denote the parameters at initialization, with the upsampling layers set to equation 8 and the coarse field set to $u^{(L)}_{\text{init}} = D^{(L)}\hat{u}$. Before optimization begins, each operator is applied at its initial state to $\hat{u}$ resampled to its input level, and the discrepancy between its output and $\hat{u}$ resampled to its output level is recorded as a residual,

$$r^{(\ell)} = D^{(\ell)}\hat{u} - U_{\theta^{(\ell+1)}_{\text{init}}}\big[D^{(\ell+1)}\hat{u}\big], \qquad \ell = 0, \dots, L-1. \tag{9}$$

The residuals are computed once, are not included in θ, and are held fixed throughout the optimization, while the operators continue to evolve with θ. The cascade equation 2 is modified to add the

residuals at each level,

$$u^{(\ell)} = U_{\theta^{(\ell+1)}}\big[u^{(\ell+1)}\big] + r^{(\ell)}, \qquad \ell = 0, \ldots, L-1. \tag{10}$$

At $\theta = \theta_{\text{init}}$, substituting equation 9 into equation 10 and proceeding from $\ell = L-1$ down to $\ell = 0$ gives $u^{(\ell)} = D^{(\ell)}\hat{u}$ at every level. Since $D^{(0)}$ is the identity, $u_\theta = u^{(0)} = \hat{u}$. Before the first update, the representation therefore reproduces the given field exactly, both at image resolution and at every intermediate resolution, rather than a smoothed version of it.

3.5 Optimization Dynamics

We optimize the coarse field $z = u^{(L)}$ and the kernel weights as two parameter groups with learning rates η_u and η_w, since the former holds displacements while the latter are dimensionless interpolation coefficients. To see how the representation shapes the optimization, let $\mathcal{P}^{(\ell)} = U_{\theta^{(1)}} \circ \cdots \circ U_{\theta^{(\ell)}}$ carry level ℓ to image resolution, so that $u_\theta = \mathcal{P}^{(L)} z + \sum_{\ell<L} \mathcal{P}^{(\ell)} r^{(\ell)}$ is multilinear in z and the weights. By the chain rule (Appendix B), under gradient flow the full-resolution field evolves as $\dot{u}_\theta = -\mathbf{K}_\theta \nabla_u \mathcal{L}$, with

$$\mathbf{K}_\theta = \eta_u \, \mathcal{P}^{(L)} \mathcal{P}^{(L)\top} + \eta_w \sum_{\ell=1}^{L} \mathcal{P}^{(\ell-1)} \Big[\sum_{q \in \Omega^{(\ell)}} \Phi_q^{(\ell)} \otimes u^{(\ell)}(q)\, u^{(\ell)}(q)^\top \Big] \mathcal{P}^{(\ell-1)\top}, \tag{11}$$

where $\Phi_q^{(\ell)}$ is a positive semi-definite block supported on the $(2s)^3$ fine sites around q. $\mathbf{K}_\theta$ is the tangent kernel of the representation (Jacot et al., 2018), and its second term mirrors the adaptive dynamics of factorized linear models (Arora et al., 2018).

A dense field corresponds to $\mathbf{K} = \eta\, \mathbf{I}$, so every frequency of the gradient, including high-frequency noise from local similarity measures, moves the field directly. With frozen kernels ($\eta_w = 0$), only the first term remains: gradient descent on an approximate quadratic B-spline FFD (Section 3.3), a low-pass operator that cannot express detail below $h^{(L)}$. The second term is what learnable kernels add. It is local, scales with the displacement magnitude $\|u^{(\ell)}(q)\|^2$ (rather than its spatial variation), and acts at the finer spacing $h^{(\ell-1)}$, so detail is added where the field is already deforming. A convolutional upsampling layer, by contrast, shares each weight across all sites and contributes at most $6s$ global directions per level, with no local adaptation. At a cold start, $u^{(\ell)} = 0$ removes the second term, and adaptation emerges as the coarse field grows, giving an implicit coarse-to-fine schedule. With a warm start, $u^{(\ell)} = D^{(\ell)}\hat{u}$ activates it immediately, concentrating instance optimization where the prior model predicted motion. The ratio η_w/η_u balances the regularity of the first term against the local freedom of the second (Figure 4).

4 Experiments

4.1 Datasets and evaluation

We conduct experiments on two benchmark brain T1-weighted MRI datasets, OASIS (Marcus et al., 2007) and CANDI (Kennedy et al., 2012), and the lung CT scan dataset, EMPIRE10 (Murphy & et al., 2011). Following the protocols described by Wu et al. (2022), we set the first 5 images of OASIS as atlases and use the next 45 volumes as moving images (for a total of 225 pairs); for CANDI, we set the first image of each of the four subcategories as atlases and register the next 5 samples of each category to every atlas (a total of 80 pairs). For EMPIRE10, we use the predefined pairing of the full dataset for evaluation. OASIS and CANDI images are center-cropped to dimensions $160 \times 192 \times 144$, while EMPIRE10 samples are resized to those dimensions. Evaluation metrics are computed in the original image dimensions. When tuning hyperparameters, we used the remaining samples from OASIS, those that were not included in the test set, as a validation split.

For each task, we report the DICE score over the ground truth segmentation maps to measure registration accuracy. We compute the DICE score over every available structure in each task. To measure the regularity of the final deformation we report the percentage of negative Jacobian determinants of the final deformation field. We also measure the execution time on a single image pair using a single NVIDIA GeForce RTX 3090 GPU. Additional metrics can be found in Appendix A.

Table 1: Quantitative comparison of our results with baseline methods on the OASIS, CANDI and EMPIRE10 datasets. We evaluate the Dice score, percentage of negative Jacobians, and inference time in seconds. Results marked with † are taken directly from previous literature.

