Metric Learning for Image Registration

Marc Niethammer
UNC Chapel Hill
mn@cs.unc.edu

Roland Kwitt
University of Salzburg
roland.kwitt@gmail.com

François-Xavier Vialard
LIGM, UPEM
francois-xavier.vialard@u-pem.fr

Abstract

Image registration is a key technique in medical image analysis to estimate deformations between image pairs. A good deformation model is important for high-quality estimates. However, most existing approaches use ad-hoc deformation models chosen for mathematical convenience rather than to capture observed data variation. Recent deep learning approaches learn deformation models directly from data. However, they provide limited control over the spatial regularity of transformations. Instead of learning the entire registration approach, we learn a spatially-adaptive regularizer within *a registration model. This allows controlling the desired level of regularity and preserving structural properties of a registration model. For example, diffeomorphic transformations can be attained. Our approach is a radical departure from existing deep learning approaches to image registration by embedding a deep learning model in an optimization-based registration algorithm to parameterize and data-adapt the registration model itself. Source code is publicly-available at* `https://github.com/uncbiag/registration`.

1. Introduction

Image registration is important in medical image analysis tasks to capture subtle, local deformations. Consequently, transformation models [21], which parameterize these deformations, have large numbers of degrees of freedom, ranging from B-spline models with many control points, to non-parametric approaches [30] inspired by continuum mechanics. Due to the large number of parameters of such models, deformation fields are typically regularized by *directly* penalizing local changes in displacement or, more *indirectly*, in velocity field(s) parameterizing a deformation.

Proper regularization is important to obtain high-quality deformation estimates. Most existing work simply imposes the same spatial regularity *everywhere* in an image. This is unrealistic. For example, consider registering brain images with different ventricle sizes, or chest images with a moving lung, but a stationary rib cage, where different deformation scales are present in different image regions. Parameterizing such deformations from first principles is difficult and may be impossible for between-subject registrations. Hence, it is desirable to *learn local regularity* from data. One could replace the registration model entirely and learn a parameterized regression function f_Θ from a large dataset. At inference time, this function then maps a moving image to a target image [12]. However, regularity of the resulting deformation does not arise naturally in such an approach and typically needs to be enforced after the fact.

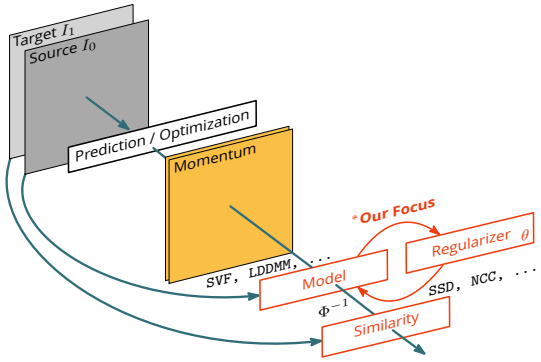

Figure 1: Architecture of our registration approach. We jointly optimize over the momentum, parameterizing the deformation Φ, and the parameters, θ, of a convolutional neural net (CNN). The CNN *locally* predicts multi-Gaussian kernel pre-weights which specify the regularizer. This approach constructs a metric such that diffeomorphic transformations can be assured in the continuum.

Existing non-parametric deformation models already yield good performance, are well understood, and use globally parameterized regularizers. Hence, we advocate building upon these models and to learn appropriate *localized* parameterizations of the regularizer by leveraging large samples of training data. This strategy not only retains theoretical guarantees on deformation regularity, but also makes it possible to encode, in the metric, the intrinsic deformation model as supported by the data.

Contributions. Our approach deviates from current approaches for (predictive) image registration in the following sense. Instead of replacing the entire registration model by

a regression function, we retain the underlying registration model and *learn* a spatially-varying regularizer. We build on top of a new *vector momentum-parameterized stationary velocity field (vSVF)* registration model which allows us to guarantee that deformations are diffeomorphic even when using a learned regularizer. Our approach jointly optimizes the regularizer (parameterized by a deep network) and the registration parameters of the vSVF model. We show state-of-the art registration results and evidence for locally varying deformation models in real data.

Overview. Fig. 1 illustrates our key idea. We start with an initial momentum parameterization of a registration model, in particular, of the vSVF. Such a parameterization is important, because it allows control over deformation regularity *on top of* the registration parameters. For a given source-target image-pair (I_0, I_1), we optimize over the momentum to obtain a spatial transformation Φ such that $I_0 \circ \Phi^{-1} \approx I_1$. As the mapping from momentum to Φ is influenced by a regularizer expressing what transformations are desirable, we jointly optimize over the regularizer parameters, θ, *and* the momentum. Specifically, we use a spatially-adaptive regularizer, parameterized by a regression model (here, a CNN). Our approach naturally combines with a prediction model, *e.g.*, [48], to obtain the momentum from a source-target image pair (avoiding optimization at runtime). Here, we *numerically optimize* over the momentum for simplicity and leave momentum prediction to future work.

Organization. In §2, we review registration models, relations to our proposed approach and introduce the vSVF model. §3 describes our metric learning registration approach and §4 discusses experimental results. Finally, §5 summarizes the main points. *Additional details can be found in the supplementary material.*

2. Background on image registration

Image registration is typically formulated as an optimization problem of the form

$$\gamma^* = \operatorname*{argmin}_{\gamma} \lambda\, \text{Reg}[\Phi^{-1}(\gamma)] + \text{Sim}[I_0 \circ \Phi^{-1}(\gamma), I_1]. \quad (2.1)$$

Here, γ parameterizes the deformation, Φ, $\lambda \geq 0$, $\text{Reg}[\cdot]$ is a penalty encouraging spatially regular deformations and $\text{Sim}[\cdot,\cdot]$ penalizes dissimilarities between two images (*e.g.*, sum-of-squared differences, cross-correlation or mutual information [20]). For low-dimensional parameterizations of Φ, *e.g.*, for affine or B-spline [36, 29] models, a regularizer may not be necessary. However, non-parametric registration models [30] represent deformations via displacement, velocity, or momentum vector fields and require regularization for a well-posed optimization problem.

In medical image analysis, diffeomorphic transformations, Φ, are often desirable to smoothly map between subjects or between subjects and an atlas space for local analyses. Diffeomorphisms can be guaranteed by estimating sufficiently smooth [14] static or time-varying velocity fields, v. The transformation is then obtained via time integration, *i.e.*, of $\Phi_t(x,t) = v \circ \Phi(x,t)$ (subscript $_t$ indicates a time derivative). Examples of such methods are the static velocity field (SVF) [42] and the large displacement diffeomorphic metric mapping (LDDMM) registration models [4, 44, 18, 1].

Non-parametric registration models require optimization over high-dimensional vector fields, often with millions of unknowns in 3D. Hence, numerical optimization can be slow. Recently, several approaches which learn a regression model to predict registration parameters from large sets of image pairs have emerged. Initial models based on deep learning [13, 24] were proposed to speed-up optical flow computations [22, 3, 8, 7, 49, 40]. Non-deep-learning approaches for the regression of registration parameters have also been studied [46, 45, 10, 9, 16]. These approaches typically have no guarantees on spatial regularity or may not straightforwardly extend to 3D image volumes due to memory constraints. Alternative approaches have been proposed which can register 3D images [35, 38, 12, 23, 2, 15] and assure diffeomorphisms [47, 48]. In these approaches, costly numerical optimization is only required during training of the regression model. Both end-to-end approaches [12, 23, 2, 15] and approaches requiring the desired registration parameters during training exist [47, 48, 35]. As end-to-end approaches differentiate through the transformation map, Φ, they were motivated by the spatial transformer work [25].

