Progressively Trained Convolutional Neural Networks for Deformable Image Registration

Koen A. J. Eppenhof, Maxime W. Lafarge, Mitko Veta, and Josien P. W. Pluim, *Fellow, IEEE*

***Abstract*— Deep learning-based methods for deformable image registration are attractive alternatives to conventional registration methods because of their short registration times. However, these methods often fail to estimate larger displacements in complex deformation fields, for which a multi-resolution strategy is required. In this article, we propose to train neural networks progressively to address this problem. Instead of training a large convolutional neural network on the registration task all at once, we initially train smaller versions of the network on lower resolution versions of the images and deformation fields. During training, we progressively expand the network with additional layers that are trained on higher resolution data. We show that this way of training allows a network to learn larger displacements without sacrificing registration accuracy and that the resulting network is less sensitive to large misregistrations compared to training the full network all at once. We generate a large number of ground truth example data by applying random synthetic transformations to a training set of images, and test the network on the problem of intrapatient lung CT registration. We analyze the learned representations in the progressively growing network to assess how the progressive learning strategy influences training. Finally, we show that a progressive training procedure leads to improved registration accuracy when learning large and complex deformations.**

***Index Terms*— Deformable image registration, progressive training, convolutional neural networks, machine learning, lung registration.**

I. Introduction

THE fast runtimes of convolutional neural networks (CNNs) have made them an attractive new approach to deformable image registration. Whereas conventional deformable registration methods are based on computationally intensive iterative optimization, convolutional neural networks can estimate full-size deformation fields in one forward pass through the network. By leveraging the parallel computing capabilities of modern graphics processing units (GPUs), this can be done in seconds instead of the many minutes conventional methods take for complex deformable image registration problems.

Deep learning-based registration methods estimate the deformation between two input images directly using a CNN. The output can be a small number of registration parameters, or a full deformation field. These methods can be broadly divided into three groups based on how the network is trained to obtain the registration parameters.

Unsupervised methods are most similar to conventional registration methods: they train the network based on a similarity metric. During training, the learned deformation field is applied to the moving image by a spatial transformer network (STN, [1]) to compute the similarity with the fixed image. A loss function based on this similarity is back-propagated through the STN. Advantages of these methods are that there is no ground truth required, but this comes at the cost of using image similarity as a surrogate for registration error. Image similarity metrics can have multiple strong local optima, that do not necessarily coincide with a correct registration [2]. Examples of unsupervised registration include [3]–[9].

Weakly supervised methods are a variation on unsupervised methods used for multi-modal image registration. The networks are trained to optimize an auxiliary task that does not suffer from the difference in modalities, for example alignment of segmentations or corresponding landmarks. The most notable examples are [10] and [11], in which the training of a network is supervised by maximizing the overlap of manually annotated tissue segmentations. The network estimates a deformation field from two input images, that is used to deform the segmentation of the moving image and backpropagates the Dice coefficient between it and the segmentation of the fixed image. Disadvantages include that segmentations are required, and that the deformation fields inside the segmented areas are not guaranteed to be realistic.

Supervised methods are trained on examples of transformed images and the associated transformation. Instead of optimizing image similarity, these methods minimize the registration error explicitly. That means that no similarity metric is required. However, the disadvantage of these methods is that a ground truth of images and transformations needs to be constructed. One way is by generating deformation fields using existing registration algorithms. However, this can be negatively impacted by registration errors made by the

Manuscript received October 15, 2019; accepted November 9, 2019. Date of publication November 15, 2019; date of current version April 30, 2020. *(Corresponding author: Koen A. J. Eppenhof.)*

K. A. J. Eppenhof, M. W. Lafarge, and M. Veta are with the Department of Biomedical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: k.a.j.eppenhof@tue.nl).

J. P. W. Pluim is with the Department of Biomedical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands, and also with the Image Sciences Institute, University Medical Center Utrecht, 3508 GA Utrecht, The Netherlands.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2019.2953788

registration algorithms. An alternative is to generate synthetic ground truths. Because limited ground truth data for registration is available, these methods often use elaborate data augmentation techniques to generate large amounts of training data. Examples of supervised methods include [12]–[18].

Recent papers have applied deep learning to a broad spectrum of deformable registration applications, including brain MRIs [4], [6], [13], [17], [19], cardiac MRIs [5], [14], prostate MRIs and ultrasound images [10], and chest CTs [3], [5], [7], [8], [11], [15], [18].

A. Multi-Resolution Registration Problems

Virtually all proposed deep learning-based algorithms for deformable registration struggle with deformation fields that consist of a combination of large global and smaller local displacements. A large class of registration problems feature this combination. For example, in lung registration a global transformation is required to compensate for the large breathing motion, while smaller local transformations are required to match individual pulmonary blood vessels.

Deep learning methods cope with these problems in two ways: 1. by pre-aligning the images using a translational, rigid, or affine transformation before estimating the deformable part of the transformation using a CNN [4], [6], [8], [13]–[15], [17]–[19]; 2. using multiple networks to estimate partial transformations that need to be concatenated in order to obtain the full transformation [3], [5], [15].

Both approaches have considerable disadvantages. First, using conventional methods for pre-alignment defeats the purpose of using deep learning, as even a simple affine pre-registration can take tens of seconds or even minutes to complete, which is much longer than the sub-second evaluation times of CNNs that estimate the deformable part of the transformation. Secondly, combining multiple networks to estimate parts of the transformation means that multiple interpolation steps are required: the transformation estimated by the first network needs to be applied to the moving image before it can serve as an input for the second network. Using multiple networks in sequence will result in an accumulation of interpolation artifacts, which is likely to affect the quality of the deformation field.

Conventional registration methods solve the problem of having combinations of large global and small local displacements by using multi-resolution methods. These methods first optimize a simpler, more global deformation from lower resolution images. This deformation is then used as initialization in the next step, when higher resolution images are used to learn a more localized transformation. A number of these resolutions steps are then used to obtain the deformation field from coarse to fine scales [20].

B. Progressive Multi-Resolution Learning

In this paper, we apply the same rationale to supervised image registration. We start the training process with low resolution images as input to the network to learn lower resolution, and thus more global, deformation fields. Once the network has been optimized for this sub-problem, it is extended. More layers are added to learn higher resolution deformation fields from higher resolution images. Throughout this paper, we call this 'progressive training'. The methodology is based on Karras et al. who first used this technique to train generative adversarial networks (GANs) to generate high-resolution images of non-existing celebrities [21]. They found that it was difficult for GANs to generate high-resolution images and showed that by building the network layer-by-layer during training it *was* possible to learn realistic faces. This methodology is in line with so-called *curriculum learning* methods that start learning on smaller simpler problems before moving on to more difficult ones [22], [23].

We have reported on how to use this methodology for image registration previously, and showed that this method leads to more accurate registration compared to training the network all at once [24]. In this paper, we improve the method substantially, and give a more detailed evaluation and analysis of the network itself.

C. Lung Registration

We demonstrate the concept of progressive training on a network for lung registration. Public availability of expert-annotated landmarks for the lung registration problem in CT images (e.g. [25]–[28]) allowed us to objectively compare methods.

The registration of lung images is difficult. The sliding motion of the lungs relative to the ribs significantly complicates the registration. Furthermore, when registering intrapatient inhale to exhale images (or vice versa), the displacements that need to be modeled are relatively large, while also requiring a very fine-grained deformation field to register individual pulmonary blood vessels. It is this combination of large and small deformations, as well as global and local deformations, that make this problem difficult. Although relatively accurate deep learning based algorithms have been proposed for this problem, most often they require a cascade of networks that operate at separate scales or a pre-alignment step (translational, rigid, or affine) that relies on conventional registration methods [3], [5], [8], [15], [18]. In this paper, we aim to train one network that can both estimate global and local deformations in one forward pass through the network, by training the network progressively. This is an extension to our previous work in which we used affine pre-registration to solve the lung registration problem [18]. The contributions of this paper are three-fold:

- We propose a progressive learning scheme to enable training on large and small transformations within the same convolutional neural network.
- We show that it is possible to perform lung registration using a convolutional neural network without any pre-registration, resulting in very fast registration times.
- We show that a neural network can be trained in a supervised way on synthetically transformed images with large deformations, and that this neural network subsequently is able to generalize to a real registration problem.