	OASIS		CANDI		EMPIRE10		
Method	DICE ↑	Neg. JD(%) ↓	DICE ↑	Neg. JD(%) ↓	DICE ↑	Neg. JD(%) ↓	Time ↓
initial ($\varphi = Id$)	0.5604 (7.2e-2)	0.0 (0.0)	0.5707 (4.5e-2)	0.0 (0.0)	0.8287 (9.2e-2)	0.0 (0.0)	-
VoxelMorph†	0.7655 (3.3e-2)	9.4e-2 (1.0e-3)	0.7459 (1.4e-2)	1.2e-1 (2.0e-2)	0.9681 (2.2e-2)	5.2e-2 (1.2e-2)	**0.16**
SYMNet†	0.7664 (3.2e-2)	8.0e-2 (1.0e-3)	**0.7489 (1.6e-2)**	9.1e-2 (1.0e-3)	0.9701 (1.2e-2)	5.4e-2 (9.0e-3)	0.28
uniGradICON	**0.7871 (3.1e-2)**	**2.2e-4 (4.0e-4)**	0.7313 (2.7e-2)	**4.9e-6 (4.1e-5)**	**0.9807 (6.3e-3)**	**2.6e-3 (8.9e-3)**	0.35
SyN	0.7031 (5.2e-2)	**0.0 (0.0)**	0.5879 (5.3e-2)	**0.0 (0.0)**	0.8583 (1.1e-1)	**0.0 (0.0)**	160.32
Log-Demons	0.7931 (4.0e-2)	9.1e-2 (7.0e-3)	0.7553 (3.6e-2)	1.0e-1 (1.0e-3)	0.9772 (1.9e-2)	5.7e-2 (1.1e-2)	43.29
FireANTs	0.7832 (4.6e-2)	6.2e-3 (8.1e-3)	0.7426 (3.4e-2)	**0.0 (0.0)**	0.9655 (4.7e-2)	9.4e-3 (3.1e-3)	**4.43**
PIRATE+†	**0.7952 (2.9e-2)**	4.9e-2 (1.0e-3)	**0.7633 (2.7e-2)**	6.6e-2 (1.0e-3)	**0.9811 (2.7e-2)**	4.7e-2 (7.0e-3)	49.47
NODEO	0.7864 (3.1e-2)	4.8e-2 (4.0e-2)	0.7589 (2.3e-2)	8.9e-4 (1.2e-3)	0.9742 (3.5e-3)	**1.1e-2 (1.5e-2)**	66.10
LCPReg-opt (Ours)	**0.8008 (3.2e-2)**	**8.6e-3 (1.4e-2)**	**0.7611 (2.3e-2)**	**6.0e-4 (5.1e-4)**	**0.9771 (3.0e-3)**	1.2e-2 (2.2e-2)	**23.13**
convexAdam-MIND	0.7496 (3.9e-2)	1.5e-2 (1.7e-2)	0.7338 (1.7e-2)	**2.9e-3 (7.8e-3)**	0.9247 (7.3e-2)	3.1e-2 (4.3e-2)	**3.04**
LCPReg-conv (Ours)	**0.7556 (3.8e-2)**	**1.2e-2 (1.5e-2)**	**0.7432 (2.8e-2)**	5.8e-3 (9.7e-3)	**0.9292 (7.2e-3)**	**2.2e-2 (3.5e-2)**	3.37
uniGradICON (w/ IO)	0.7942 (2.9e-2)	6.1e-3 (5.7e-2)	0.7481 (2.7e-2)	**3.5e-5 (7.3e-4)**	0.9820 (6.1e-3)	1.2e-2 (2.5e-2)	68.08
LCPReg-uGI (Ours)	**0.8053 (3.0e-2)**	**3.9e-3 (4.3e-3)**	**0.7708 (2.7e-2)**	3.5e-4 (5.7e-4)	**0.9826 (5.3e-3)**	**8.6e-3 (1.5e-2)**	**9.67**

4.2 Experimental setup

To effectively evaluate the contribution of our locally connected pyramid representation, we integrate it with three separate optimization frameworks. First, we replace the Neural ODE (Chen et al., 2018) in NODEO (Wu et al., 2022) with our network and use the same optimization objective. This includes a local normalized cross-correlation (LNCC) similarity term and separate regularization losses for deformation smoothness, magnitude, and negative Jacobian determinants; together with a smoothing filter applied to the final output. Second, we use the convexAdam (Siebert et al., 2021; 2024; Heinrich et al., 2014) framework, initialized using modality independent neighborhood descriptors (MIND) (Heinrich et al., 2012), where we insert our pyramid representation in place of the vector field in Adam optimization stage (this setting is labeled LCPReg-conv). Finally, we integrate our pyramid in place of the multi-resolution cascade of U-Net (Ronneberger et al., 2015) neural networks used by Tian et al. (2024) in the foundation image registration model called uniGradICON. Within this regime, instance optimization is regularized using an inverse consistency loss term; in our equivalent setting, dubbed LCPReg-uGI, we simultaneously optimize two separate pyramid networks for both the forward and backward transformation. In LCPReg-conv and LCPReg-uGI we use the initial predictions of the respective base models to warm start the optimization, as described in Section 3.4. For a more detailed description of each setting, together with the default hyperparameters, refer to Appendix C.

By default, we use three levels of the pyramid ($L = 2$) and only upsample by a factor of two ($s = 2$). We use channel-wise weights for every upsampling layer except for the finest one (at level $\Omega^{(0)}$). For LCPReg-opt and LCPReg-uGI we normalize the displacement vector to the unit cube and set the learning rates to: $\eta_u = 0.002$ and $\eta_w = 0.002$. In LCPReg-conv we keep the voxel spacing of the original method, and use the convexAdam learning rate of $\eta_u = 1$; and set $\eta_w = 0.0002$. For fair comparison, we adopt the same number of optimization iterations as the baseline methods, which is 300 for LCPReg-opt, 80 for LCPReg-conv and 50 for LCPReg-uGI. Finally, we depart from our standard data preprocessing in LCPReg-uGI and uniGradICON, and resize every image to $175 \times 175 \times 175$ instead and pad each dimension by 1 voxel when inputting to the pyramid.

4.3 Results

In Table 1 we directly compare the results of our three configurations to their respective baseline models (NODEO, convexAdam-MIND and uniGradICON with instance optimization).

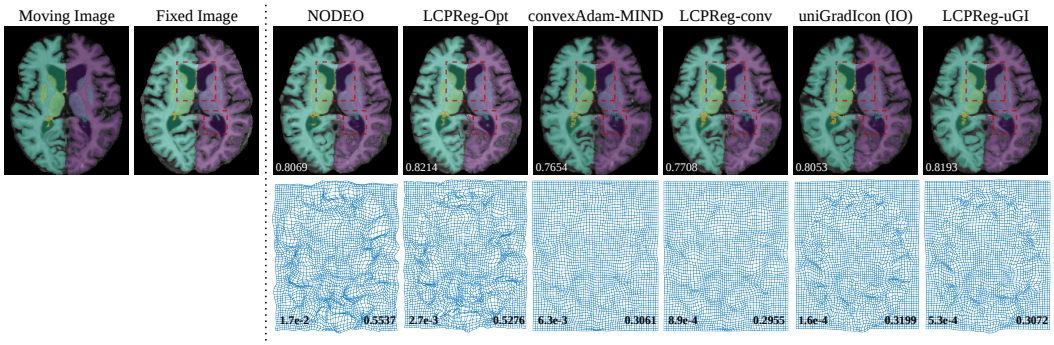

Figure 2: Visualization of the labeled warped images (top) and the corresponding deformation field (bottom) for baseline methods and our corresponding LCP integrations. The dice score for the given transformation is displayed in the bottom left of each warped image, the percentage of negative Jacobian determinants and the standard deviation of the log Jacobian are displayed on the bottom left and right, respectively, of each deformation field. Samples are from the OASIS dataset.

LCPReg-opt consistently outperforms the Neural ODE parametrization of NODEO with a 65% decrease in computation time. With LCPReg-conv we achieve overall better accuracy than the default Adam optimization; however, both approaches are limited by the MIND initialization. The greatest improvement can be seen with LCPReg-uGI, where the lightweight locally connected network substantially lowers computation time by roughly 86%, while also showing consistent improvements in registration accuracy. In addition to the three baseline methods, Table 1 contains results from several other modern and traditional approaches. We separate them into learning-based models: VoxelMorph (Balakrishnan et al., 2019), SYMNet (Mok & Chung, 2020a), and uniGradICON with no instance optimization; and pair-wise optimization models: SyN (Avants et al., 2008), Log-Demons (Vercauteren et al., 2009), FireANTs (Jena et al., 2026; 2024a), and PIRATE+ (Hu et al., 2024). Results for VoxelMorph, SYMNet and PIRATE+ were collected from their respective literature and are provided as referential information, as the comparisons to the three baselines are the most valuable in isolating our contribution. Our best setting, LCPReg-uGI, generally outperforms all of the other evaluated models by achieving the highest DICE score on every task with comparable regularity. Refer to Appendix A for results on additional metrics on all the evaluated datasets. Visual comparisons of each of our models with their corresponding baseline methods are shown in Figure 2.