One of the main conceptual downsides of current regression approaches is that they either explicitly encode regularity when computing the registration parameters to obtain the training data [47, 48, 35], impose regularity as part of the loss [23, 2, 15] to avoid ill-posedness, or use low-dimensional parameterizations to assure regularity [38, 12]. Consequentially, these models *do not* estimate a deformation model from data, but instead impose it by choosing a regularizer. Ideally, one would like a registration model which (1) regularizes according to deformations present in data, (2) is fast to compute via regression and which (3) retains desirable theoretical properties of the registration model (*e.g.*, guarantees diffeomorphisms) even when predicting registration parameters via regression.

Approaches which predict momentum fields [47, 48] are fast and can guarantee diffeomorphisms. Yet, no model exists which estimates a local spatial regularizer of a form that guarantees diffeomorphic transformations and that can be combined with a fast regression formulation. Our goal is to close this gap via a momentum-based registration variant. While we will not explore regressing the momentum parameterization here, such a formulation is expected to be straightforward, as our proposed model has a momentum-

parameterization similar to what has already been used successfully for regression with a deep network [48].

2.1. Fluid-type registration algorithms

To capture large deformations and to guarantee diffeomorphic transformations, registration methods inspired by fluid mechanics have been highly successful, *e.g.*, in neuroimaging [1]. Our model follows this approach. The map Φ is obtained via time-integration of a sought-for velocity field $v(x,t)$. Specifically, $\Phi_t(x,t) = v(\Phi(x,t),t)$, $\Phi(x,0) = x$. For sufficiently smooth (*i.e.*, sufficiently regularized) velocity fields, v, one obtains diffeomorphisms [14]. The corresponding instance of Eq. (2.1) is

$$v^* = \operatorname*{argmin}_v \lambda \int_0^1 \|v\|_L^2 \, \mathrm{d}t + \mathrm{Sim}[I_0 \circ \Phi^{-1}(1), I_1], \text{ s.t.}$$
$$\Phi_t^{-1} + D\Phi^{-1} v = 0, \text{ and } \Phi^{-1}(0) = \mathrm{id} \ .$$

Here, D denotes the Jacobian (of Φ^{-1}), $\|v\|_L^2 = \langle L^\dagger L v, v\rangle$ is a spatial norm defined using the differential operator L and its adjoint $L^\dagger$. A specific L implies an expected deformation model. In its simplest form, L is *spatially-invariant* and encodes a desired level of smoothness. As the vector-valued momentum, m, is given by $m = L^\dagger L v$, one can write the norm as $\|v\|_L^2 = \langle m, v\rangle$.

In LDDMM [4], one seeks time-dependent vector fields $v(x,t)$. A simpler, but less expressive, approach is to use *stationary velocity fields* (SVF), $v(x)$, instead [35]. While SVF's are optimized directly over the velocity field v, we propose a *vector momentum SVF (vSVF)* formulation, *i.e.*,

$$\begin{aligned} m^* = \operatorname*{argmin}_{m_0} & \lambda \langle m_0, v_0\rangle + \mathrm{Sim}[I_0 \circ \Phi^{-1}(1), I_1] \\ & \text{s.t. } \Phi_t^{-1} + D\Phi^{-1} v = 0 \\ & \Phi^{-1}(0) = \mathrm{id}, \text{ and } v_0 = (L^\dagger L)^{-1} m_0 \ , \end{aligned} \tag{2.2}$$

which is optimized over the vector momentum m_0. vSVF is a simplification of vector momentum LDDMM [44]. We use vSVF for simplicity, but our approach directly translates to LDDMM and is motivated by the desire for LDDMM regularizers adapting to a deforming image.

3. Metric learning

In practice, L is predominantly chosen to be spatially-invariant. Only limited work on *spatially-varying* regularizers exists [33, 31, 39] and even less work focuses on *estimating* a spatially-varying regularizer. A notable exception is the estimation of a spatially-varying regularizer in atlas-space [43] which builds on a left-invariant variant of LDDMM [37]. Instead, our goal is to *learn* a spatially-varying regularizer which takes as inputs a momentum vector field and an image and computes a smoothed vector field. Therefore, our approach, not only leads to spatially varying metrics but can address pairwise registration, contrary to atlas-based learning methods, and it can adapt to deforming images during time integration for LDDMM[1]. We focus on extensions to the multi-Gaussian regularizer [34] as a first step, but note that learning more general regularization models would be possible.

3.1. Parameterization of the metrics

Metrics on vector fields of dimension M are positive semi-definite (PSD) matrices of M^2 coefficients. Directly learning these M^2 coefficients is impractical, since for typical 3D image volumes M is in the range of millions. We therefore restrict ourselves to a class of spatially-varying mixtures of Gaussian kernels.

Multi-Gaussian kernels. It is customary to directly specify the map from momentum to vector field via Gaussian smoothing, *i.e.*, $v = G \star m$ (here, $\star$ denotes convolution). In practice, multi-Gaussian kernels are desirable [34] to capture multi-scale aspects of a deformation, where

$$v = \left(\sum_{i=0}^{N-1} w_i G_i \right) \star m \ , \ w_i \geq 0, \ \sum_{i=0}^{N-1} w_i = 1 \ . \tag{3.1}$$

G_i is a normalized Gaussian centered at zero with standard deviation σ_i and w_i is a positive weight. The class of kernels that can be approximated by such a sum is already large[2]. A naïve approach to estimate the regularizer would be to learn w_i and σ_i. However, estimating either the variances or weights benefits from adding penalty terms to encourage desired solutions. Assume, for simplicity, that we have a single Gaussian, G, $v = G \star m$, with standard deviation σ. As the Fourier transform is an L^2 isometry, we can write

$$\begin{aligned} \int m(x)^\top v(x) \, \mathrm{d}x &= \langle m, v\rangle = \langle \hat{m}, \hat{v}\rangle \\ &= \langle \hat{v}/\hat{G}, \hat{v}\rangle = \int e^{\pi^2 2\sigma^2 k^\top k} v(k)^\top v(k) \, \mathrm{d}k \ , \end{aligned} \tag{3.2}$$

where $\hat{\cdot}$ denotes the Fourier transform and k the frequency. Since $\hat{G}$ is a Gaussian without normalization constant, it follows that we need to explicitly penalize small σ's if we want to favor smoother transformations (with large σ's). Indeed, the previous formula shows that a constant velocity field has the same norm for every positive σ. More generally, in theory, it is possible to reproduce a given deformation by the use of different kernels. Therefore, a penalty function on the parameterizations of the kernel is desirable. We design this penalty via a simple form of *optimal mass transport (OMT)* between the weights, as explained in the following.

[1] We use vSVF here and leave LDDMM as future work.

[2] All the functions $h : \mathbb{R}_{>0} \mapsto \mathbb{R}$ such that $h(|x-y|)$ is a kernel on $\mathbb{R}^d$ for every $d \geq 1$ are in this class.