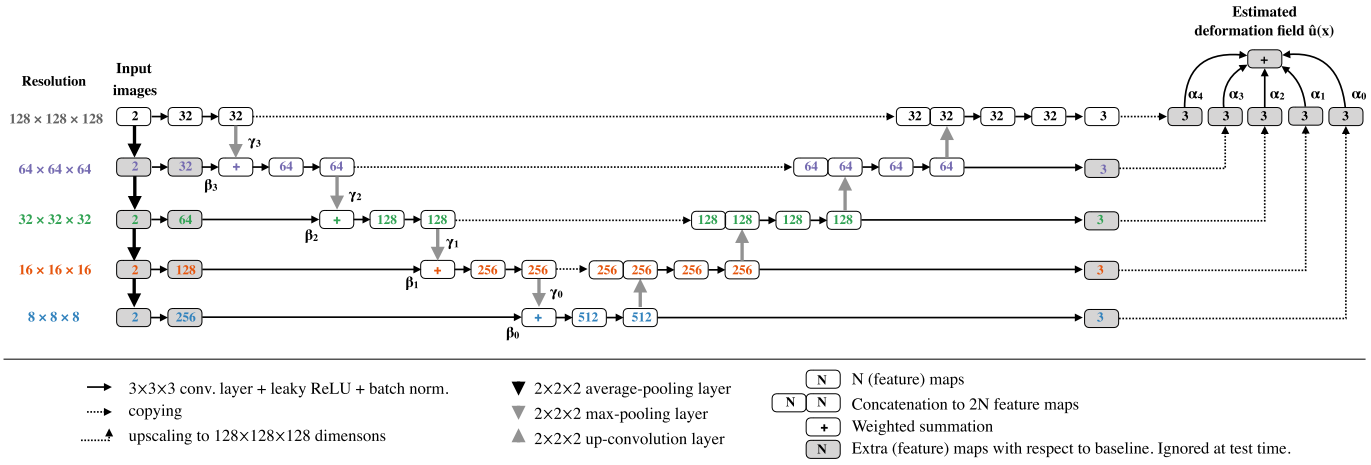

Fig. 1. The progressive learning architecture. The gray blocks indicate feature maps that are learned during training but ignored at test time. Compared to a normal U-net architecture we add input layers on the left side for every resolution level, each followed by an extra convolutional layer that matches the number of feature maps in that level, a summation node that sums the output of these convolutional layers and the output of the pooling layer in the level above it, and output maps at every level, which are summed up weighted by $\boldsymbol{\alpha}$ to obtain the final deformation field.

II. Method

We formally define the registration problem as the estimation of a vector field $\mathbf{u} : \Omega_F \rightarrow \mathbb{R}^d$ that aligns a moving image I_M with a fixed image I_F, i.e. for a point $\mathbf{x}$ in the domain Ω_F of the fixed image I_F, the associated voxel in image I_M can be found at position $\mathbf{T}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$. The networks we propose estimate the displacement field $\mathbf{u}$ for every voxel in the fixed domain Ω_F.

A. Baseline Network Architecture

We use a variant of the standard 3D U-net first introduced by Çiçek et al. [29]. We make four changes to this architecture: we change the number of channels in the input image to two such that it can contain the fixed and moving image; we change the number of output channels to three, such that the output of the network can represent the x, y, and z components of the vectors in the displacement field; we halve the number of channels in each convolutional layer to reduce the memory footprint of the network; and use the Leaky ReLU activation function with leakiness 10^{-2} instead of standard ReLU to prevent vanishing gradients during training [30]. Due to GPU memory limitations, we use a batch size of one, and use the accumulation of moving averages of the batch normalization parameters instead of computing them from one batch [31]. Throughout the remainder of the paper we refer to this architecture as the 'baseline architecture'.

B. Progressive Network Architecture

From the baseline architecture, we create the 'progressive architecture' by adding a number of additional layers. It is important to note that these extra layers are *only* required for training, and while the weights in these layers *are* optimized, they are ultimately not used at test time.

The baseline U-net architecture consists of five resolution levels: parts of the network in which the input and output feature maps have a certain resolution. In our network, the top resolution level outputs maps of $128 \times 128 \times 128$ voxels. The level below outputs $64 \times 64 \times 64$ voxel maps, and so on, each level halving the resolution along each axis. The bottom level thus uses $8 \times 8 \times 8$ voxel maps. The progressive training scheme builds up the network level-by-level, starting with the $8 \times 8 \times 8$ level. This level is trained on images and deformation fields that have been downsized to the same resolution. Once a level has converged, the next higher level is introduced in the training process.

Training the network in this way requires input and output layers at each level. Therefore, we add a convolutional layer at the right side of each level that outputs the three maps for the deformation field. Similarly, to match the number of feature maps at the left side of the level, we require an extra convolutional layer that maps the two input images to the matching number of feature maps. The output of that layer is added to the output of the pooling layer of the resolution level above it. The resulting architecture is shown in Figure 1.

Once a level has been trained, and the training has moved to the next level, the two extra convolutional layers on either side of the resolution level are redundant. Once the full network has been trained, this holds for all eight convolutional layers that have been added with respect to the baseline architecture, and the architecture is effectively equal to the baseline.

C. Smooth Transition Schedules

Like in the work by Karras et al., the introduction of new layers into the architecture is gradual, by slowly increasing the weights of the output of the new layer, and slowly decreasing the output of the layer below it [21]. Without such a transition, the introduction of a new level will be so disruptive that the gradient of the loss function will explode. Each level i is assigned a weight $\alpha_i \in [0, 1]$ that determines how much the output contributes to the deformation field. The output deformation fields $\mathbf{u}_i(\mathbf{x})$ of each level are scaled to the initial $128 \times 128 \times 128$ dimensions, and summed, weighted by weights $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4]$, such that the total deformation field

is given by

$$\mathbf{u}(\mathbf{x}) = \sum_{i=0}^{4} \alpha_i \mathbf{u}_i(\mathbf{x}), \ \mathbf{x} \in \Omega_F \tag{1}$$

Throughout training, the weights α_i sum to one, and at any time one or two weights are non-zero. The weights change according to a schedule that determines which levels are active. We chose trapezoidal functions for these schedules, defined by

$$\alpha_i(t) = \begin{cases} 0 & \text{for} \quad t < \tau_i \quad \text{and} \quad t > \tau_i + \Delta \\ \dfrac{t - \tau_i}{\delta} & \text{for} \quad \tau_i \le t \le \tau_i + \delta \\ 1 & \text{for} \quad \tau_i + \delta \le t \le \tau_i + \Delta - \delta \\ \dfrac{\tau_i + \Delta - t}{\delta} & \text{for} \quad \tau_i + \Delta - \delta \le t \le \tau_i + \Delta. \end{cases} \tag{2}$$

where t is the training iteration, τ_i is the start of training for a particular level i, Δ is the length of the time period in which the level is trained, and δ is the duration of the transition at the start and end of this period. For this paper, δ was set to 2000 iterations, Δ to 6000 iterations, and $\tau_i = -\frac{1}{2}\Delta + (\Delta - \delta)i$. The schedules are shown in Figure 5A. For the final resolution level, the weight stays equal to 1 for $t > \tau_i + \delta$. Therefore, at the start of training the weights are $\boldsymbol{\alpha} = [1, 0, 0, 0, 0]$. At the end of training, the weights are $\boldsymbol{\alpha} = [0, 0, 0, 0, 1]$, which also describes the baseline network.