Table 2: Ablation for isolating the different components of our method. We analyze settings without layers ($L = 0$), learnable residuals instead of upsample operators (Residual Pyramid) and naive warm starts.

	OASIS		
Configuration	DICE ↑	Neg. JD(%) ↓	Time ↓
LCPReg-uGI	0.8049 (3.0e-2)	3.4e-3 (3.7e-3)	9.67
$L = 0, \eta_u = 0.0002$	0.7922 (3.2e-2)	8.2e-3 (7.9e-3)	8.59
$L = 0, \eta_u = 0.0003$	0.7926 (3.1e-2)	1.6e-2 (1.4e-2)	
$L = 0, \eta_u = 0.0004$	0.7923 (3.1e-2)	2.9e-2 (2.3e-2)	
warm start only $u^{(0)}$	0.8047 (3.6e-2)	4.9e-3 (8.3e-3)	9.65
warm start only $u^{(L)}$	0.8007 (3.1e-2)	3.3e-3 (5.5e-3)	
Residual Pyramid			
$\eta_u = 0.0001$	0.7986 (3.1e-2)	2.7e-3 (2.2e-3)	13.45
$\eta_u = 0.0002$	0.7994 (3.1e-2)	8.6e-3 (1.4e-2)	
$\eta_u = 0.0003$	0.7990 (3.2e-2)	1.6e-2 (1.3e-2)	

4.4 Ablations

Table 2 isolates the components of LCPReg-uGI on the OASIS validation split. Removing the pyramid entirely ($L = 0$), which reduces the representation to a warm-started dense displacement field, lowers the Dice score to roughly 0.792 and increases folding as the learning rate grows. Replacing the locally connected upsampling operators with learnable per-level residuals (Residual Pyramid) recovers most of the accuracy, but at a 39% higher optimization time. The naive warm starts confirm the need for the full Laplacian pyramid initialization: seeding only the coarse field $u^{(L)}$ discards the fine detail of the uniGradICON prediction and slightly degrades Dice to 0.8007, while seeding only

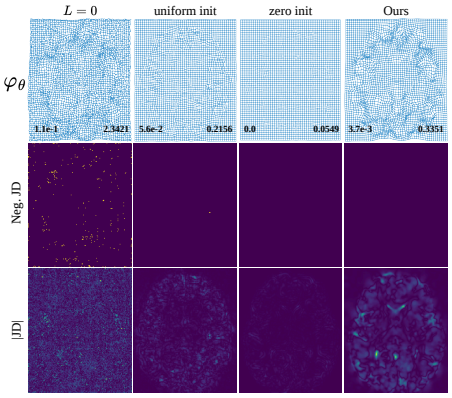

Figure 3: The influence of different weight initialization approaches on the result. We show the final deformation field (top), negative Jacobian determinants (middle) and the magnitude of the Jacobian (bottom).

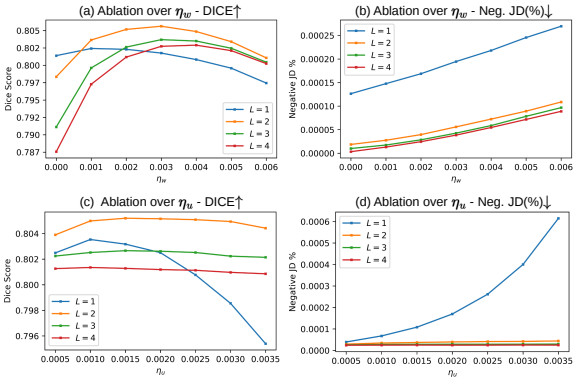

Figure 4: Ablation of different values for the two learning rates in our method for different numbers of LCP layers. We show the DICE score and negative Jacobian%. When ablating either η_w, or η_u, the other's value is set to 0.002.

the finest level $u^{(0)}$ preserves accuracy but yields marginally more irregular deformations (4.9e-3% vs. 3.4e-3% negative Jacobians). Setting $\eta_w = 0$ in Figure 4(a) freezes the upsampling kernels at their B-spline initialization, reducing LCP to an approximate quadratic B-spline FFD; this consistently lowers the Dice score, and the drop grows with the number of layers, confirming that learnable kernels are needed to recover detail below the coarse spacing $h^{(L)}$.

We analyze the effect of different initialization methods for the upsampling operator weights in Figure 3. In it we use our LCPReg-uGI setting to test uniform and zero initialization, which do not converge to any meaningful displacement within the 50 iteration steps. On the other hand, not using any LCP layers, and optimizing directly on the displacement vector field, causes highly irregular deformations compared to our default setting. This can be further verified in Figure 4 where we ablate over the number of LCP layers for different learning rates η_u and η_w. This ablation shows a clear advantage when using more layers in our pyramid, empirically demonstrating the utility of our method. However, the additional parameters make for more gradual convergence, which manifests in a trade-off of slightly lower Dice score with the increase of layers past $L = 2$. The figure also shows the relationship between the two learning rates and their effect on optimization convergence. The irregularity of the output deformation directly scales with the increase of step size, while for registration accuracy we found the optimal values for both η_u and η_w are between 0.002 and 0.003.

5 Conclusion

We introduced the Locally Connected Pyramids for image registration (LCPReg), a displacement representation for pair-wise and instance optimization in 3D deformable image registration. LCP explicitly optimizes only a coarse displacement field and refines it to image resolution through a cascade of separable, locally connected upsampling layers. This allows the representation to adapt to local deformation while keeping the total parameter count similar to that of a dense voxel field. Combined with an approximate B-spline initialization and a Laplacian-pyramid warm start setup, LCPReg serves as a drop-in replacement for the displacement representations of existing pipelines. Integrated into NODEO, convexAdam, and uniGradICON, LCP improved Dice scores on OASIS, CANDI, and EMPIRE10 with comparable deformation regularity to the baseline methods. Notably, it substantially reduced optimization time for NODEO and uniGradICON. In its uniGradICON integration, LCP achieved the highest Dice score on every task among all evaluated methods. These results suggest that representations designed specifically for per-pair optimization, rather than borrowed from generalization-oriented architectures, are a promising direction. Extending LCP to velocity field representation or multimodal settings is a natural direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools for formulating mathematical claims, assisting in the writing of proofs and implementing methods. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments and interpret results; and the tasks to generate synthetic data sets, clean and reformat datasets, support qualitative and thematic data analysis and assist with translation are not applicable to this work. Additionally, we used generative AI tools to create or edit software code and draft parts of the research paper. We have reviewed all AI-assisted work. Parts of the paper drafted by LLMs were manually edited before inclusion in the final manuscript, references and citations were not drafted by AI and only written manually. LLM-generated code was manually verified via unit testing. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed explanations of the three optimization settings in Appendix C; this includes specifications about the loss objectives of the original baseline methods that we adapted, together with the default hyperparameter configurations that we used in our comparisons. A common practice in image registration works is to randomly pair images within a dataset to construct training, validation, and test splits. However, we have found that results can vary greatly when random splits are seeded differently, which makes comparisons between different publications difficult. For this reason, we explicitly define the image pairings for our test sets in Section 4.1. Finally, we provide our full code in the supplementary material and intend to provide a link to a public repository in the camera-ready version.

REFERENCES

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International conference on machine learning*, pp. 244–253. PMLR, 2018.

Brian B Avants, Charles L Epstein, Murray Grossman, and James C Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis*, 12(1):26–41, 2008.

Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. An unsupervised learning model for deformable medical image registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9252–9260, 2018.

Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. Voxelmorph: a learning framework for deformable medical image registration. *IEEE transactions on medical imaging*, 38(8):1788–1800, 2019.

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

P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. doi: 10.1109/TCOM.1983.1095851.

Xiaohuan Cao, Jianhua Yang, Jun Zhang, Dong Nie, Minjeong Kim, Qian Wang, and Dinggang Shen. Deformable image registration based on similarity-steered cnn regression. In *International conference on medical image computing and computer-assisted intervention*, pp. 300–308. Springer, 2017.

Junyu Chen, Eric C Frey, and Yong Du. Unsupervised learning of diffeomorphic image registration via transmorph. In *International Workshop on Biomedical Image Registration*, pp. 96–102. Springer, 2022a.

Junyu Chen, Eric C Frey, Yufan He, William P Segars, Ye Li, and Yong Du. Transmorph: Transformer for unsupervised medical image registration. *Medical image analysis*, 82:102615, 2022b.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Xiang Chen, Renjiu Hu, Jiacheng Wang, Min Liu, Yaonan Wang, Jiazheng Wang, Rongguang Wang, Gaolei Li, and Hang Zhang. Neural optimization for image registration via joint modeling of global affine and local deformation transformations. *IEEE Transactions on Medical Imaging*, 2026.

Bob D De Vos, Floris F Berendsen, Max A Viergever, Hessam Sokooti, Marius Staring, and Ivana Išgum. A deep learning framework for unsupervised affine and deformable image registration. *Medical image analysis*, 52:128–143, 2019.

Başar Demir, Lin Tian, Hastings Greer, Roland Kwitt, François-Xavier Vialard, Raúl San José Estépar, Sylvain Bouix, Richard Rushmore, Ebrahim Ebrahim, and Marc Niethammer. Multigradicon: A foundation model for multimodal medical image registration. In *International Workshop on Biomedical Image Registration*, pp. 3–18. Springer, 2024.

Hastings Greer, Roland Kwitt, François-Xavier Vialard, and Marc Niethammer. Icon: Learning regular maps through inverse consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3396–3405, 2021.

Mengjie Guo, Xinxing Cheng, Wenqi Lu, Qingjie Meng, Guanyu Yang, Yang Chen, Ziyun Ding, Alejandro F Frangi, and Jinming Duan. Foundation-guided representation alignment for multimodal medical image registration. In *European Conference on Computer Vision*, pp. 518–535. Springer, 2026.

Kun Han, Shanlin Sun, Xiangyi Yan, Chenyu You, Hao Tang, Junayed Naushad, Haoyu Ma, Deying Kong, and Xiaohui Xie. Diffeomorphic image registration with neural velocity field. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1869–1879, 2023.

Mattias P Heinrich and Lasse Hansen. Voxelmorph++ going beyond the cranial vault with keypoint supervision and multi-channel instance optimisation. In *International workshop on biomedical image registration*, pp. 85–95. Springer, 2022.

Mattias P Heinrich, Mark Jenkinson, Manav Bhushan, Tahreema Matin, Fergus V Gleeson, Michael Brady, and Julia A Schnabel. Mind: Modality independent neighbourhood descriptor for multimodal deformable registration. *Medical image analysis*, 16(7):1423–1435, 2012.

Mattias P Heinrich, Bartlomiej W Papież, Julia A Schnabel, and Heinz Handels. Non-parametric discrete registration with convex optimisation. In *Biomedical Image Registration: 6th International Workshop, WBIR 2014, London, UK, July 7-8, 2014. Proceedings 6*, pp. 51–61. Springer, 2014.

Monica Hernandez. Pde-lddmm meets nodes: Introducing neural ordinary differential equation solvers in pde-constrained large deformation diffeomorphic metric mapping. *Journal of Computational Science*, 85:102507, 2025.

Malte Hoffmann, Benjamin Billot, Douglas N Greve, Juan Eugenio Iglesias, Bruce Fischl, and Adrian V Dalca. Synthmorph: learning contrast-invariant registration without acquired images. *IEEE transactions on medical imaging*, 41(3):543–558, 2021.

Joel Honkamaa and Pekka Marttinen. Sitreg: Multi-resolution architecture for symmetric, inverse consistent, and topology preserving image registration. *arXiv preprint arXiv:2303.10211*, 2023.

Junhao Hu, Weijie Gan, Zhixin Sun, Hongyu An, and Ulugbek Kamilov. A plug-and-play image registration network. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=DGez4B2a6Y`.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Rohit Jena, Pratik Chaudhari, and James C Gee. Fireants: Adaptive riemannian optimization for multi-scale diffeomorphic registration. *Nature Communications*, 2024a.

Rohit Jena, Deeksha Sethi, Pratik Chaudhari, and James C Gee. Deep learning in medical image registration: Magic or mirage? *Advances in Neural Information Processing Systems*, 37:108331–108353, 2024b.

Rohit Jena, Vedant Zope, Pratik Chaudhari, and James C Gee. A scalable distributed framework for multimodal GigaVoxel image registration. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8dLexnao2h.

Bailiang Jian, Jiazhen Pan, Morteza Ghahremani, Daniel Rueckert, Christian Wachinger, and Benedikt Wiestler. Mamba? catch the hype or rethink what really helps for image registration. In *International workshop on biomedical image registration*, pp. 86–97. Springer, 2024.

Ankita Joshi and Yi Hong. Diffeomorphic image registration using lipschitz continuous residual networks. In *International Conference on Medical Imaging with Deep Learning*, pp. 605–617. PMLR, 2022.

David N Kennedy, Christian Haselgrove, Steven M Hodge, Pallavi S Rane, Nikos Makris, and Jean A Frazier. Candishare: a resource for pediatric neuroimaging data. *Neuroinformatics*, 10: 319–322, 2012.

Boah Kim, Jieun Kim, June-Goo Lee, Dong Hwan Kim, Seong Ho Park, and Jong Chul Ye. Unsupervised deformable image registration using cycle-consistent cnn. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 166–174. Springer, 2019.

Boah Kim, Dong Hwan Kim, Seong Ho Park, Jieun Kim, June-Goo Lee, and Jong Chul Ye. Cyclemorph: cycle consistent unsupervised deformable image registration. *Medical image analysis*, 71:102036, 2021.

Julian Krebs, Tommaso Mansi, Hervé Delingette, Li Zhang, Florin C Ghesu, Shun Miao, Andreas K Maier, Nicholas Ayache, Rui Liao, and Ali Kamen. Robust non-rigid registration through agent-based action learning. In *International conference on medical image computing and computer-assisted intervention*, pp. 344–352. Springer, 2017.

Daniel S Marcus, Tracy H Wang, Jamie Parker, John G Csernansky, John C Morris, and Randy L Buckner. Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults. *Journal of cognitive neuroscience*, 19(9):1498–1507, 2007.

Mohammadjavad Matinkia and Nilanjan Ray. Learning diffeomorphism for medical image registration with time-embedded architectures using semigroup regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28775–28785, 2026a.

Mohammadjavad Matinkia and Nilanjan Ray. Two-parameter flow map learning for continuous-time diffeomorphic image registration. In *Computer Vision–ECCV 2026: 19th European Conference, Malmö, Sweden, September 8–12, 2026, Proceedings, Part XLI*, pp. 207. Springer Nature, 2026b.