OMT on multi-Gaussian kernel weights. Consider a multi-Gaussian kernel as in Eq. (3.1), with standard deviations $0 < \sigma_0 \leq \sigma_1 \leq \cdots \leq \sigma_{N-1}$. It would be desirable to obtain *simple* transformations explaining deformations with large standard deviations. Interpreting the multi-Gaussian kernel weights as a distribution, the most desirable configuration would be $w_{i\neq N-1} = 0,\ w_{N-1} = 1$, *i.e.*, using only the Gaussian with largest variance. We want to penalize weight distributions deviating from this configuration, with the largest distance given to $w_0 = 1,\ w_{i\neq 0} = 0$. This can be achieved via an *OMT penalty*. Specifically, we define this penalty on $w = [w_0, \ldots, w_{N-1}]$ as

$$\mathrm{OMT}(w) = \sum_{i=0}^{N-1} w_i \left| \log \frac{\sigma_{N-1}}{\sigma_i} \right|^r , \tag{3.3}$$

where $r \geq 1$ is a chosen power. In the following, we set $r = 1$. This penalty is zero if $w_{N-1} = 1$ and will have its largest value for $w_0 = 1$. It can be standardized as

$$\widehat{\mathrm{OMT}}(w) = \left| \log \frac{\sigma_{N-1}}{\sigma_0} \right|^{-r} \sum_{i=0}^{N-1} w_i \left| \log \frac{\sigma_{N-1}}{\sigma_i} \right|^r \tag{3.4}$$

with $\widehat{\mathrm{OMT}}(w) \in [0, 1]$ by construction.

Localized smoothing. This multi-Gaussian approach is a *global* regularization strategy, *i.e.*, the *same* multi-Gaussian kernel is applied *everywhere*. This leads to efficient computations, but does not allow capturing localized changes in the deformation model. We therefore introduce *localized* multi-Gaussian kernels, embodying the idea of tissue-dependent localized regularization. Starting from a sum of kernels $\sum_{i=0}^{N-1} w_i G_i$, we let the weights w_i vary spatially, *i.e.*, $w_i(x)$. To ensure diffeomorphic deformations, we set the weights $w_i(x) = G_{\sigma_{\text{small}}} \star \omega_i(x)$, where $\omega_i(x)$ are *pre-weights* which are convolved with a Gaussian with small standard deviation. An appropriate definition for how to use these weights to go from the momentum to the velocity is required to assure diffeomorphic transformations. Multiple approaches are possible. We use the model

$$\begin{aligned} v_0(x) &\overset{\text{def.}}{=} (K(w) \star m_0)(x) \\ &= \sum_{i=0}^{N-1} \sqrt{w_i(x)} \int_y G_i(|x-y|) \sqrt{w_i(y)} m_0(y)\, \mathrm{d}y\,, \end{aligned} \tag{3.5}$$

which, for spatially constant $w_i(x)$, reduces to the standard multi-Gaussian approach. In fact, this model guarantees diffeomorphisms, as long as the pre-weights are not too degenerate, as ensured by our model described hereafter. This fact is proven in the supplementary material (A.1). Motivated by the physical interpretation of these pre-weights and by diffeomorphic registration guarantees, we require a spatial regularization of these pre-weights via TV or H^1. We use color-TV [6] for our experiments. As the spatial transformation is directly governed by the weights, we impose the OMT penalty locally. Based on Eq. (2.2), we optimize the following:

$$\begin{aligned} m^* = \operatorname*{argmin}_{m_0} \lambda \langle m_0, v_0 \rangle \ + \mathrm{Sim}[I_0 \circ \Phi^{-1}(1), I_1] + \\ \lambda_{\mathrm{OMT}} \int \widehat{\mathrm{OMT}}(w(x))\, \mathrm{dx} + \\ \lambda_{\mathrm{TV}} \sqrt{\sum_{i=0}^{N-1} \left(\int \gamma(\|\nabla I_0(x)\|) \|\nabla \omega_i(x)\|_2\, \mathrm{dx} \right)^2} , \end{aligned} \tag{3.6}$$

subject to the constraints $\Phi_t^{-1} + D\Phi^{-1} v = 0$ and $\Phi^{-1}(0) = \mathrm{id}$; $\lambda_{\mathrm{TV}}, \lambda_{\mathrm{OMT}} \geq 0$. The partition of unity defining the metric, intervenes in the L^2 scalar product $\langle m_0, v_0 \rangle$.

Further, in Eq. (3.6), the OMT penalty is integrated pointwise over the image-domain to support spatially-varying weights; $\gamma(x) \in \mathbb{R}^+$ is an *edge indicator function*, *i.e.*,

$$\gamma(\|\nabla I\|) = (1 + \alpha \|\nabla I\|)^{-1}, \text{ with } \alpha > 0\ ,$$

to encourage weight changes coinciding with image edges.

Local regressor. To learn the regularizer, we propose a *local regressor* from the image and the momentum to the pre-weights of the multi-Gaussian. Given the momentum m and image I (the source image I_0 for vSVF; $I(t)$ at time t for LDDMM) we learn a mapping of the form: $f_\theta : \mathbb{R}^d \times \mathbb{R} \to \Delta^{N-1}$, where Δ^{N-1} is the $N-1$ unit/probability simplex[3]. We will parametrize f_θ by a CNN in §3.1.1. The following attractive properties are worth pointing out:

1) The variance of the multi-Gaussian is bounded by the variances of its components. We retain these bounds and can therefore *specify a desired regularity level*.
2) A globally smooth set of velocity fields is still computed (in Fourier space) which allows capturing large-scale regularity without a large receptive field of the local regressor. Hence, the CNN can be kept efficient.
3) The local regression strategy makes the approach suitable for more general registration models, *e.g.*, for LDDMM, where one would like the regularizer to follow the *deforming* source image over time.

3.1.1 Learning the CNN regressor

For simplicity we use a fairly shallow CNN with two layers of filters and leaky ReLU (`lReLU`) [27] activations. In detail, the data flow is as follows: $\texttt{conv}(d+1, n_1) \to \texttt{BatchNorm} \to \texttt{lReLU} \to \texttt{conv}(n_1, N) \to \texttt{BatchNorm} \to$

[3] We only explore mappings dependent on the source image I_0 in our experiments, but more general mappings also depending on the momentum, for example, should be explored in future work.

`weighted-linear-softmax`. Here `conv`(a, b) denotes a convolution layer with a input channels and b output feature maps. We used $n_1 = 20$ for our experiments and convolutional filters of spatial size 5 (5×5 in 2D and $5 \times 5 \times 5$ in 3D). The `weighted-linear-softmax` activation function, which we formulated, maps inputs to Δ^{N-1}. We designed it such that it operates around a setpoint of weights w_i which correspond to the global weights of the multi-Gaussian kernel. This is useful to allow models to start training from a pre-specified, reasonable initial configuration of global weights, parameterizing the regularizer. Specifically, we define the *weighted linear softmax* $\sigma_w : \mathbb{R}^k \to \Delta^{N-1}$ as

$$\sigma_w(z)_j = \frac{\text{clamp}_{0,1}(w_j + z_j - \overline{z})}{\sum_{i=0}^{N-1} \text{clamp}_{0,1}(w_i + z_i - \overline{z})} \ , \qquad (3.7)$$

where $\sigma_w(z)_j$ denotes the j-th component of the output, $\overline{z}$ is the mean of the inputs, z, and the clamp function clamps the values to the interval $[0, 1]$. The removal of the mean in Eq. (3.7) assures that one moves along the probability simplex. That is, if one is outside the clamping range, then

$$\sum_{i=0}^{N-1} \text{clamp}_{0,1}(w_i + z_i - \overline{z}) = \sum_{i=0}^{N-1} w_i + z_i - \overline{z} = \sum_{i=0}^{N-1} w_i = 1$$

and consequentially, in this range, $\sigma_w(z)_j = w_j + z_j - \overline{z}$. This is linear in z and moves along the tangent plane of the probability simplex by construction. As a CNN with small initial weights will produce an output close to zero, the output of $\sigma_w(z)$ will initially be close to the desired setpoint weights, w_j, of the multi-Gaussian kernel. Once the pre-weights, $\omega_i(x)$, have been obtained via this CNN, we compute multi-Gaussian weights via Gaussian smoothing. We use $\sigma = 0.02$ in 2D and $\sigma = 0.05$ in 3D throughout all experiments (§4).