In addition to the weights α_i, we introduce weights on the summations on the left side of each resolution level. These weights β_i and γ_i aid the smooth transition by weighting the output of the additional convolutional layer and the output of the pooling layer, slowly increasing the weight on the pooling layer, and decreasing the weight on the convolutional layer. These weights are dependent on α_i and defined as

$$\beta_i = \sum_{j=0}^{i} \alpha_j \quad \text{and} \quad \gamma_i = 1 - \beta_i. \tag{3}$$

When only the $8 \times 8 \times 8$ level is active (i.e. $\boldsymbol{\alpha} = [1, 0, 0, 0, 0]$), only the output of the extra convolutional layer at that level is active ($\beta_0 = 1$), and the output of the pooling layer at the $16 \times 16 \times 16$ level is ignored ($\gamma_0 = 0$). When the second level is active, the reverse is true. A further overview of the transitions between levels is shown in Figure 2.

D. Training Set

The network is trained on the 1,010 images of the LIDC-IDRI data set of lung CT images [32], [33]. These images have between 65 and 764 slices, and 512×512 slice dimensions. The slice thickness is between 0.60 and 5.00 mm, and the pixel size is between 0.46 and 0.98 mm. The scans were acquired on CT scanners from four different manufacturers.

E. Augmentation and Preprocessing

Like in [18], we create random deformations of the training set images. For each iteration of training we randomly select one of the images I and create two random transformations, $\mathbf{T}_1$ and $\mathbf{T}_2$ The purpose of the first transformation is to

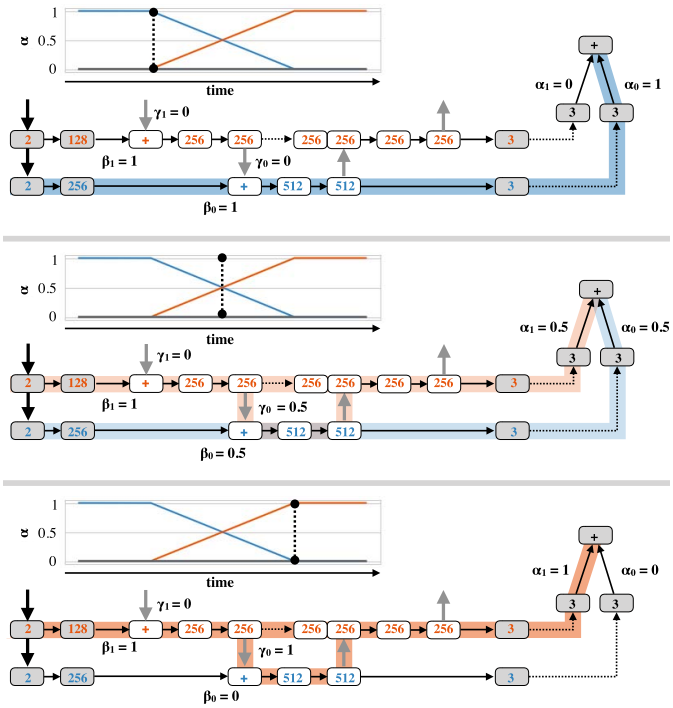

Fig. 2. Progressive learning with two levels. At the top the status of the network at the start of training is shown, where only the $8 \times 8 \times 8$ level is trained. In the middle the 50% transition between the first two levels, when both contribute equally to the final result, is shown. At the bottom the state after the transition is shown, when only the second level is contributing and the architecture is effectively a two-level U-net. The learning schedule is shown for each of the three states, with the dashed line indicating the depicted time point.

augment the data set. By using small deformable transformations for augmentation, we can create realistic variations of the images, in which the orientation and size of the lungs is maintained. The resulting image $I_1 = I(\mathbf{T}_1)$ is the equivalent of the moving image I_M in a registration problem. In registration, the fixed image I_F is similar to the moving image I_M, except for an extra transformation, which we simulate with $\mathbf{T}_2$. This requires a concatenation of both transformations, such that we obtain a second image $I_2 = I(\mathbf{T}_1 \circ \mathbf{T}_2)$. Note that the transformations are defined as displacements of voxel coordinates, and that therefore $\mathbf{T}_2$ is applied to those coordinates first. The second image is the equivalent of the fixed image. Given these two images, the registration task is to find $\mathbf{T}_2$.

The transformations $\mathbf{T}_1$ and $\mathbf{T}_2$ are created as random third-order B-spline transformations by sampling control point displacements from a uniform distribution. The grid sizes and ranges of the displacements are summarized in Table I. These ranges are constrained in such a way that the resulting transformations are smooth and do not fold. In fact, because of these constraints the transformations are diffeomorphic, which means that a concatenation of the transformations is diffeomorphic as well [34]. We interpolate the images using linear interpolation such that the resulting image is $128 \times 128 \times 128$ voxels. The gradient magnitude of the images I_1 and I_2 is used as input to the network. Using gradient images for image registration is common [35]. In lung registration it is especially valuable to focus the registration

TABLE I
B-SPLINE PARAMETERS FOR THE INDIVIDUAL TRANSFORMATIONS $\mathbf{T}_1 = \mathbf{t}_1$ AND $\mathbf{T}_2 = \mathbf{t}_{2,1} \circ \mathbf{t}_{2,2}$. THE RANGES OF THE MULTIVARIATE UNIFORM DISTRIBUTIONS ARE GIVEN FOR EACH DIRECTION

Transformation	Grid	Displacement ranges (voxels)		
		x	y	z
$\mathbf{t}_1$	$2 \times 2 \times 2$	$[-6.4, 6.4]$	$[-6.4, 6.4]$	$[-6.4, 6.4]$
$\mathbf{t}_{2,1}$	$4 \times 4 \times 4$	$[-3.2, 3.2]$	$[-6.4, 6.4]$	$[-12.8, 12.8]$
$\mathbf{t}_{2,2}$	$8 \times 8 \times 8$	$[-3.2, 3.2]$	$[-3.2, 3.2]$	$[-3.2, 3.2]$

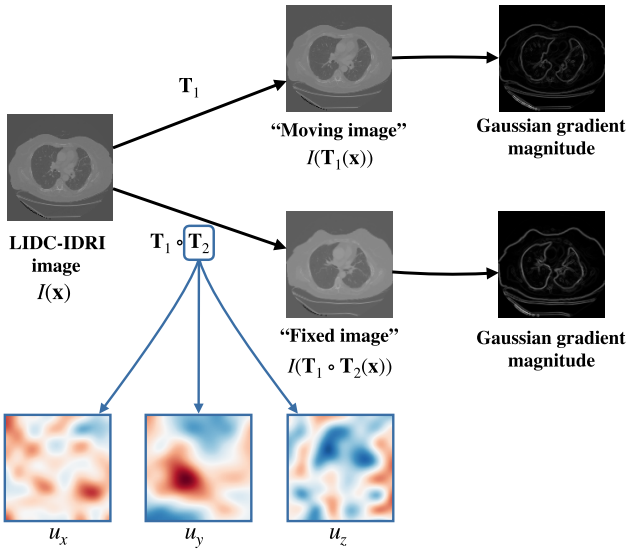

Fig. 3. The training set is constructed by applying augmentation transformation $\mathbf{T}_1$ and learned transformation $\mathbf{T}_2$ to images from the LIDC-IDRI data set. The learned displacement field $\mathbf{u}(\mathbf{x})$ is equal to $\mathbf{T}_2(\mathbf{x}) - \mathbf{x}$. The use of Gaussian gradient magnitude is problem specific, and not part of the core methodology.

on image edges, because intensities in the lungs will not be constant at corresponding points inside the lungs due to differences in density during breathing [8], [36]. We found that using gradient information gives substantially better registration results compared to training on I_1 and I_2 directly. We used Gaussian derivatives with $\sigma = 0.5$ voxels to compute the gradient magnitude. The full training set creation pipeline is summarized in Figure 3. The distributions of vectors in the learned deformation fields are shown in Figure 4. The implementation of the augmentation and learned transformations is publicly available.[1]