Michael I Miller, Alain Trouvé, and Laurent Younes. Geodesic shooting for computational anatomy. *Journal of mathematical imaging and vision*, 24:209–228, 2006.

Zhe Min, Zachary MC Baum, Shaheer U Saeed, Mark Emberton, Dean C Barratt, Zeike A Taylor, and Yipeng Hu. Non-rigid medical image registration using physics-informed neural networks. In *International conference on information processing in medical imaging*, pp. 601–613. Springer, 2023.

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

Tony CW Mok and Albert Chung. Affine medical image registration with coarse-to-fine vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20835–20844, 2022.

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

Tony CW Mok, Zi Li, Yingda Xia, Jiawen Yao, Ling Zhang, Jingren Zhou, and Le Lu. Deformable medical image registration under distribution shifts with neural instance optimization. In *International Workshop on Machine Learning in Medical Imaging*, pp. 126–136. Springer, 2023.

Keelin Murphy and et al. Evaluation of registration methods on thoracic ct: The empire10 challenge. *IEEE Transactions on Medical Imaging*, 30(11):1901–1920, 2011. doi: 10.1109/TMI.2011.2158349.

Huaqi Qiu, Chen Qin, Andreas Schuh, Kerstin Hammernik, and Daniel Rueckert. Learning diffeomorphic and modality-invariant registration using b-splines. In *Medical imaging with deep learning*, 2021.

Huaqi Qiu, Kerstin Hammernik, Chen Qin, Chen Chen, and Daniel Rueckert. Embedding gradient-based optimization in image registration networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 56–65. Springer, 2022.

Marc-Michel Rohé, Manasi Datar, Tobias Heimann, Maxime Sermesant, and Xavier 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.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Hanna Siebert, Lasse Hansen, and Mattias P Heinrich. Fast 3d registration with accurate optimisation and little learning for learn2reg 2021. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 174–179. Springer, 2021.

Hanna Siebert, Christoph Großbröhmer, Lasse Hansen, and Mattias P Heinrich. Convexadam: Self-configuring dual-optimisation-based 3d multitask medical image registration. *IEEE Transactions on Medical Imaging*, 2024.

Vignesh Sivan, Teodora Vujovic, Raj Kumar Ranabhat, Alexander Wong, Stewart Mclachlin, and Michael Hardisty. Recurrence with correlation network for medical image registration. *Applied Sciences*, 16(4):2084, 2026.

Hessam Sokooti, Bob De Vos, Floris Berendsen, Boudewijn PF Lelieveldt, Ivana Išgum, and Marius Staring. Nonrigid image registration using multi-scale 3d convolutional neural networks. In *International conference on medical image computing and computer-assisted intervention*, pp. 232–239. Springer, 2017.

Shanlin Sun, Kun Han, Chenyu You, Hao Tang, Deying Kong, Junayed Naushad, Xiangyi Yan, Haoyu Ma, Pooya Khosravi, James S Duncan, et al. Medical image registration via neural fields. *Medical Image Analysis*, 97:103249, 2024.

Lin Tian, Hastings Greer, François-Xavier Vialard, Roland Kwitt, Raúl San José Estépar, Richard Jarrett Rushmore, Nikolaos Makris, Sylvain Bouix, and Marc Niethammer. Gradicon: Approximate diffeomorphisms via gradient inverse consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18084–18094, 2023.

Lin Tian, Hastings Greer, Roland Kwitt, Francois-Xavier Vialard, Raúl San José Estépar, Sylvain Bouix, Richard Rushmore, and Marc Niethammer. unigradicon: A foundation model for medical image registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 749–760. Springer, 2024.

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

Jelmer M Wolterink, Jesse C Zwienenberg, and Christoph Brune. Implicit neural representations for deformable image registration. In *International Conference on medical imaging with deep learning*, pp. 1349–1359. PMLR, 2022.

Yifan Wu, Tom Z Jiahao, Jiancong Wang, Paul A Yushkevich, M Ani Hsieh, and James C Gee. Nodeo: A neural ordinary differential equation based optimization framework for deformable image registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20804–20813, 2022.

Junshen Xu, Eric Z Chen, Xiao Chen, Terrence Chen, and Shanhui Sun. Multi-scale neural odes for 3d medical image registration. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part IV 24*, pp. 213–223. Springer, 2021.

Liutong Zhang, Lei Zhou, Ruiyang Li, Xianyu Wang, Boxuan Han, and Hongen Liao. Cascaded feature warping network for unsupervised medical image registration. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 913–916. IEEE, 2021.

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

Shengyu Zhao, Tingfung Lau, Ji Luo, Eric I-Chao Chang, and Yan Xu. Unsupervised 3d end-to-end medical image registration with volume tweening network. *IEEE journal of biomedical and health informatics*, 24(5):1394–1404, 2019b.

APPENDIX

A ADDITIONAL EVALUATION METRICS

In addition to the DICE and negative Jacobian percentages reported in the main paper, we evaluate our method with HD95(mm) for segmentation accuracy, the total sum of negative Jacobian determinants, and the standard deviation of the log Jacobians. Results on these metrics for OASIS and CANDI are shown in Table 3 and for EMPIRE10 in Table 4.

Table 3: Quantitative comparison of HD95, sum of negative Jacobians and standard deviation of log Jacobians for OASIS and CANDI.

	OASIS			CANDI		
Method	HD95(mm) ↓	Sum JD ↓	SDLogJD ↓	HD95(mm) ↓	Sum JD ↓	SDLogJD ↓
NODEO	2.350 (0.552)	133.34 (40.2)	0.705 (0.145)	2.607 (0.393)	10.83 (10.19)	0.392 (0.036)
LCPReg-opt (Ours)	**2.257 (0.560)**	**27.17 (29.21)**	**0.606 (0.105)**	**2.597 (0.391)**	**3.557 (3.429)**	**0.337 (0.020)**
convexAdam-MIND	2.653 (0.672)	**68.51 (98.06)**	**0.364 (0.090)**	2.757 (0.439)	**10.33 (30.21)**	**0.206 (0.065)**
LCPReg-conv (Ours)	**2.584 (0.646)**	140.18 (164.10)	0.480 (0.111)	**2.708 (0.435)**	19.83 (37.81)	0.261 (0.071)
uniGradICON (w/ IO)	2.190 (0.479)	**3.99 (8.25)**	0.384 (0.036)	2.637 (0.426)	**0.08 (0.37)**	**0.201 (0.026)**
LCPReg-uGI (Ours)	**2.112 (0.482)**	9.31 (24.52)	**0.354 (0.039)**	**2.452 (0.432)**	0.73 (1.32)	0.215 (0.025)

Table 4: Quantitative comparison of HD95, sum of negative Jacobians and standard deviation of log Jacobians for EMPIRE10

	EMPIRE10		
Method	HD95(mm) ↓	Sum JD ↓	SDLogJD ↓
NODEO	5.119 (14.01)	661.4 (823.3)	**0.452 (0.172)**
LCPReg-opt (Ours)	**3.504 (11.13)**	**445.8 (615.4)**	0.455 (0.210)
convexAdam-MIND	**15.71 (15.93)**	7119 (6445)	0.991 (0.681)
LCPReg-conv (Ours)	**15.70 (15.91)**	**5911 (5005)**	**0.954 (0.637)**
uniGradICON (w/ IO)	1.535 (0.713)	7978 (1271)	0.537 (0.721)
LCPReg-uGI (Ours)	**1.371 (0.536)**	**4737 (9304)**	**0.513 (0.345)**

B OPTIMIZATION DYNAMICS OF LCP

This section derives the tangent kernel $\mathbf{K}_\theta$ in equation 11 and states the consequences summarized in Section 3.5 in more detail.