3.2. Discretization, optimization, and training

Discretization. We discretize the registration model using central differences for spatial derivatives and 20 steps in 2D (10 in 3D) of 4th order Runge-Kutta integration in time. Gaussian smoothing is done in the Fourier domain. The entire model is implemented in `PyTorch`[4]; all gradients are computed by automatic differentiation [32].

Optimization. Joint optimization over the momenta of a set of registration pairs and the network parameters is difficult in 3D due to GPU memory limitations. Hence, we use a customized variant of stochastic gradient descent (SGD) with Nesterov momentum (0.9) [41], where we split optimization variables (1) that are *shared* and (2) *individual* between registration-pairs. Shared parameters are for the CNN. Individual parameters are the momenta. Shared parameters are kept in memory and individual parameters, including their current optimizer states, are saved and restored for every random batch. We use a batch-size of 2 in 3D and 100 in 2D and perform 5 SGD steps for each batch. Learning rates are 1.0 and 0.25 for the individual and the shared parameters in 3D and 0.1 and 0.025 in 2D, respectively. We use gradient clipping (at a norm of one, separately for the gradients of the shared and the individual parameters) to help balance the energy terms. We use `PyTorch`'s `ReduceLROnPlateau` learning rate scheduler with a reduction factor of 0.5 and a patience of 10 to adapt the learning rate during training.

Curriculum strategy: Optimizing *jointly* over momenta, global multi-Gaussian weights and the CNN does not work well in practice. Instead, we train in two stages: (1) In the initial global stage, we pick a reasonable set of global Gaussian weights and optimize only over the momenta. This allows further optimization from a reasonable starting point. Local adaptations (via the CNN) can then immediately capture local effects rather than initially being influenced by large misregistrations. In all experiments, we chose these global weights to be linear with respect to their associated variances, *i.e.*, $w_i = \sigma_i^2 / (\sum_{j=0}^{N-1} \sigma_j^2)$. Then, (2) starting from the result of (1), we optimize over the momenta *and* the parameters of the CNN to obtain spatially-localized weights. We refer to stages (1) and (2) as *global* and *local* optimization, respectively. In 2D, we run global/local optimization for 50/100 epochs. In 3D, we run for 25/50 epochs. Gaussian variances are set to $\{0.01, 0.05, 0.1, 0.2\}$ for images in $[0, 1]^d$. We use normalized cross correlation (NCC) with $\sigma = 0.1$ as similarity measure. See §B of the supplementary material for further implementation details.

4. Experiments

We tested our approach on three dataset types: (1) 2D synthetic data with known ground truth (§4.1), (2) 2D slices of a real 3D brain magnetic resonance (MR) images (§4.2), and (3) multiple 3D datasets of brain MRIs (§4.3). Images are first affinely aligned and intensity standardized by matching their intensity quantile functions to the average quantile function over all datasets. We compute deformations at half the spatial resolution in 2D (0.4 times in 3D) and upsample Φ^{-1} to the original resolution when evaluating the similarity measure so that fine image details can be considered. This is not necessary in 2D, but essential in 3D to reduce GPU memory requirements. We use this approach in 2D for consistency.

All evaluations (except for §4.2 and for the within dataset results of §4.3) are with respect to a separate testing set. For testing, the previously learned regularizer parameters are fixed and numerical optimization is over momenta only (in particular, 250/500 iterations in 2D and 150/300 in 3D for global/local optimization).

[4] Available at `https://github.com/uncbiag/registration`, also including various other registration models such as LDDMM.

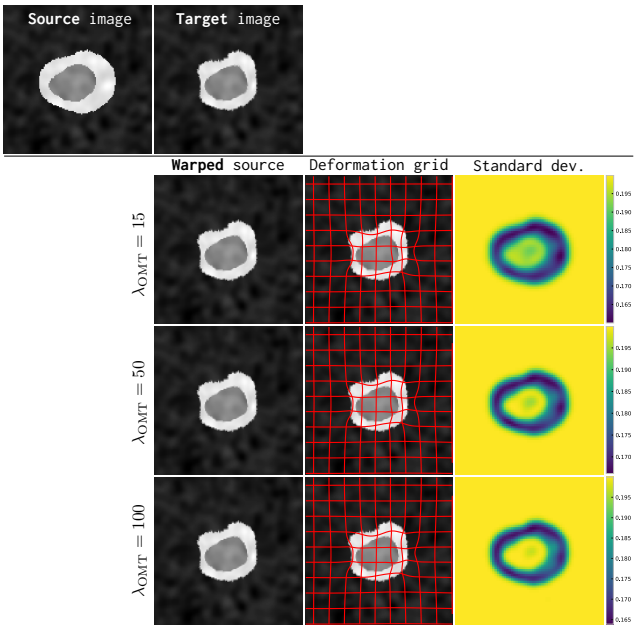

Figure 2: Example registration results using local metric optimization for the synthetic test data. Results are shown for different values of λ_{OMT} with the total variation penalty fixed to $\lambda_{\text{TV}} = 0.1$. Visual correspondence between the warped source and the target images are high for all settings. Estimates for the standard deviation stay largely stable. However, deformations are slightly more regularized for higher OMT penalties. This can also be seen based on the standard deviations (*best viewed zoomed*).

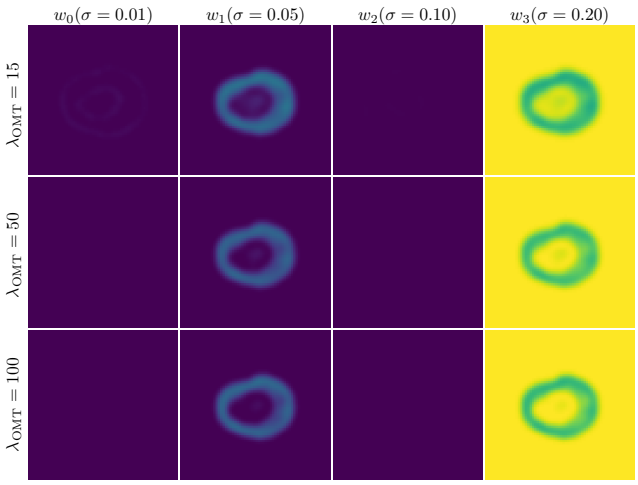

Figure 3: Estimated multi-Gaussian weights (blue=0; yellow=1) for the registrations in Fig. 2 w.r.t. different λ_{OMT}'s. Weight estimates are very stable across λ_{OMT}. While the overall standard deviation (Fig. 2) approximates the ground truth, the weights for the outer ring differ (ground truth weights are $[0.05, 0.55, 0.3, 0.1]$) from the ground truth. They approximately match for the background and the interior (ground truth $[0, 0, 0, 1]$).

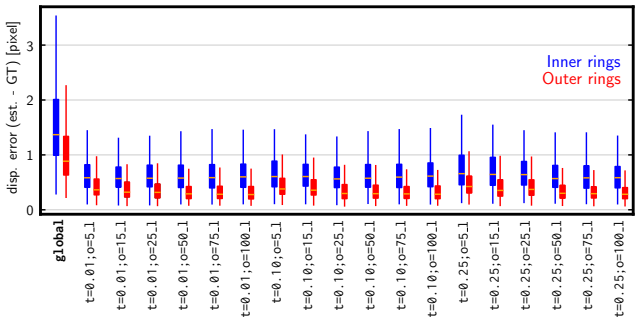

Figure 4: *Displacement error* (in pixel) with respect to the ground truth (GT) for various values of the total variation penalty, λ_{TV} (`t`) and the OMT penalty, λ_{OMT} (`o`). Results for the inner and the outer rings show subpixel registration accuracy for all *local* metric optimization results (`*_1`). Overall, local metric optimization substantially improves registrations over the results obtained via initial global multi-Gaussian regularization (`global`).