F. Training

The network is trained by minimizing the sum of squared errors of the estimated deformation field $\hat{\mathbf{u}}_t$ at iteration t with respect to the generated ground truth deformation field $\mathbf{u}$. We only calculate this loss for voxels inside a mask M. We use a single circular mask to mask out the CT scanner bore in all images. In the LIDC-IDRI images, the CT scanner bore results in a very strong circular edge in the gradient magnitude images. Because this edge should not be deformed

[1] https://www.github.com/tueimage/gryds

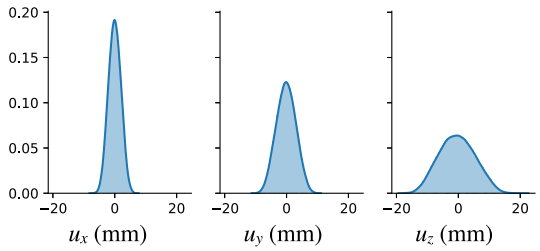

Fig. 4. Probability distributions of the displacements $\mathbf{u}_{2}{}_{\triangleright}\mathbf{x}_{\triangleleft} = \mathbf{T}_2(\mathbf{x}) - \mathbf{x}$ generated in the training set.

in the training set, we remove it from the images by applying a circular mask with a radius of 250 pixels centered at the center pixel of each 512×512 slice. This mask is transformed using $\mathbf{T}_1$ to force the network to only learn the deformation for non-zero pixels. The loss function then becomes

$$L(t) = \frac{\sum_{\mathbf{x} \in \Omega_F} M(\mathbf{T}_1(\mathbf{x})) \|\mathbf{u}(\mathbf{x}) - \hat{\mathbf{u}}_t(\mathbf{x})\|_2^2}{\sum_{\mathbf{x} \in \Omega_F} M(\mathbf{T}_1(\mathbf{x}))}. \quad (4)$$

The reason we opt for the L_2 norm here instead of the L_1-norm as we have used in prior work ([18]), is the significantly larger displacements in the training set, which are better minimized by a quadratic loss function. The loss function is minimized using stochastic gradient descent with a constant learning rate of 10^{-2} and a momentum of 0.5.

G. Independent Test Data Set

To validate the registration accuracy we use lung CT data from the public DIR-Lab data set [25], [26], CREATIS study [28], and POPI model [27]. These sets contain pairs of inhale/exhale images of the lungs, each accompanied with sets of expert-annotated corresponding landmarks that are located on anatomically distinctive points within the lung field [25], [26]. The DIR-Lab set contains ten pairs of images with 300 corresponding landmarks each; the CREATIS data set is comprised of five pairs of images with 100 corresponding landmarks each; and the POPI model consists of one image with 41 corresponding landmarks. The images were cropped around the lung field and resized to $128 \times 128 \times 128$ voxel dimensions, with voxel spacing ranging between $1.61 \times 1.16 \times 1.52$ mm^3 and $2.40 \times 1.84 \times 2.46$ mm^3.

III. EXPERIMENTS AND RESULTS

We train both the progressive and baseline network (without progressive learning) on the same data to make a fair comparison of both training strategies. The training occurs in parallel, such that the randomly generated training set (Section II-E) are used to train both networks. We also apply the *exact* same weight initialization (Glorot normally distributed initialization [37]) at the start of training for both networks, such that a bias in either network as a result of the initialization of the weights is impossible.

In this section, we perform experiments regarding the optimization of the networks, the registration accuracy during training, the deformation fields generated by each resolution level, and quantitative registration results. The networks are evaluated on the DIR-Lab, CREATIS, and POPI data sets described in Section II-G.

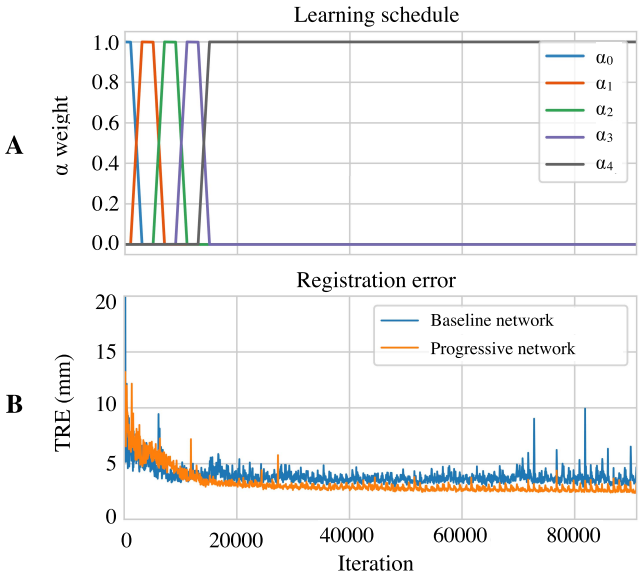

Fig. 5. **A**. The learning schedule for the progressive network. **B**. The registration error as function of the iteration.

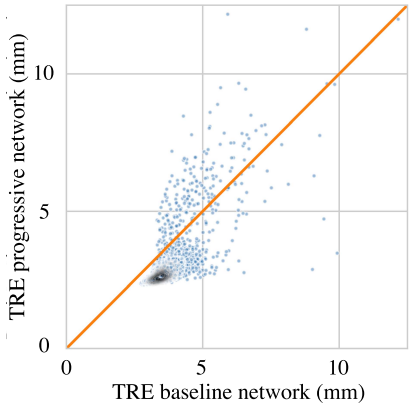

Fig. 6. Registration error (TRE) of progressive network plotted against the registration error of the baseline network throughout training, plotted for each snapshot of the networks' weights. The kernel density estimation of the points is indicated as a gray heat map, and shows that for the majority of the snapshots the progressively trained network results in better TREs compared to the baseline method.

A. Registration Performance Throughout Training

Every fifty iterations of training, we save a snapshot of the network weights. This allows us to evaluate the registration performance throughout training on the test set (Figure 5B). We can inspect the average registration error evaluated on the landmarks by plotting the registration error of the progressive network against the registration error of the baseline network throughout the training process (Figure 6). This shows that the progressively trained network converges to a lower registration error than the conventionally trained baseline network.

B. Qualitative Results

In Figure 7 we show the predicted deformation fields for each of the resolution levels of the progressive network throughout training, evaluated on a random example created using the procedure explained in Section II-E. This visualization shows how the network learns resolution-by-resolution. In Figure 8, examples of registered images for both network architectures are shown, as well as the initial situation before registration.

C. Quantitative Results

We evaluate both versions of the network on the DIR-Lab, CREATIS, and POPI data sets, by measuring the target registration error (TRE) on the landmarks. For two corresponding landmarks $\mathbf{x} \in \Omega_F$ and $\mathbf{y} \in \Omega_M$, the TRE for the estimated transformation $\hat{\mathbf{T}}$ is defined as $\text{TRE}(\mathbf{x}, \mathbf{y}) = \|\hat{\mathbf{T}}(\mathbf{x}) - \mathbf{y}\|$. The results are shown in Table II. On average, the error made by the progressive network is 1.0 millimeter smaller compared to the baseline network. This result is also reflected by the correlation plots in Figure 9, that show the estimated displacement of each landmark in these data sets by the network against the true displacements. The significance of the difference between the TREs obtained by the progressive and baseline network was tested with a one-sided Wilcoxon signed-rank test. The progressive network TREs were found to be significantly smaller ($W = 153$, $p = 0.0003$). We also tested the amount of folding in the estimated deformation fields by measuring the percentage of voxels with negative Jacobian determinant within the field of view (i.e. everything within the CT scanner's bore). For the progressive network the amount of folding was on average $0.39 \pm 0.21\%$. For the baseline network it amounted to $0.18 \pm 0.11\%$. The percentages per image pair are displayed in Table II.