Notation. We treat a vector field on $\Omega^{(\ell)}$ as a vector in $\mathbb{R}^{|\Omega^{(\ell)}|} \otimes \mathbb{R}^3$, with the first factor indexing spatial sites and the second indexing displacement components. Let $z = u^{(L)}$ denote the coarse field and let

$$\mathcal{P}^{(\ell)} = U_{\theta^{(1)}} \circ \cdots \circ U_{\theta^{(\ell)}} : \Omega^{(\ell)} \to \Omega^{(0)}, \qquad \mathcal{P}^{(0)} = \mathrm{Id}, \tag{12}$$

be the linear map that carries level ℓ to image resolution under the current weights. Unrolling the warm-started cascade equation 10 gives

$$u_\theta = \mathcal{P}^{(L)} z + \sum_{\ell=0}^{L-1} \mathcal{P}^{(\ell)} r^{(\ell)}, \tag{13}$$

with $r^{(\ell)} = 0$ at a cold start. For fixed weights, u_θ is affine in z; jointly, it is multilinear in z and the weights. We write $g = \nabla_u \mathcal{L}(u_\theta)$ for the gradient of the objective with respect to the full-resolution field.

Local kernel derivatives. For a coarse site $q \in \Omega^{(\ell)}$, an axis a, and an offset $k \in \Xi_s$, define the scalar field on $\Omega^{(\ell-1)}$

$$\phi^{(\ell)}_{q,a,k}(x) = \mathbf{1}[\xi_a = k] \prod_{b \neq a} w^{(\ell)}_{q,b}[\xi_b], \qquad \xi = \frac{x - q}{h^{(\ell)}}, \tag{14}$$

which is zero outside the $(2s)^3$ fine sites surrounding q. By equation 4 and equation 6, $u^{(\ell)}(q)$ enters $U_{\theta^{(\ell)}}[u^{(\ell)}]$ only through the term $\kappa^{(\ell)}_q(\xi)\, u^{(\ell)}(q)$, and $\kappa^{(\ell)}_q$ is linear in each individual entry $w^{(\ell)}_{q,a}[k]$. Therefore

$$\frac{\partial\, U_{\theta^{(\ell)}}[u^{(\ell)}]}{\partial w^{(\ell)}_{q,a}[k]} = \phi^{(\ell)}_{q,a,k} \otimes u^{(\ell)}(q). \tag{15}$$

Proposition 1 (Tangent kernel of LCP) *Under the gradient flow $\dot z = -\eta_u \nabla_z \mathcal{L}$ and $\dot\theta^{(\ell)} = -\eta_w \nabla_{\theta^{(\ell)}} \mathcal{L}$ for $\ell = 1, \dots, L$, the full-resolution field evolves as $\dot u_\theta = -\mathbf{K}_\theta\, g$, with*

$$\mathbf{K}_\theta = \eta_u\, \mathcal{P}^{(L)} \mathcal{P}^{(L)\top} + \eta_w \sum_{\ell=1}^{L} \mathcal{P}^{(\ell-1)} M^{(\ell)} \mathcal{P}^{(\ell-1)\top}, \qquad M^{(\ell)} = \sum_{q \in \Omega^{(\ell)}} \Phi^{(\ell)}_q \otimes u^{(\ell)}(q)\, u^{(\ell)}(q)^\top, \tag{16}$$

where $\Phi^{(\ell)}_q = \sum_{a=1}^{3} \sum_{k \in \Xi_s} \phi^{(\ell)}_{q,a,k}\, \phi^{(\ell)\top}_{q,a,k}$ is positive semi-definite and supported on the neighborhood of q.

Proof. Let $J^{(\ell)} = \partial u_\theta / \partial \theta^{(\ell)}$. By the chain rule and equation 13, $\dot u_\theta = \mathcal{P}^{(L)} \dot z + \sum_\ell J^{(\ell)} \dot\theta^{(\ell)}$. Since $\nabla_z \mathcal{L} = \mathcal{P}^{(L)\top} g$ and $\nabla_{\theta^{(\ell)}} \mathcal{L} = J^{(\ell)\top} g$, substituting the flow gives $\mathbf{K}_\theta = \eta_u \mathcal{P}^{(L)} \mathcal{P}^{(L)\top} + \eta_w \sum_\ell J^{(\ell)} J^{(\ell)\top}$. It remains to compute $J^{(\ell)}$. From equation 10, $u_\theta = \mathcal{P}^{(\ell-1)}\big(U_{\theta^{(\ell)}}[u^{(\ell)}] + r^{(\ell-1)}\big) + \sum_{m<\ell-1} \mathcal{P}^{(m)} r^{(m)}$, where $\mathcal{P}^{(\ell-1)}$ depends only on $\theta^{(1)}, \dots, \theta^{(\ell-1)}$, $u^{(\ell)}$ depends only on z and $\theta^{(\ell+1)}, \dots, \theta^{(L)}$, and the residuals are constant. Hence $\partial u_\theta / \partial w^{(\ell)}_{q,a}[k] = \mathcal{P}^{(\ell-1)}\big(\phi^{(\ell)}_{q,a,k} \otimes u^{(\ell)}(q)\big)$ by equation 15. Summing the outer products of these columns over (q, a, k) and using $(A \otimes B)(C \otimes D)^\top = AC^\top \otimes BD^\top$ gives $J^{(\ell)} J^{(\ell)\top} = \mathcal{P}^{(\ell-1)} M^{(\ell)} \mathcal{P}^{(\ell-1)\top}$. □

For a discrete step of size η, the same expression holds up to $O(\eta^2)$ terms.

Channel-wise kernels. With the channel-wise kernels of equation 7, the derivative with respect to $w^{(\ell)}_{q,a,c}[k]$ is $\phi^{(\ell)}_{q,a,c,k} \otimes e_c\, u^{(\ell)}_c(q)$, where e_c is the c-th unit vector and $\phi^{(\ell)}_{q,a,c,k}$ is defined as in equation 14 with the factors of component c. The local term becomes

$$M^{(\ell)} = \sum_{q \in \Omega^{(\ell)}} \sum_{c=1}^{3} \Phi^{(\ell)}_{q,c} \otimes u^{(\ell)}_c(q)^2\, e_c e_c^\top. \tag{17}$$

Shared kernels therefore contribute the rank-one component matrix $u^{(\ell)}(q)\, u^{(\ell)}(q)^\top$, which accelerates motion along the current displacement direction at q, whereas channel-wise kernels contribute a diagonal matrix that scales each component independently.

Frozen kernels. For $\eta_w = 0$, $\mathbf{K}_\theta = \eta_u \mathcal{P}^{(L)} \mathcal{P}^{(L)\top}$, whose rank is at most $3|\Omega^{(L)}|$. Any component of g orthogonal to the range of $\mathcal{P}^{(L)}$ is discarded. At the initialization equation 8, $\mathcal{P}^{(L)}$ is the L-fold Chaikin cascade, so the dynamics are those of an approximate quadratic B-spline free-form deformation with control points on $\Omega^{(L)}$. With a warm start, the residuals in equation 13 preserve the detail of $\hat u$ below $h^{(L)}$, but the optimization cannot modify it.