4.1. Results on 2D synthetic data

We created 300 synthetic 128×128 image pairs of randomly deformed concentric rings (see supplementary material, §C). Shown results are on 100 separate test cases.

Fig. 2 shows registrations for $\lambda_{\text{OMT}} \in \{15, 50, 100\}$. The TV penalty was set to $\lambda_{\text{TV}} = 0.1$. The estimated standard deviations, $\sigma^2(x) = \sum_{i=0}^{N-1} w_i(x)\sigma_i^2$, capture the trend of the ground truth, showing a large standard deviation (*i.e.*, high regularity) in the background and the center of the image and a smaller standard deviation in the outer ring. The standard deviations are stable across OMT penalties, but show slight increases with higher OMT values. Similarly, deformations get progressively more regular with larger OMT penalties (as they are regularized more strongly), but visually all registration results show very similar good correspondence. Note that while TV was used to train the model, the CNN output is not explicitly TV regularized, but nevertheless is able to produce largely constant regions that are well aligned with the boundaries of the source image. Fig. 3 shows the corresponding estimated weights. They are stable for a wide range of OMT penalties.

Finally, Fig. 4 shows displacement errors relative to the ground truth deformation for the interior and the exterior ring of the shapes. Local metric optimization significantly improves registration (over initial global multi-Gaussian regularization); these results are stable across a wide range of penalties with median displacement errors < 1 pixel.

4.2. Results on real 2D data

We used the same settings as for the synthetic dataset. However, here our results are for 300 random registration pairs of axial slices of the LPBA40 dataset [26].

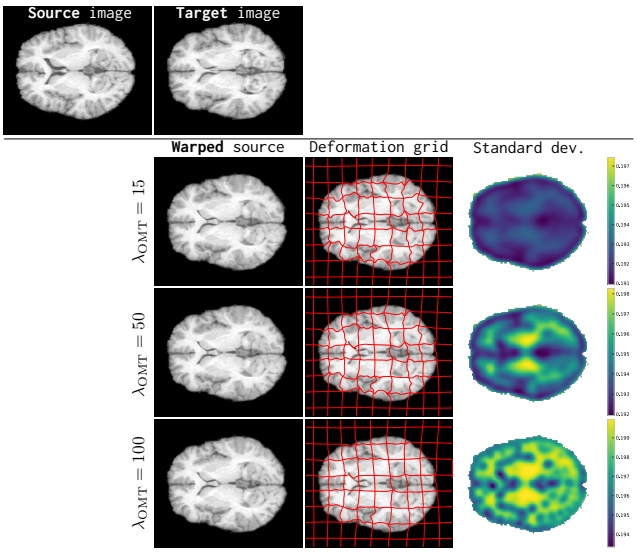

Figure 5: Example registration results using local metric optimization for different λ_{OMT}'s and $\lambda_{\text{TV}} = 0.1$. Visual correspondences between the warped source images and the target image are high for all values of the OMT penalty. Standard deviation estimates capture the variability of the ventricles and increased regularity with increased values for λ_{OMT} (*best viewed zoomed*).

Fig. 5 shows results for $\lambda_{\text{OMT}} \in \{15, 50, 100\}$; $\lambda_{\text{TV}} = 0.1$. Larger OMT penalties yield larger standard deviations and consequentially more regular deformations. Most regions show large standard deviations (high regularity), but lower values around the ventricles and the brain boundary – areas which may require substantial deformations.

Fig. 6 shows the corresponding estimated weights. We have no ground truth here, but observe that the model produces consistent regularization patterns for all shown OMT values ({15,50,100}) and allocates almost all weights to the Gaussians with the lowest and the highest standard deviations, respectively. As λ_{OMT} increases, more weight shifts from the smallest to the largest Gaussian.

4.3. Results on real 3D data

We experimented using the 3D CUMC12, MGH10, and IBSR18 datasets [26]. These datasets contain 12, 10, and 18 images. *Registration evaluations are with respect to all 132 registration pairs of CUMC12*. We use $\lambda_{\text{OMT}} = 50$, $\lambda_{\text{TV}} = 0.1$ for all tests[5]. Once the regularizer has been learned, we keep it fixed and optimize for the vSVF vector momentum. We trained independent models on CUMC12, MGH10, and IBSR18 using 132 image pairs on CUMC12, 90 image pairs on MGH10, and a random set of 150 image pairs on IBSR18. We tested the resulting three models on CUMC12 to assess the performance of a dataset-specific model and the ability to transfer models across datasets.

[5]We did not tune these parameters and better settings may be possible.

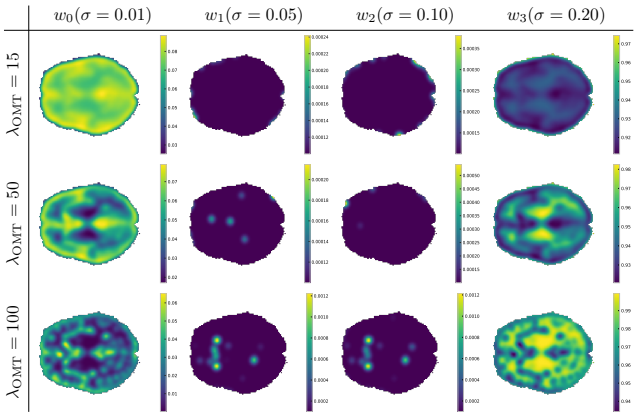

Figure 6: Estimated multi-Gaussian weights for different λ_{OMT} for real 2D data. Weights are mostly allocated to the Gaussian with the largest standard deviation (see colorbars; best viewed zoomed). A shift from w_0 to w_3 can be observed for larger values of λ_{OMT}. While weights shift between OMT setting, the ventricle area is always associated with more weight on w_0 (*best viewed zoomed*).

Tab. 1 and Fig. 7 compare to the registration methods in [26] and across different stages of our approach for different training/testing pairs. We also list the performance of the most recent VoxelMorph (`VM`) variant [11]. We kept the original architecture configuration, swept over a selection of VoxelMorph's hyperparameters and report the best results here. Each VoxelMorph model was trained for 300 epochs which, in our experiments, was sufficient for convergence. Overall, our approach shows the best results among all models when trained/tested on CUMC12 (`c/c local`); though results are not significantly better than for SyN, SPM5D, and VoxelMorph. Local metric optimization shows strong improvements over initial global multi-Gaussian regularization. Models trained on MGH10 and IBSR18 (`m/c local` and `i/c local`) also show good performance, slightly lower than for the model trained on CUMC12 itself, but higher than all other competing methods. This indicates that the trained models transfer well across datasets. While the top competitor in terms of median overlap (SPM5D) produces outliers (cf. Fig. 7), our models do not. In case of VoxelMorph we observed that adding more training pairs (*i.e.*, using all pairs of IBSR18, MGH18 & LBPA40) did not improve results (*cf.* Tab. 1 `*/c VM`).