D. Comparison to Existing Methods for Lung Registration

The progressive network results in an average TRE of 2.37 ± 1.77 mm for the full validation set, and 2.43 ± 1.81 mm for the DIR-Lab set only. The DIR-Lab set has been used extensively to validate other registration methods in literature. This allows us to compare the performance to other existing methods (Table III). Conventional methods can reach registration errors of 1.3 mm on this task. An existing deep learning method that has been tested on DIR-Lab reached 2.64 ± 4.32 mm [5]. It is important to note that the deep learning methods are considerably faster, as the speed indications in Table III show.

E. Effects of Learning Schedule and Initialization

To test the effect of the choice of learning schedule, we have additionally tested different parameterizations of δ and Δ in the schedule in Equation (2). We show the average TRE for the DIR-Lab, POPI, and CREATIS datasets for each of these settings. All other settings, such as weight initialization and training set, are identical.

To test the effect of different weight initializations, we trained four additional baseline and progressive networks (with the standard $\delta = 2000, \Delta = 6000$ setting) starting from different random Glorot initializations [37]. The average TREs for the baseline networks were 4.20 ± 3.93 mm, 3.25 ± 3.51 mm, 3.44 ± 3.87 mm, and 3.98 ± 4.15 mm. For the progressive networks they were 2.61 ± 1.90, 2.27 ± 1.53 mm, 2.50 ± 1.78 mm, and 2.58 ± 1.84 mm. This indicates only small

TABLE II
AVERAGE TARGET REGISTRATION ERRORS (MM) EVALUATED ON LANDMARKS (MEAN ± STANDARD DEVIATION), AND PERCENTAGE OF IMAGE DOMAIN THAT SHOWS FOLDING (JACOBIAN DETERMINANT BELOW ZERO) FOR EACH OF THE IMAGE PAIRS IN THE TEST SET

			Before	Baseline network		Progressive network	
Data set	**Image pair**	**Landmarks**	**TRE (mm)**	**TRE (mm)**	**Folding**	**TRE (mm)**	**Folding**
DIR-Lab	1	300	3.89 ± 2.78	2.16 ± 2.12	0.33%	1.65 ± 0.79	0.83%
	2	300	4.34 ± 3.90	1.65 ± 0.89	0.24%	1.52 ± 0.85	0.39%
	3	300	6.94 ± 4.05	1.91 ± 1.05	0.27%	1.77 ± 0.97	0.75%
	4	300	9.83 ± 4.85	2.70 ± 1.80	0.35%	2.20 ± 1.18	0.76%
	5	300	7.48 ± 5.50	2.42 ± 1.71	0.02%	2.28 ± 1.57	0.23%
	6	300	10.89 ± 6.96	4.15 ± 2.30	0.38%	3.17 ± 1.82	0.28%
	7	300	11.03 ± 7.42	4.53 ± 3.46	0.16%	2.82 ± 1.72	0.23%
	8	300	14.99 ± 9.00	10.26 ± 6.51	0.11%	4.10 ± 3.18	0.01%
	9	300	7.92 ± 3.97	3.57 ± 2.11	0.16%	2.70 ± 1.46	0.30%
	10	300	7.30 ± 6.34	2.17 ± 1.59	0.21%	2.06 ± 1.62	0.44%
CREATIS	1	100	5.75 ± 2.60	2.34 ± 1.34	0.16%	1.83 ± 0.84	0.43%
	2	100	13.98 ± 7.18	4.72 ± 2.88	0.06%	3.25 ± 2.34	0.33%
	3	100	7.69 ± 5.05	1.74 ± 0.86	0.09%	1.62 ± 0.75	0.19%
	4	100	7.34 ± 4.89	2.55 ± 1.82	0.18%	1.96 ± 1.16	0.43%
	5	107	7.13 ± 5.08	2.31 ± 1.13	0.08%	2.10 ± 1.52	0.47%
	6	113	6.67 ± 3.68	2.14 ± 1.29	0.02%	1.83 ± 1.45	0.18%
POPI	1	41	6.29 ± 3.13	2.63 ± 1.98	0.18%	2.44 ± 1.57	0.31%
	All	3,661	8.37 ± 6.40	3.39 ± 3.48	0.18 ± 0.11%	2.37 ± 1.77	0.39 ± 0.21%

TABLE III
COMPARISON WITH EXISTING METHODS ON THE DIR-LAB DATA SET. FOR EACH METHOD THE TRE (MEAN ± STANDARD DEVIATION) AND REGISTRATION TIME IS REPORTED AS AN INDICATION. ⋆ INDICATE METHODS BASED ON DEEP LEARNING

Method	**Mean TRE (mm)**	**Algorithm duration**
Vandemeulebroucke et al. (2011) [28]	1.95 ± 1.47	N/A
Schmidt-Richberg et al. (2013) [38]	2.13 ± 1.82	N/A
Heinrich et al. (2013) [39]	1.43 ± 1.3	7.97 min.
Delmon et al. (2013) [40]	1.66 ± 1.14	58 min.
Berendsen et al. (2014) [41]	1.36 ± 1.01	N/A
De Vos et al. (2019) [5]⋆	2.64 ± 4.32	1.08 ± 0.14 sec.
Proposed method ⋆	2.43 ± 1.81	0.56 ± 0.08 sec.

TABLE IV
AVERAGE TRE (MEAN ± STANDARD DEVIATION) FOR THE DIR-LAB, POPI, AND CREATIS DATASETS FOR DIFFERENT SETTINGS OF δ AND Δ IN EQUATION (2)

δ	Δ	**Effect**	**Mean TRE (mm)**
0	**0**	**Baseline**	**3.39 ± 3.48**
0	4000	No transitions (square)	None⋆
1000	3000	Short trapezoidal phases	2.63 ± 2.08
2000	4000	Only transitions (triangular)	2.50 ± 1.94
2000	**6000**	**Trapezoidal as proposed**	**2.37 ± 1.77**
4000	12000	Long trapezoidal phases	2.69 ± 1.83

⋆) This network crashes because of the lack of transition.

changes between initializations and a systematic improvement for the progressive network over the baseline network.

IV. DISCUSSION

A. Experimental Results

From the comparison between both tested networks we can conclude that progressive learning significantly improves the registration error with respect to regular training by 1.0 millimeter, and that the progressive training leads to better correlation with the true landmark displacements of the tested pulmonary CT image pairs. From the TRE as measured during training as shown in Figure 5A, we can conclude that the progressively trained network is more stable with regards to the TRE, and from Figure 5B we see that the progressively trained network converges to a lower registration error. Figure 8 shows that the progressively trained architecture performs better than the baseline architecture for larger initial misregistration, which is especially apparent for DIR-Lab case 8, for which the pre-registration error is the highest of any of the images. For this image, the baseline architecture fails to register the top part of the lungs, which the progressive network is able to register correctly. The figure also shows that most of the residual misregistration occurs in homogeneous areas with few anatomical features, which suggests that the network requires these features to estimate the local deformation. The networks are able to estimate deformation vector fields with very limited folding without strong constraints on the predicted deformation fields. Because the transformations in the training set are diffeomorphic by construction and therefore display no folding, we hypothesize that the network only learns to generate transformations with little folding at test time as well. The amount of folding caused by the progressive network is larger than that of the baseline network, which is likely caused by the fact that the transformations predicted by the progressive network are more complex, which means they are more accurate but also create more opportunity for folding.