Convolutional upsampling. If the upsampling at level ℓ instead uses a single separable kernel shared across all sites, each weight $w^{(\ell)}_a[k]$ enters every site, and its column of $J^{(\ell)}$ is $\mathcal{P}^{(\ell-1)} \sum_q \phi^{(\ell)}_{q,a,k} \otimes u^{(\ell)}(q)$. The second term of equation 16 then has rank at most $6s$ per level ($18s$ for channel-wise kernels), independently of the grid size. It adds a small number of global directions to the dynamics, but none that act locally.

Cold and warm starts. At a cold start, $z = 0$ implies $u^{(\ell)} = 0$ at every level, so $M^{(\ell)} = 0$ and the initial dynamics are those of the frozen-kernel case. Because each $u^{(\ell)}$ is linear in z for fixed weights, $M^{(\ell)}$ is quadratic in z, and the adaptive term grows quadratically with the coarse displacement during the early iterations. With the warm start of Section 3.4, $u^{(\ell)} = D^{(\ell)}\hat{u}$ at initialization, so

$$M^{(\ell)}\big|_{\theta_{\text{init}}} = \sum_{q \in \Omega^{(\ell)}} \Phi_q^{(\ell)} \otimes D^{(\ell)}\hat{u}(q)\, D^{(\ell)}\hat{u}(q)^\top, \tag{18}$$

and the adaptive term is largest where the prior estimate $\hat{u}$ has large displacements. We note that this adaptation depends on the magnitude of the displacement rather than on its spatial variation. A large, locally uniform displacement also increases the effective step size; after affine pre-alignment, however, the remaining displacement is predominantly non-rigid, and the two are closely related.

Adam. With Adam, the scalar learning rates are replaced by diagonal per-parameter preconditioners D_z and $D_w^{(\ell)}$, giving $\mathbf{K}_\theta = \mathcal{P}^{(L)} D_z \mathcal{P}^{(L)\top} + \sum_\ell J^{(\ell)} D_w^{(\ell)} J^{(\ell)\top}$. This leaves the range and locality of each term unchanged. Heuristically, ignoring momentum and assuming the second-moment estimates are near stationarity, the entries of $D_w^{(\ell)}$ scale inversely with the gradient magnitude $|\nabla_{w_{q,a}^{(\ell)}[k]} \mathcal{L}| = |(\phi_{q,a,k}^{(\ell)} \otimes u^{(\ell)}(q))^\top \mathcal{P}^{(\ell-1)\top} g|$, which is itself proportional to $\|u^{(\ell)}(q)\|$. The magnitude dependence of the adaptive term is therefore reduced from quadratic to approximately linear, but not removed.

Fixed output smoothing. In LCPReg-opt and LCPReg-conv, a fixed linear smoothing operator $\mathcal{S}$ is applied to the output of the pyramid (Appendices C.1 and C.2). The kernel then becomes $\mathcal{S}\mathbf{K}_\theta\mathcal{S}^\top$, which further attenuates high frequencies but does not alter the structure of either term.

C Implementation Details

In this section we provide the implementation details of the three baseline methods we integrate our LCP with, together with the exact hyperparameter settings that are specific to each one.

C.1 LCPReg-opt and NODEO

NODEO (Wu et al., 2022) represents the transformation as the flow of a neural ODE, $\mathrm{d}q/\mathrm{d}t = \mathcal{K}\, v_\vartheta(q(t), t)$, where q is the cloud of voxel coordinates, v_ϑ is a convolutional network, and $\mathcal{K}$ is a Gaussian filter. In LCPReg-opt we replace the network and the ODE integration with the pyramid of equation 2, while keeping NODEO's Gaussian smoothing, similarity measure and regularization terms. The smoothing is applied to the output of the finest level, so that in this setting the representation is

$$u_\theta = \mathcal{K} * u^{(0)}, \qquad \varphi_\theta = \mathrm{Id} + u_\theta, \tag{19}$$

where $\mathcal{K}$ is a uniform box filter that replaces each displacement vector by the mean over the cubic window $B(x)$ of k^3 voxels centered at x,

$$(\mathcal{K} * u)(x) = \frac{1}{k^3} \sum_{y \in B(x)} u(y), \tag{20}$$

with the field zero-padded outside $\Omega^{(0)}$. Although NODEO describes this operator as a Gaussian filter, the public implementation uses a box filter, which we adopt here. The filter acts as a hard constraint on spatial smoothness, encouraging each voxel to move similarly to its neighbors. Throughout this section we write $N = |\Omega^{(0)}|$ and $I_W = I_M \circ \varphi_\theta$ for the warped moving image.

Similarity. The similarity term is based on local normalized cross-correlation (LNCC), $\mathcal{L}_{sim}(I_F, I_W) = 1 - \mathrm{LNCC}(I_F, I_W)$, with

$$\mathrm{LNCC}(I_F, I_W) = \frac{1}{N} \sum_{x \in \Omega^{(0)}} \frac{\sum_{y \in W(x)} \big(I_F(y) - \bar{I}_F(x)\big)\big(I_W(y) - \bar{I}_W(x)\big)}{\sqrt{\sum_{y \in W(x)} \big(I_F(y) - \bar{I}_F(x)\big)^2 \sum_{y \in W(x)} \big(I_W(y) - \bar{I}_W(x)\big)^2}}, \tag{21}$$

where $W(x)$ is a cubic window of w^3 voxels centered at x, and $\bar{I}_F(x)$ and $\bar{I}_W(x)$ are the local means of the two images over $W(x)$. Following NODEO, we set $w = 21$.

Regularization. The regularization term combines three penalties,

$$\lambda_{reg}\,\mathcal{L}_{reg}(\varphi_\theta) = \lambda_J\,\mathcal{L}_J + \lambda_{mag}\,\mathcal{L}_{mag} + \lambda_{smt}\,\mathcal{L}_{smt}. \tag{22}$$

The first term penalizes folding through the Jacobian determinant of the transformation,

$$\mathcal{L}_J = \frac{1}{N}\sum_{x\in\Omega^{(0)}} \Big(\mathrm{ReLU}\big(\epsilon - \det D\varphi_\theta(x)\big)\Big)^2, \qquad D\varphi_\theta = \mathbf{I} + \nabla u_\theta, \tag{23}$$

where ∇u_θ is computed by finite differences. The rectifier restricts the penalty to determinants below a small margin $\epsilon > 0$, which slightly overcorrects so that the optimization is pushed away from the fold boundary rather than onto it. The squared (L_2) penalty discourages large individual folds rather than promoting sparsity in their number. The second term penalizes the magnitude of the displacement,

$$\mathcal{L}_{mag} = \frac{1}{N}\sum_{x\in\Omega^{(0)}} \big\|u_\theta(x)\big\|_2^2. \tag{24}$$

In NODEO this term is the kinetic energy of the flow, $\int_0^1 \|\mathcal{K}v_\vartheta\|_2^2\,\mathrm{d}t$. Under the single Euler step used in the original experiments, the smoothed velocity coincides with the displacement, and the two formulations are equivalent. The third term penalizes the spatial gradient of the displacement,

$$\mathcal{L}_{smt} = \frac{1}{N}\sum_{x\in\Omega^{(0)}} \big\|\nabla u_\theta(x)\big\|_2^2, \tag{25}$$

again approximated by finite differences. Although $\mathcal{K}$ already enforces smoothness, $\mathcal{L}_{smt}$ further reduces local irregularities that the filter alone does not remove.