In Tab. 2, we list statistics for the determinant of the Jacobian of Φ^{-1} on CUMC12, where the model was also trained on. This illustrates how transformation regularity changes between the global and the local regularization approaches. As expected, the initial global multi-Gaussian regularization results in highly regular registrations (*i.e.*, determinant of Jacobian close to one). Local metric optimization achieves significantly improved target volume overlap measures (Fig. 7) while keeping good spatial regularity, clearly showing the utility of our local regularization model. Note

Method	mean	std	1%	5%	50%	95%	99%	p	MW-stat	sig?
FLIRT	0.394	0.031	0.334	0.345	0.396	0.442	0.463	<1e−10	17394.0	✓
AIR	0.423	0.030	0.362	0.377	0.421	0.483	0.492	<1e−10	17091.0	✓
ANIMAL	0.426	0.037	0.328	0.367	0.425	0.483	0.498	<1e−10	16925.0	✓
ART	0.503	0.031	0.446	0.452	0.506	0.556	0.563	<1e−4	11177.0	✓
Demons	0.462	**0.029**	0.407	0.421	0.461	0.510	0.531	<1e−10	15518.0	✓
FNIRT	0.463	0.036	0.381	0.410	0.463	0.519	0.537	<1e−10	15149.0	✓
Fluid	0.462	0.031	0.401	0.410	0.462	0.516	0.532	<1e−10	15503.0	✓
SICLE	0.419	0.044	0.300	0.330	0.424	0.475	0.504	<1e−10	17022.0	✓
SyN	0.514	0.033	0.454	0.460	0.515	0.565	0.578	0.073	9677.0	✗
SPM5N8	0.365	0.045	0.257	0.293	0.370	0.426	0.455	<1e−10	17418.0	✓
SPM5N	0.420	0.031	0.361	0.376	0.418	0.471	0.494	<1e−10	17160.0	✓
SPM5U	0.438	0.029	0.373	0.394	0.437	0.489	0.502	<1e−10	16773.0	✓
SPM5D	0.512	0.056	0.262	0.445	0.523	0.570	0.579	0.311	9043.0	✗
c/c VM	0.517	0.034	**0.456**	0.460	0.518	0.571	0.580	0.244	9211.0	✗
m/c VM	0.510	0.034	0.448	0.453	0.509	0.564	0.574	0.011	10197.0	✓
i/c VM	0.510	0.034	0.450	0.453	0.508	0.564	0.573	0.012	10170.0	✓
*/c VM	0.509	0.033	0.450	0.453	0.509	0.561	0.570	0.007	10318.0	✓
m/c global	0.480	0.031	0.421	0.430	0.482	0.530	0.543	<1e−10	13864.0	✓
m/c local	0.517	0.034	0.454	0.461	0.521	0.568	0.578	0.257	9163.0	✗
c/c global	0.480	0.031	0.421	0.430	0.482	0.530	0.543	<1e−10	13864.0	✓
c/c local	**0.520**	0.034	0.455	**0.463**	**0.524**	**0.572**	**0.581**	-	-	-
i/c global	0.480	0.031	0.421	0.430	0.482	0.530	0.543	<1e−10	13863.0	✓
i/c local	0.518	0.035	0.454	0.460	0.522	0.571	0.581	0.338	8972.0	✗

Table 1: Statistics for mean (over all labeled brain structures, disregarding the background) target overlap ratios on CUMC12 for different methods. Prefixes for results based on global and local regularization indicate training/testing combinations identified by first initials of the datasets. For example, `m/c` means trained/tested on MGH10/CUMC12. Statistical results are for the null-hypothesis of equivalent mean target overlap with respect to `c/c local`. Rejection of the null-hypothesis (at $\alpha = 0.05$) is indicated with a check-mark (✓). All p-values are computed using a paired one-sided Mann Whitney rank test [28] and corrected for multiple comparisons using the Benjamini-Hochberg [5] procedure with a family-wise error rate of 0.05. Best results are **bold**, showing that our methods exhibits state-of-the-art performance.

	mean	1%	5%	50%	95%	99%
Global	1.00(0.02)	0.60(0.07)	0.71(0.03)	0.98(0.03)	1.39(0.05)	1.69(0.14)
Local	0.98(0.02)	0.05(0.04)	0.24(0.03)	0.84(0.03)	2.18(0.07)	3.90(0.23)

Table 2: Mean (standard deviation) of *determinant of Jacobian* of Φ^{-1} for global and local regularization with $\lambda_{\text{TV}} = 0.1$ and $\lambda_{\text{OMT}} = 50$ for CUMC12 within the brain. Local metric optimization (local) improves target overlap measures (see Fig. 7) at the cost of less regular deformations than for global multi-Gaussian regularization. However, the reported determinants of Jacobian are still all positive, indicating no folding.

that all reported determinant of Jacobian values in Tab. 2 are positive, indicating no foldings, which is consistent with our diffeomorphic guarantees; though these are only guarantees for the continuous model at convergence, which do not consider potential discretization artifacts.

5. Conclusions

We proposed an approach to learn a *local* regularizer, parameterized by a CNN, which integrates with deformable registration models and demonstrates good performance on both synthetic and real data. While we used vSVF for computational efficiency, our approach could directly be integrated with LDDMM (resulting in local, time-varying regularization). It could also be integrated with predictive regis-

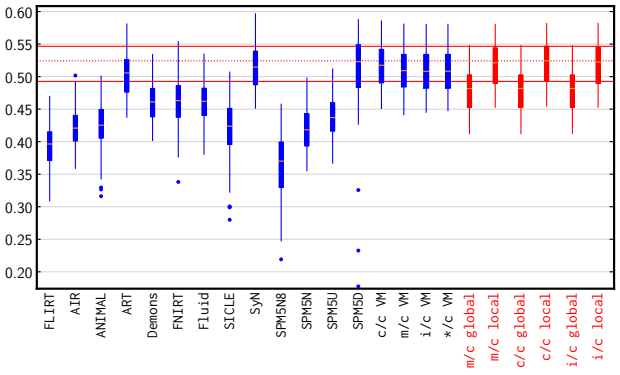

Figure 7: Mean target overlap ratios on CUMC12 (in 3D) with $\lambda_{\text{TV}} = 0.1$ and $\lambda_{\text{OMT}} = 50$. Our approach (marked red) gives the best result overall. Local metric optimization greatly improves results over the initial global multi-Gaussian regularization. Best results are achieved for the model that was trained on this dataset (`c/c local`), but models trained on MGH10 (`m/c local`) and on IBSR18 (`i/c local`) transfer well and show almost the same level of performance. The dashed line is the median mean target overlap ratio (*i.e.*, mean over all labels, median over all registration pairs).

tration approaches, *e.g.*, [48]. Such an integration would remove the computational burden of optimization at runtime, yield a fast registration model, allow end-to-end training and, in particular, promises to overcome the two key issues of current deep learning approaches to deformable image registration: (1) the lack of control over spatial regularity of approaches training mostly based on image similarities and (2) the inherent limitation on registration performance by approaches which try to learn optimal registration parameters for a given registration method and a *chosen* regularizer.

To the best of our knowledge, our model is the first approach to learn a local regularizer of a registration model by predicting local multi-Gaussian pre-weights. This is an attractive approach as it (1) allows retaining the theoretical properties of an underlying (well-understood) registration model, (2) allows imposing bounds on local regularity, and (3) focuses the effort on learning some aspects of the registration model from data, while refraining from learning the *entire* model which is inherently ill-posed. The estimated local regularizer might provide useful information in of itself and, in particular, indicates that a spatially non-uniform deformation model is supported by real data.

Much experimental and theoretical work remains. More sophisticated CNN models should be explored; the method should be adapted for fast end-to-end regression; more general parameterizations of regularizers should be studied (*e.g.*, allowing sliding), and the approach should be developed for LDDMM.

Acknowledgements. This work was supported by grants NSF EECS-1711776, NIH 1-R01-AR072013 and the Austrian Science Fund (FWF project P 31799).

References

[1] B. Avants, N. Tustison, and G. Song. Advanced normalization tools (ANTS). *Insight Journal*, 2:1–35, 2009.