Figure 7 gives insight into how the network learns to estimate a particular deformation field. Because the progressively trained network has to learn the deformation field using the $8 \times 8 \times 8$ resolution level initially, the network can learn a coarse version of the deformation field quickly, and then improve on it using the higher resolution levels. From these

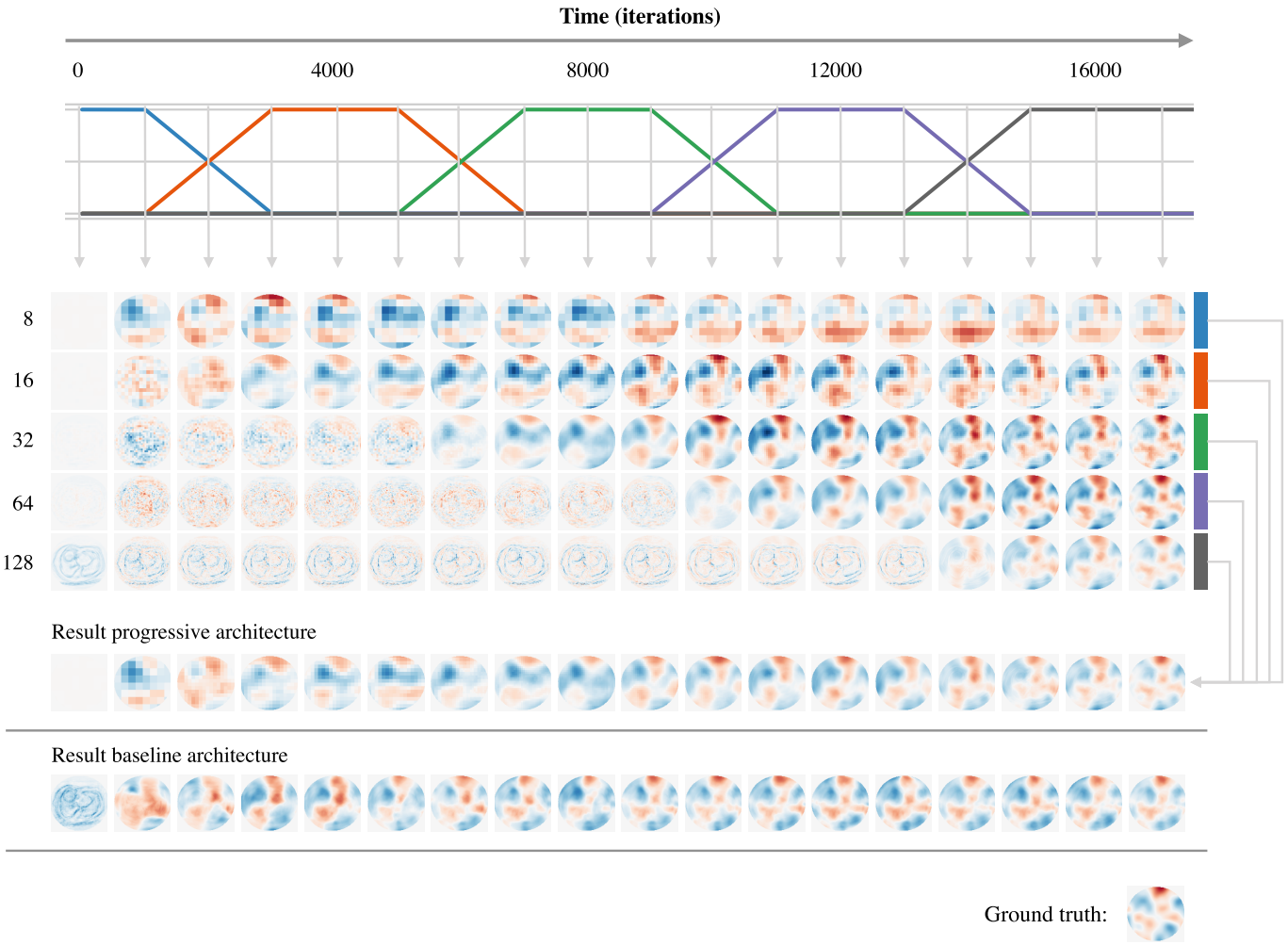

Fig. 7. Example of output maps for $\hat{u}_x$ at each resolution level throughout training for a synthetic deformation field. The resultant deformation field from the progressively trained network and the deformation field estimated by the baseline network are shown below. Note that when lower resolution levels are not supervised anymore, they can still be further optimized through back-propagation. The noisy output in the fields that have not been optimized yet are a result of a propagation of outputs from the lower resolution levels.

maps we can see that the deformation fields at the lower resolution levels are improved upon at later stages of training as well. The reason the lower resolution layers improve in later stages is that during the transition between levels, the network becomes dependent on the lower level when training the newly added level. This level can then improve upon the previous level's estimate of the deformation. After the addition of the final level we train longer for convergence, but the gains in accuracy are relatively marginal.

B. Learning Schedule

In principle the choice of progressive learning schedule is free. One limitation we put on the schedule is that the weights α_i sum up to one. This also allows other schedules, such as more sigmoidal functions. We have experimented with these but have found no advantages compared to the trapezoidal functions. Of course, it is possible to use longer or shorter schedules. With trapezoidal schedules we found that if each level is allowed to train for a minimum of 1000 iterations before transitioning to the next level, the progressive learning improves the registration accuracy over the baseline network. Importantly, when the transitions are omitted, i.e. when $\delta = 0$, the training loss explodes when the second level is introduced, showing that the smooth transitions are a necessity. We found that as long is δ is sufficiently large, the progressive network is better than the baseline, and the difference between the results of different settings is relatively small.

C. Application to Lung CT Images

The proposed network is specifically trained for pulmonary CT registration, but multi-resolution strategies are common in many image registration problems. For these problems, deep learning-based alternatives could benefit from progressive learning. In this paper, we use the Gaussian gradient magnitude of the images as input to the network. It may seem surprising that the edges of the images contain enough information in the lower resolution versions that the progressive training starts with for estimation of the deformation field. However, we can observe from Figure 5 that this is the case, as the network is able to reduce the registration error in the first iterations of

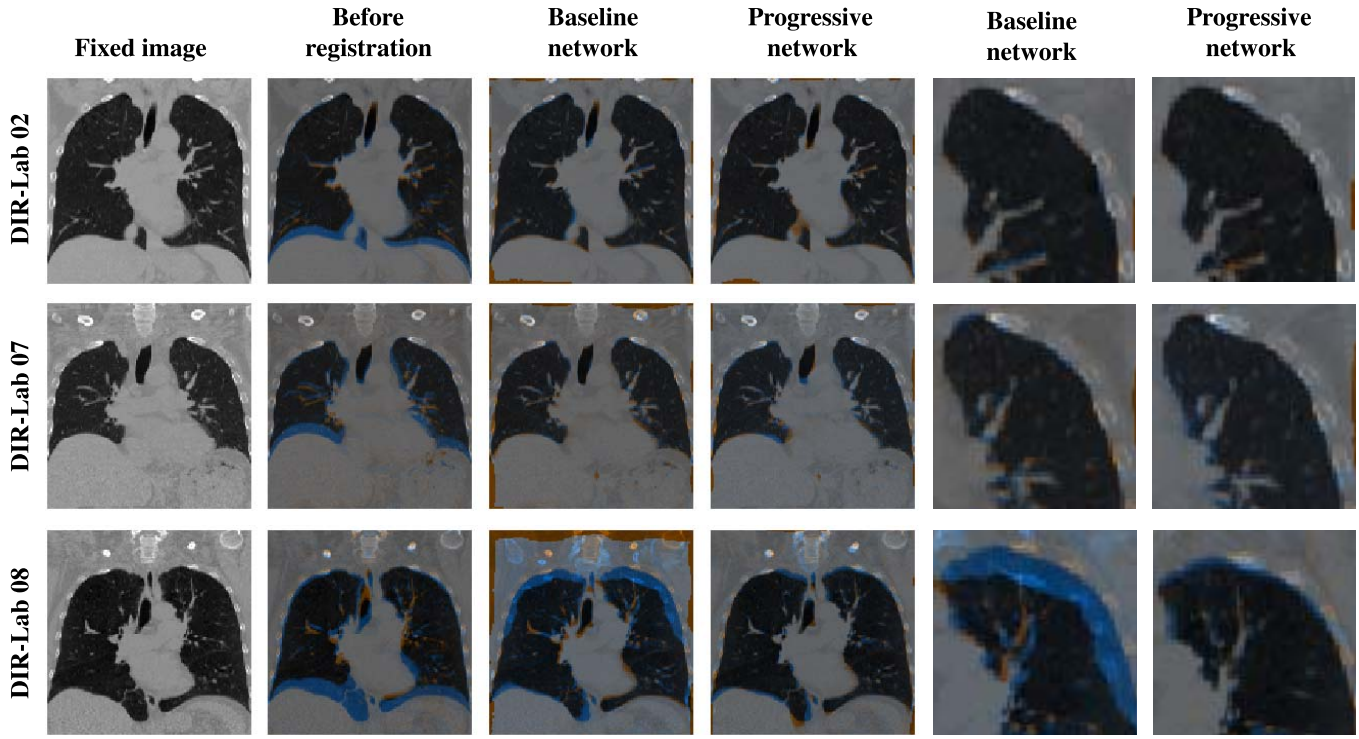

Fig. 8. Examples of registration in DIR-Lab images 2 (lowest initial misregistration), 7, and 8 (highest initial misregistration). The fixed images are shown in orange, while the (transformed) moving images are shown in blue. The rightmost two columns show zoomed-in versions of the center two columns.