Hyperparameters. Following the default configuration of NODEO, we set $\lambda_J = 2.5$, $\lambda_{mag} = 0.00005$, $\lambda_{smt} = 0.05$ and $\epsilon = 0.5$, and use a box filter of size $k = 15$. The window size of the LNCC similarity terms is set to 21. Both NODEO and LCPReg-opt are optimized for 300 iterations with Adam.

C.2 LCPReg-conv and convexAdam

convexAdam (Siebert et al., 2024) registers an image pair in two stages. Both images are first mapped to modality-independent neighborhood descriptors (MIND) (Heinrich et al., 2012), $F_F = \Phi_{r,d}(I_F)$ and $F_M = \Phi_{r,d}(I_M)$, where $\Phi_{r,d} : \Omega \to \mathbb{R}^C$ computes $C = 12$ self-similarity channels per voxel with patch radius r and dilation d. A coupled convex discrete optimization over a correlation volume of these features produces an initial displacement field $\hat{u}$, which is then refined by Adam-based instance optimization. In LCPReg-conv the first stage is left unchanged, and its output $\hat{u}$ is used to warm start the pyramid as described in Section 3.4. The pyramid replaces the displacement grid that convexAdam optimizes directly in the second stage, while the similarity measure, regularization and smoothing described below are kept.

Displacement model. The Adam stage operates on a grid Ω_g with spacing of g voxels, on which the features are average-pooled to F_F^g and F_M^g. Before the loss is evaluated at each iteration, the displacement is passed through a smoothing operator $\mathcal{S}$ that acts as a B-spline deformation model,

$$u_\theta = \mathcal{S}\big(u^{(0)}\big), \qquad \mathcal{S} = \underbrace{\mathcal{B}\circ\cdots\circ\mathcal{B}}_{n_{\mathcal{B}}}, \qquad (\mathcal{B}*u)(x) = \frac{1}{27}\sum_{y\in B_3(x)} u(y), \tag{26}$$

where $\mathcal{B}$ is a 3^3 box filter, i.e. a zeroth-order B-spline kernel, and $B_3(x)$ is the $3\times3\times3$ neighborhood of x. Recursive application of the box filter approximates a B-spline kernel of order $n_{\mathcal{B}} - 1$; for small smoothing scales convexAdam alternatively uses a Gaussian kernel. The smoothed field is then upsampled trilinearly to the image resolution and scaled by g to express the displacement in image voxels.

Similarity. The similarity term is the mean squared difference between the MIND features of the fixed image and the warped features of the moving image,

$$\mathcal{L}_{sim} = \frac{1}{|\Omega_g|} \sum_{x \in \Omega_g} \left\| F_M^g\big(x + u_\theta(x)\big) - F_F^g(x) \right\|_2^2, \tag{27}$$

where $F_M^g(x + u_\theta(x))$ is evaluated by trilinear interpolation, which makes the term differentiable with respect to u_θ. Because MIND descriptors are invariant to local contrast, a sum-of-squared-differences measure on the features remains appropriate where one on the raw intensities would not.

Regularization. Smoothness is enforced by a diffusion penalty on the smoothed displacement,

$$\lambda_{reg}\, \mathcal{L}_{reg} = \frac{\lambda}{|\Omega_g|} \sum_{x \in \Omega_g} \left\| \nabla u_\theta(x) \right\|_2^2, \tag{28}$$

with ∇u_θ approximated by forward finite differences along each axis. The diffusion term and the B-spline model are complementary. $\mathcal{S}$ restricts the field to a smooth function space, while $\mathcal{L}_{reg}$ penalizes the remaining high-frequency variation within it. Finer grids ($g = 1$) therefore require a larger λ to compensate for their added flexibility.

Hyperparameters. We use the default MIND configuration of convexAdam: MIND radius $r = 1$ and dilation $d = 2$, coarse grid spacing $grid_sp = 2$ and displacement half-width $disp_hw = 2$ for the convex stage, and Adam grid spacing $grid_sp_adam = 1$, regularization weight $\lambda = 1.6$. Post optimization smoothing is not applied. The optimization runs for 80 iterations.

C.3 LCPReg-uGI and uniGradICON

uniGradICON (Tian et al., 2024) is a foundation registration model based on the GradICON architecture (Tian et al., 2023): a three-level multi-resolution cascade of U-Nets operating at $\frac{1}{4}$, $\frac{1}{2}$ and full resolution, followed by a fourth U-Net at full resolution. Given an ordered pair of images, the network predicts a transformation that maps the first image into the space of the second, and swapping the inputs yields the reverse transformation. We denote the two directions by

$$\varphi_{MF} = \mathrm{Id} + u_{MF}, \qquad \varphi_{FM} = \mathrm{Id} + u_{FM}, \tag{29}$$

such that $I_M \circ \varphi_{MF}$ is aligned with I_F and $I_F \circ \varphi_{FM}$ is aligned with I_M. All images are resized to 175^3 voxels and displacements are expressed in normalized coordinates on the unit cube.

Symmetric objective. Both at training time and during instance optimization, uniGradICON minimizes a symmetric similarity term together with the gradient inverse consistency (GradICON) regularizer,

$$\mathcal{L} = \mathcal{L}_{sim}\big(I_F,\, I_M \circ \varphi_{MF}\big) + \mathcal{L}_{sim}\big(I_M,\, I_F \circ \varphi_{FM}\big) + \frac{\lambda}{N} \sum_{x \in \Omega^{(0)}} \Big\| \nabla\big(\varphi_{MF} \circ \varphi_{FM}\big)(x) - \mathbf{I} \Big\|_F^2, \tag{30}$$

where $\mathcal{L}_{sim} = 1 - \mathrm{LNCC}$ as in equation 21, $\mathbf{I}$ is the 3×3 identity matrix, and $\| \cdot \|_F$ is the Frobenius norm. The composition $\varphi_{MF} \circ \varphi_{FM}$ is evaluated by trilinear interpolation of φ_{MF} at the points $\varphi_{FM}(x)$, and its Jacobian is approximated by finite differences. Rather than penalizing the deformation itself, as a diffusion regularizer does, the regularizer only requires the two maps to be approximate inverses of each other up to first order. The weaker constraint lets the data determine which deformations are plausible, which is what allows a single set of hyperparameters to be used across tasks.

Forward and backward pyramids. In the original instance optimization, both maps are produced by the same network with swapped inputs, so a single set of weights parameterizes both directions and is refined jointly under equation 30. The LCP representation does not take the images as input, and a single pyramid can therefore only represent one of the two maps. In LCPReg-uGI we instead optimize two independent pyramids,

$$u_{MF} = u_{\theta_{MF}}, \qquad u_{FM} = u_{\theta_{FM}}, \qquad \theta = \theta_{MF} \cup \theta_{FM}, \tag{31}$$

each structured as in equation 3, and minimize equation 30 jointly over both. The two pyramids interact only through the inverse consistency term, which couples their gradients: the update to each field depends on the current state of the other. Each pyramid is warm started, as described in Section 3.4, from the corresponding uniGradICON prediction, $\hat{u}_{MF}$ from the ordered pair (I_M, I_F) and $\hat{u}_{FM}$ from (I_F, I_M), so that optimization begins exactly at the output of the pretrained network in both directions. The two pyramids double the number of optimized parameters, but even so the total remains far below the number of network weights optimized by the original instance optimization. After optimization, only the forward field u_{MF} is used for evaluation.

Hyperparameters. The original work defines a single hyperparameter for its regularization $\lambda = 1.5$, which we set to be the same value. Optimization is conducted for 50 iterations, and the LNCC window size is set to 21.