[2] G. Balakrishnan, A. Zhao, M. Sabuncu, J. Guttag, and A. Dalca. An unsupervised learning model for deformable medical image registration. In *CVPR*, 2018.

[3] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM computing surveys (CSUR)*, 27(3):433–466, 1995.

[4] M. Beg, M. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *IJCV*, 61(2):139–157, 2005.

[5] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Series B Stat. Methodol.*, pages 289–300, 1995.

[6] P. Blomgren and T. Chan. Color TV: total variation methods for restoration of vector-valued images. *TMI*, 7(3):304–309, 1998.

[7] A. Borzi, K. Ito, and K. Kunisch. Optimal control formulation for determining optical flow. *SIAM J. Sci. Comput.*, 24(3):818–847, 2003.

[8] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[9] T. Cao, N. Singh, V. Jojic, and M. Niethammer. Semi-coupled dictionary learning for deformation prediction. In *ISBI*, 2015.

[10] C.-R. Chou, B. Frederick, G. Mageras, S. Chang, and S. Pizer. 2D/3D image registration using regression learning. *CVIU*, 117(9):1095–1106, 2013.

[11] A. Dalca, G. Balakrishnan, J. Guttag, and M. Sabuncu. Unsupervised learning for fast probabilistic diffeomorphic registration. In *MICCAI*, 2018.

[12] B. de Vos, F. Berendsen, M. Viergever, M. Staring, and I. Išgum. End-to-end unsupervised deformable image registration with a convolutional neural network. In *DLMIA*, 2017.

[13] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *CVPR*, 2015.

[14] P. Dupuis, U. Grenander, and M. I. Miller. Variational problems on flows of diffeomorphisms for image matching. *Q. Appl. Math.*, pages 587–600, 1998.

[15] J. Fan, X. Cao, Z. Xue, P.-T. Yap, and D. Shen. Adversarial similarity network for evaluating image alignment in deep learning based registration. In *MICCAI*, 2018.

[16] B. Gutierrez-Becker, D. Mateus, L. Peter, and N. Navab. Guiding multimodal registration with learned optimization updates. *MedIA*, 41:2–17, 2017.

[17] S. Hanson and L. Pratt. Comparing biases for minimal network construction with back-propagation. In *NIPS*, 1988.

[18] G. Hart, C. Zach, and M. Niethammer. An optimal control approach for deformable registration. In *CVPR Workshops*, pages 9–16, 2009.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[20] G. Hermosillo, C. Chefd'Hotel, and O. Faugeras. Variational methods for multimodal image matching. *IJCV*, 50(3):329–343, 2002.

[21] M. Holden. A review of geometric transformations for nonrigid body registration. *TMI*, 27(1):111, 2008.

[22] B. Horn and B. G. Schunck. Determining optical flow. *Artif. Intell.*, 17(1-3):185–203, 1981.

[23] Y. Hu, M. Modat, E. Gibson, N. Ghavami, E. Bonmati, C. Moore, M. Emberton, J. Noble, D. Barratt, and T. Vercauteren. Label-driven weakly-supervised learning for multimodal deformarle image registration. In *ISBI*, 2018.

[24] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[25] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015.

[26] A. Klein, J. Andersson, B. Ardekani, J. Ashburner, B. Avants, M.-C. Chiang, G. Christensen, D. Collins, J. Gee, P. Hellier, et al. Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. *Neuroimage*, 46(3):786–802, 2009.

[27] A. Maas, A. Hannun, and A. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.

[28] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.*, 18(1):50–60, 1947.

[29] M. Modat, G. Ridgway, Z. Taylor, M. Lehmann, J. Barnes, D. Hawkes, N. Fox, and S. Ourselin. Fast free-form deformation using graphics processing units. *Comput. Methods Programs Biomed.*, 98(3):278–284, 2010.

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

[31] D. Pace, S. Aylward, and M. Niethammer. A locally adaptive regularization based on anisotropic diffusion for deformable image registration of sliding organs. *TMI*, 32(11):2114–2126, 2013.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Workshop on Automatic Differentiation*, 2017.

[33] L. Risser, F.-X. Vialard, H. Baluwala, and J. Schnabel. Piecewise-diffeomorphic image registration: Application to the motion estimation between 3D CT lung images with sliding conditions. *MedIA*, 17(2):182–193, 2013.

[34] L. Risser, F.-X. Vialard, R. Wolz, M. Murgasova, D. Holm, and D. Rueckert. Simultaneous multi-scale registration using large deformation diffeomorphic metric mapping. *TMI*, 30(10):1746–1759, 2011.

[35] M. Rohé, M. Datar, T. Heimann, M. Sermesant, and X. Pennec. SVF-Net: Learning deformable image registration using shape matching. In *MICCAI*, 2017.

[36] D. Rueckert, L. I. Sonoda, C. Hayes, D. Hill, M. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *TMI*, 18(8):712–721, 1999.

[37] T. Schmah, L. Risser, and F.-X. Vialard. Left-invariant metrics for diffeomorphic image registration with spatially-varying regularisation. In *MICCAI*, 2013.

[38] H. Sokooti, B. de Vos, F. Berendsen, B. Lelieveldt, I. Išgum, and M. Staring. Nonrigid image registration using multi-scale 3D convolutional neural networks. In *MICCAI*, 2017.

[39] R. Stefanescu, X. Pennec, and N. Ayache. Grid powered nonlinear image registration with locally adaptive regularization. *MedIA*, 8(3):325–342, 2004.

[40] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.

[41] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

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

[43] F.-X. Vialard and L. Risser. Spatially-varying metric learning for diffeomorphic image registration: A variational framework. In *MICCAI*, 2014.

[44] F.-X. Vialard, L. Risser, D. Rueckert, and C. Cotter. Diffeomorphic 3D image registration via geodesic shooting using an efficient adjoint calculation. *IJCV*, 97(2):229–241, 2012.

[45] Q. Wang, M. Kim, Y. Shi, G. Wu, and D. Shen. Predict brain MR image registration via sparse learning of appearance and transformation. *MedIA*, 20(1):61–75, 2015.

[46] Q. Wang, M. Kim, G. Wu, and D. Shen. Joint learning of appearance and transformation for predicting brain MR image registration. In *IPMI*, 2013.

[47] X. Yang, R. Kwitt, and M. Niethammer. Fast predictive image registration. In *Deep Learning and Data Labeling for Medical Applications*, pages 48–57. 2016.

[48] X. Yang, R. Kwitt, M. Styner, and M. Niethammer. Quicksilver: Fast predictive image registration–a deep learning approach. *NeuroImage*, 158:378–396, 2017.

[49] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *Joint Pattern Recognition Symposium*, 2007.

A. Supplementary material

This supplementary material contains additional information describing our approach. §A.1 discusses the theoretical properties of our model and proves that the resulting spatial transformations are diffeomorphic in the continuum. Possible undesirable effects of the numerical discretization are not studied or addressed in this work. §B provides some critical implementation details for the CNN regressing the local pre-weights of the multi-Gaussian regularizer based on an input image. Lastly, §C provides details on how the synthetic data for our synthetic experiments was created.