training from these images. In addition, in contrast with our previous work ([18]), the current method does not require lung masks for the fixed image: a mask of the CT scanner bore is sufficient, which will be equal for every image made with a particular scanner.

D. Limitations

The current implementation of the network has a $128 \times 128 \times 128$ input size. Although these dimensions are convenient for the resolution levels in the U-net, it is possible to use different input dimensions as long as they are divisible by 2^{N-1} where N is the number of levels. This, and the fact that larger input sizes that conform to this will have a large footprint on the GPU memory, can be a limiting factor for other applications. For the lung CT images, we found the current input size a good compromise between simplicity, memory use, and image resolution.

The test data from the DIR-Lab, POPI, and CREATIS data sets comprise data from 17 patients. Pairs of medical images with manually labeled corresponding landmarks is the standard for validating deformable image registration, but unfortunately, this kind of data is in short supply. Although the number of test cases is therefore limited, the fact that there is a lot of variation among them in terms of population, scanner, and hospital, and the fact that we used a completely separate set of images for training, suggests that the network can generalize to new data very well.

Of course, in the current paper, specific choices were made regarding the architecture and learning schedule that can be changed to fit a new application. We have tested multiple trapezoidal schedules with different values for the parameters δ and Δ, which showed their setting only have a small effect. The small differences in results suggest that it may prove advantageous to find an optimal training schedule or even train each level for a level-specific number of iterations. Within the current application, it would be difficult to optimize so many hyperparameters, given the limited amount of test data.

E. Advantages of the Proposed Method

Because the network architecture is no different after training than the network we proposed in previous work ([18]), the network is equally fast, with registration speeds of 0.55 seconds for images of $128 \times 128 \times 128$ dimensions. This is much faster than conventional registration methods for pulmonary CT registration, for which indications of runtime are shown in Table III. The errors that are obtained on the DIR-Lab data set are substantially better than previous end-to-end networks we have used [42]. There is some room for improvement when we compare to conventional methods, which can reach errors of about one millimeter at considerably longer runtimes [18].

F. Further Possibilities

From the proposed network, an obvious extension would be to create a progressively trained network that *combines* the estimated deformation fields at every level by concatenation, i.e. if we have transformations $\mathbf{T}_8$ through $\mathbf{T}_{128}$, we can train the network to learn a net transformation $\mathbf{T}_{net} = \mathbf{T}_8 \circ \mathbf{T}_{16} \circ \ldots \circ \mathbf{T}_{128}$ progressively. In that case, the training procedure would be similar to the proposed method, but

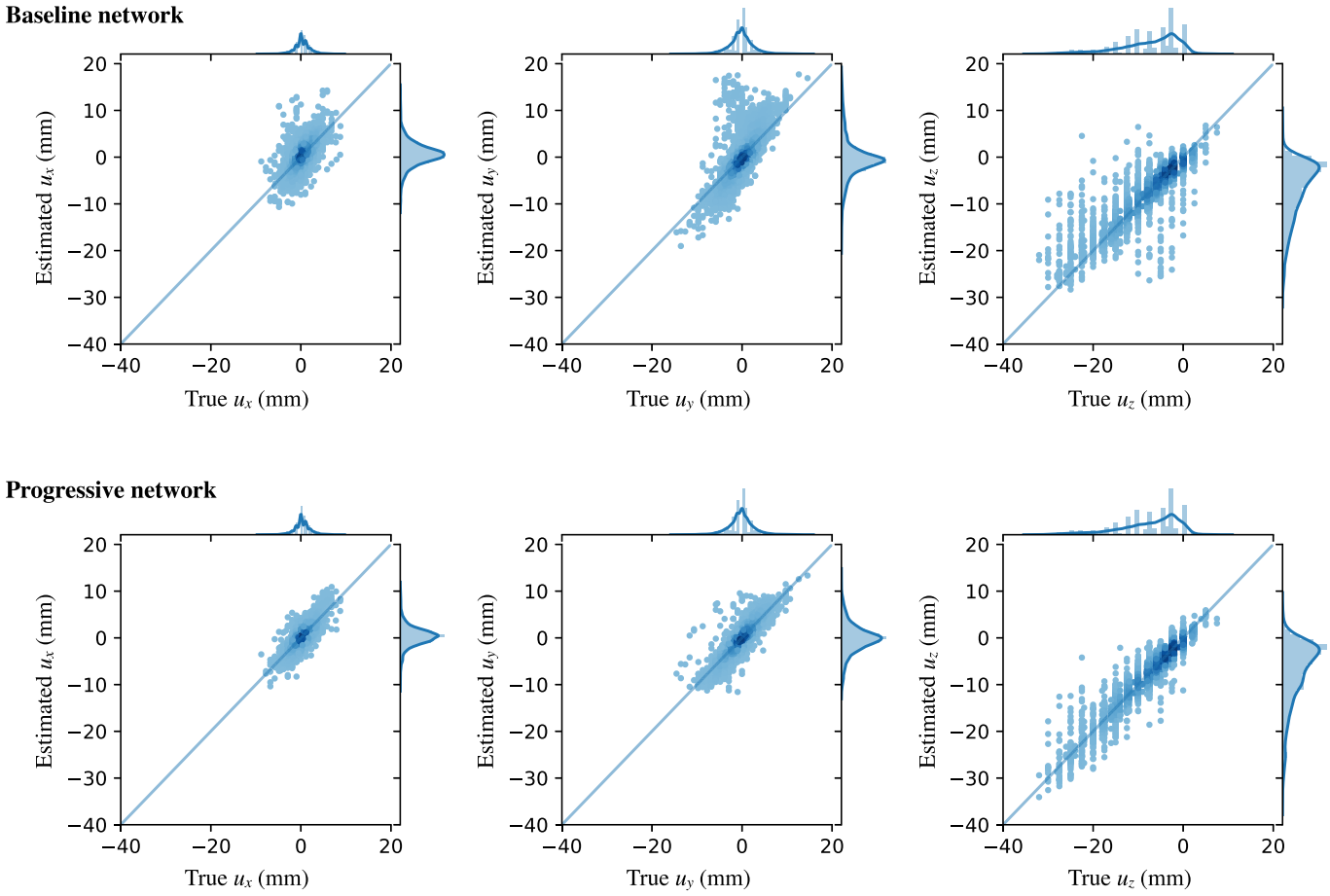

Fig. 9. Correlation plots for both networks. Each point is the displacement of a landmark in the DIR-Lab, CREATIS, and POPI data sets. From left to right, the *x*, *y*, and *z* components of the displacements are shown.

instead of directly summing the deformation fields according to a schedule, the concatenated transforms *up to* a certain resolution level are summed. As an example, in the first transition, the network would be trained to optimize

$$\hat{\mathbf{u}}(\mathbf{x}) = \alpha_0 \mathbf{u}_0(\mathbf{x}) + \alpha_1 \left(\mathbf{x} + \mathbf{u}_1(\mathbf{x} + \mathbf{u}_2(\mathbf{x}))\right). \tag{5}$$

We suggest that it is also possible to use progressive learning for unsupervised training of registration applications, by adding an STN at the end of the network, and using a conventional similarity metric to train the network.