A.1. Localized multi-Gaussian kernels

Starting from a sum of kernels $\sum_{i=0}^{N-1} w_i G_i$, we let the coefficient w_i be spatially varying. In order to ensure the diffeomorphic property of deformations, we set the weights $w_i(x) = G_{\sigma_{\text{small}}} \star \omega_i(x) + \varepsilon_i$, where $\omega_i(x)$ are pre-weights which are convolved with a Gaussian filter with small standard deviation and ε_i is a small positive real that acts as a constant offset parameter[6]. We have

$$\text{Reg}_{\text{vSVF}} = \lambda \langle m_0, v_0 \rangle + \lambda_{\text{OMT}} \int \text{OMT}(w(x))\,\mathrm{d}x + \lambda_{\text{TV}} \sqrt{\sum_{i=0}^{N-1} \left(\int \gamma(\|\nabla I_0(x)\|) \|\nabla \omega_i(x)\|_2 \,\mathrm{d}x \right)^2}, \quad \text{(A.1)}$$

where m_0 and v_0 are the initial momentum and vector field, respectively. Note that the partition of unity defining the metric, intervenes in the L^2 scalar product $\langle m_0, v_0 \rangle$ since, with $\varepsilon_i > 0$ a positive offset,

$$\begin{aligned} v_0(x) &= (K(w) \star m_0)(x) \\ &= \sum_{i=0}^{N-1} \sqrt{w_i(x)} \int_y G_i(|x-y|) \sqrt{w_i(y)} m_0(y)\,\mathrm{d}y\,, \end{aligned} \quad \text{(A.2)}$$

whose spatial smoothness is enough to guarantee the deformation to be diffeomorphic. Due to the convolution of the pre-weights, the vector field v_0 has a bounded norm in the space of C^1 vector fields which implies that its flow is a diffeomorphism at every time. In fact, we have:

Proposition 1. *The minimization of the objective functional* (A.1) *over a collection of image pairs provides diffeomorphic deformations for every pair of images. At every stage of the optimization procedure, the deformations are guaranteed to be diffeomorphic.*

[6]We enforce this small positive constant by clamping the pre-weights to $[\epsilon, 1]$. One could also directly integrate this into the weighted linear softmax definition by clamping to $[\epsilon, 1]$ instead of $[0, 1]$.

Proof. We have the existence of a constant K such that

$$\|f\|_{C^1} \leq K \|f\|_{H_i} \leq K \|f\|_{H_N}\,, \quad \text{(A.3)}$$

for every $f \in H_N$.

Denote by $\Phi : (I, m) \mapsto \omega$ the nonlinear map learnt by the neural network. At every step of the optimization, and at convergence (for a finite sample of pairs of images, each pair is denoted by the index j), the functional (A.1) is finite, which implies that $\Phi(I_j, m_j)$ is pointwisely bounded on the domain and is in TV, therefore, $G_{\text{small}} \star w_i$ has a bounded C^1 norm, as well as $\sqrt{w_i}$ since $w_i > \varepsilon_i > 0$. In addition, $E_j = \langle m_j, K(w) m_j \rangle$ is also finite and gives an upper bound for $\|G_N \star (w_i m_j)\|_{H_N}$. Thus, we have

$$\begin{aligned} \| \sum_{i=0}^{N-1} \sqrt{w_i(x)} G_i(|x-y|) \sqrt{w_i(y)} \star m_j \|_{C^1} \\ \leq KN \sup_{i=1,\ldots,N} (\|\sqrt{w_i}\|_{C^1} \|\sqrt{w_i} m_j\|_{H_N})\,. \end{aligned} \quad \text{(A.4)}$$

Therefore, the norm of the velocity field $v(x) = \sqrt{w_i(x)} G_i(|x-y|) \star \sqrt{w_i} m_j$ is bounded in C^1 and its flow is a diffeomorphism. □

Also, there is a corresponding variational derivation of the spatially varying kernel with the square root which is presented next.

A.1.1 Variational derivation

Let us detail the variational definition of the spatially varying kernel used in Equation (A.2). Consider

$$\|v\|_H^2 = \inf \left\{ \sum_{i=0}^{N-1} \|v_i\|_{H_i}^2 \,\middle|\, \sum_{i=0}^{N-1} \sqrt{w_i} v_i = v \right\}. \quad \text{(A.5)}$$

Using Lagrange multipliers, we get critical points of the functional

$$\sum_{i=0}^{N-1} \frac{1}{2} \|v_i\|_{H_i}^2 + \langle p, \sum_{i=0}^{N-1} \sqrt{w_i} v_i - v \rangle\,, \quad \text{(A.6)}$$

therefore we get

$$L_i v_i + w_i p = 0 \quad \forall i = 0, \ldots, N-1\,, \quad \text{(A.7)}$$

where L_i is the inverse of the kernel G_i. Hence, there exists p such that

$$\|v\|_H^2 = \sum_{i=0}^{N-1} \langle G_i \sqrt{w_i} p, \sqrt{w_i} p \rangle$$

for the norm. Moreover, since $v_i = G_i \sqrt{w_i} p$, we have

$$v = \sum_{i=0}^{N-1} \sqrt{w_i} G_i (\sqrt{w_i} p)\,. \quad \text{(A.8)}$$

B. Implementation details

CNN initialization/penalty. Directly using the CNN as described in §3.1.1 does, in our experience, not lead to stable estimation results for the weights. Proper initialization and penalizing undesirable weights is therefore essential. Specifically, we use the following approaches:

1) *Initialization:* We initialize all bias terms to zero and use the initialization scheme from [19] for the convolutional weights. For the last batch normalization layer we initialize the slope to a small value (0.025) to avoid massive weight changes at the beginning as the registration is very sensitive to such changes.

2) *Weighted linear softmax input penalty:* As the weighted linear softmax function clamps inputs, values within the clamping range will no longer produce gradients. In our experiments this was a highly problematic behavior as it appeared to lead to cases where one could not easily recover from poor locations in the input space to the weighted linear softmax[7]. Hence, we penalize the inputs when they are outside the $[0, 1]$ range as follows:

$$\mathrm{rp}(z) = \sum_{i=0}^{N-1} \left(w_i + z_i - \overline{z} - \mathrm{clamp}_{\epsilon,1}(w_i + z_i - \overline{z})\right)^2 . \tag{B.1}$$

Here, $\mathrm{clamp}_{\epsilon,1}$ clamps values to the interval $[\epsilon, 1]$. An $\epsilon > 0$ is required as the square root is not differentiable at zero. This penalty is integrated over all of space and added to the overall registration energy, *i.e.*,

$$\mathrm{RP}(z(x)) = \int \mathrm{rp}(z(x))\, \mathrm{d}x \ . \tag{B.2}$$

We did not experiment with weightings of this term and simply added it as is. In practice this appeared to be fine (but may warrant further investigation) as the term results in zero penalty when the input values to the weighted linear softmax are not clamped and it is operating in its linear regime.

3) *Weight decay:* We use a small weight decay [17] (set to 1e-5) applied to all the network weights. However, we did not extensively experiment with this parameter. Hence, its practical necessity is not clear to us at the moment. We added it to mitigate possible drift in the estimated parameters (*e.g.*, very large weights of the convolutional filters).

[7] Similarly, if one uses a standard softmax function then exponential terms may result in very small gradients.

C. Generation of synthetic data

To be able to validate with respect to a known ground truth we construct synthetic data as follows:

1) We generate concentric circular regions with random radii and associate different multi-Gaussian weights to these regions. We associate a fixed multi-Gaussian weight to the background.

2) We randomly create vector momenta at the borders of the concentric circles. Specifically, we randomly create 10 different sectors and, within each sector, we randomly create either all positive or negative momenta orthogonal to the circle boundaries. These momenta are smoothed afterwards.

3) Based on 2), we create a deformation.

4) We randomly create a noisy image of the same dimension as the image of the concentric circles and smooth it. We add this smoothed noise image to the concentric circle image and deform it and its associated weights given the deformation from 3). The resulting image is our synthetic source image. We also transform the image without noise.

5) We repeat steps 2) to 4), starting from the synthetic source image without noise. The resulting deformation is applied to the (noisy) synthetic source image to create the synthetic target image.

These steps are repeated to obtain a desired set of image pairs.