V. Conclusion

We have proposed a progressively trained neural network for deformable image registration that can deal with deformations of multiple scales more accurately than conventionally trained networks. The proposed method has been applied to lung CT images, but can generally be applied to registration tasks in which large global and smaller local deformations need to be modeled, without sacrificing the fast registration times of CNNs.

References

[1] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. NIPS*, 2015, pp. 2017–2025.

[2] T. Rohlfing, "Image similarity and tissue overlaps as surrogates for image registration accuracy: Widely used but unreliable," *IEEE Trans. Med. Imag.*, vol. 31, no. 2, pp. 153–163, Feb. 2012.

[3] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum, "End-to-end unsupervised deformable image registration with a convolutional neural network," in *Proc. Int. Workshop Deep Learn. Med. Image Anal.*, 2017, pp. 204–212.

[4] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. V. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 9252–9260.

[5] B. D. de Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring, and I. Išgum, "A deep learning framework for unsupervised affine and deformable image registration," *Med. Image Anal.*, vol. 52, pp. 128–143, Feb. 2018.

[6] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning for fast probabilistic diffeomorphic registration," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, part 1, 2018, pp. 729–738.

[7] S. Christodoulidis *et al.*, "Linear and deformable image registration with 3D convolutional neural networks," in *Proc. Int. Workshop Reconstruction Anal. Moving Body Organs*, 2018, pp. 13–22.

[8] A. Hering and S. Heldmann, "Unsupervised learning for large motion thoracic CT follow-up registration," *Proc. SPIE*, vol. 10949, Mar. 2019, Art. no. 109491B.

[9] M. D. Ketcha *et al.*, "Effect of statistical mismatch between training and test images for CNN-based deformable registration," *Proc. SPIE*, vol. 10949, Mar. 2019, Art. no. 109490T.

[10] Y. Hu *et al.*, "Weakly-supervised convolutional neural networks for multimodal image registration," *Med. Image Anal.*, vol. 49, pp. 1–13, Oct. 2018.

[11] Y. Ha, L. H. Hansen, M. Wilms, and M. P. Heinrich, "Geometric deep learning and heatmap prediction for large deformation registration of abdominal and thoracic CT," in *Proc. Med. Imag. Deep Learn. (MIDL)*, 2019.

[12] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[13] X. Cao *et al.*, "Deformable image registration based on similarity-steered CNN regression," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, part 1, 2017, pp. 300–308.

[14] M.-M. Rohé, M. Datar, T. Heimann, M. Sermesant, and X. Pennec, "SVF-Net: Learning deformable image registration using shape matching," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, part 1, 2017, pp. 266–274.

[15] H. Sokooti, B. D. de Vos, F. F. Berendsen, B. P. F. Lelieveldt, I. Išgum, and M. Staring, "Nonrigid image registration using multi-scale 3D convolutional neural networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, part 1, 2017, pp. 232–239.

[16] K. A. J. Eppenhof and J. P. W. Pluim, "Error estimation of deformable image registration of pulmonary CT scans using convolutional neural networks," *J. Med. Imag.*, vol. 5, no. 2, May 2018, Art. no. 024003.

[17] X. Cao, J. Yang, J. Zhang, Q. Wang, P.-T. Yap, and D. Shen, "Deformable image registration using a cue-aware deep regression network," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 9, pp. 1900–1911, Sep. 2018.

[18] K. A. J. Eppenhof and J. P. W. Pluim, "Pulmonary CT registration through supervised learning with convolutional neural networks," *IEEE Trans. Med. Imag.*, vol. 38, no. 5, pp. 1097–1105, May 2019.

[19] X. Yang, R. Kwitt, M. Styner, and M. Niethammer, "Quicksilver: Fast predictive image registration–A deep learning approach," *NeuroImage*, vol. 158, pp. 378–396, Sep. 2017.

[20] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Med. Image Anal.*, vol. 2, no. 1, pp. 1–36, Mar. 1998.

[21] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–26.

[22] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71–99, Jul. 1993.

[23] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 41–48.

[24] K. A. J. Eppenhof, M. W. Lafarge, and J. P. W. Pluim, "Progressively growing convolutional networks for end-to-end deformable image registration," *Proc. SPIE*, vol. 10949, Mar. 2019, Art. no. 109491C.

[25] R. Castillo *et al.*, "A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets," *Phys. Med. Biol.*, vol. 54, no. 7, pp. 1849–1870, 2009.

[26] E. Castillo, R. Castillo, J. Martinez, M. Shenoy, and T. Guerrero, "Four-dimensional deformable image registration using trajectory modeling," *Phys. Med. Biol.*, vol. 55, no. 1, pp. 305–327, 2010.

[27] J. Vandemeulebroucke, D. Sarrut, and P. Clarysse, "The POPI-model, a point-validated pixel-based breathing thorax model," in *Proc. 15th Int. Conf. Comput. Radiat. Therapy (ICCR)*, 2007, pp. 1–8.

[28] J. Vandemeulebroucke, S. Rit, J. Kybic, P. Clarysse, and D. Sarrut, "Spatiotemporal motion estimation for respiratory-correlated imaging of the lungs," *Med. Phys.*, vol. 38, no. 1, pp. 166–178, 2011.

[29] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, part 2, 2016, pp. 424–432.

[30] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 1–6.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.

[32] K. Clark *et al.*, "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *J. Digit. Imag.*, vol. 26, no. 6, pp. 1045–1057, 2013.

[33] S. G. Armato, III., *et al.*, "The lung image database consortium (LIDC) and image database resource initiative (IDRI): A completed reference database of lung nodules on CT scans," *Med. Phys.*, vol. 38, no. 2, pp. 915–931, 2011.

[34] D. Rueckert, P. Aljabar, R. A. Heckemann, J. V. Hajnal, and A. Hammers, "Diffeomorphic registration using B-splines," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, part 2, 2006, pp. 702–709.

[35] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Image registration by maximization of combined mutual information and gradient information," *IEEE Trans. Med. Imag.*, vol. 19, no. 8, pp. 809–814, Aug. 2000.

[36] J. Rühaak, S. Heldmann, T. Kipshagen, and B. Fischer, "Highly accurate fast lung CT registration," *Proc. SPIE*, vol. 8669, Mar. 2013, Art. no. 86690Y.

[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 249–256.

[38] A. Schmidt-Richberg, R. Werner, H. Handels, and J. Ehrhardt, "Estimation of slipping organ motion by registration with direction-dependent regularization," *Med. Image Anal.*, vol. 16, no. 1, pp. 150–159, 2012.

[39] M. P. Heinrich, M. Jenkinson, M. Brady, and J. A. Schnabel, "MRF-based deformable registration and ventilation estimation of lung CT," *IEEE Trans. Med. Imag.*, vol. 32, no. 7, pp. 1239–1248, Jul. 2013.

[40] V. Delmon, S. Rit, R. Pinho, and D. Sarrut, "Registration of sliding objects using direction dependent B-splines decomposition," *Phys. Med. Biol.*, vol. 58, no. 5, pp. 1303–1314, 2013.

[41] F. F. Berendsen, A. N. T. J. Kotte, M. A. Viergever, and J. P. Pluim, "Registration of organs with sliding interfaces and changing topologies," *Proc. SPIE*, vol. 9034, Mar. 2014, Art. no. 90340E.

[42] K. A. J. Eppenhof, M. W. Lafarge, P. Moeskops, M. Veta, and J. P. W. Pluim, "Deformable image registration using convolutional neural networks," *Proc. SPIE*, vol. 10574, Mar. 2018, Art. no. 10574S.