Decoder-Only Image Registration

Xi Jia, Wenqi Lu, Xinxing Cheng, and Jinming Duan

***Abstract*— In unsupervised medical image registration, encoder-decoder architectures are widely used to predict dense, full-resolution displacement fields from paired images. Despite their popularity, we question the necessity of making both the encoder and decoder learnable. To address this, we propose LessNet, a simplified network architecture with only a learnable decoder, while completely omitting a learnable encoder. Instead, LessNet replaces the encoder with simple, handcrafted features, eliminating the need to optimize encoder parameters. This results in a compact, efficient, and decoder-only architecture for 3D medical image registration. We evaluate our decoder-only LessNet on five registration tasks: 1) inter-subject brain registration using the OASIS-1 dataset, 2) atlas-based brain registration using the IXI dataset, 3) cardiac ES-ED registration using the ACDC dataset, 4) inter-subject abdominal MR registration using the CHAOS dataset, and 5) multi-study, multi-site brain registration using images from 13 public datasets. Our results demonstrate that LessNet can effectively and efficiently learn both dense displacement and diffeomorphic deformation fields. Furthermore, our decoder-only LessNet can achieve comparable registration performance to benchmarking methods such as VoxelMorph and TransMorph, while requiring significantly fewer computational resources. Our code and pre-trained models are available at `https://github.com/xi-jia/LessNet`.**

***Index Terms*— Decoder-Only, Image Registration, U-Net, Efficient, Diffeomorphic**

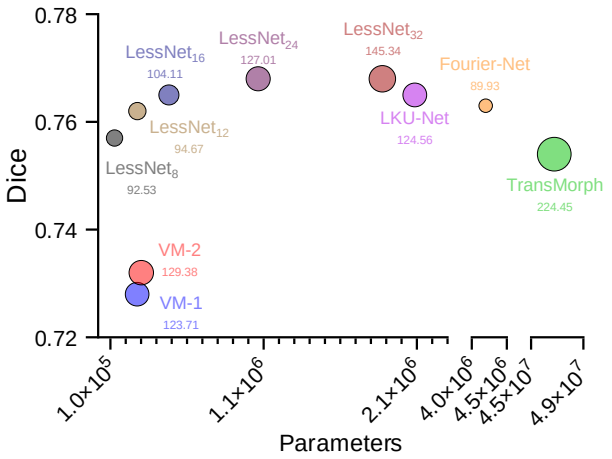

Fig. 1: Comparison of registration performance, training time, and the number of trainable parameters across different networks. In the figure, y-axis represents registration accuracy (measured by Dice) on the testing set, while the x-axis denotes the number of trainable parameters. The area of each marker corresponds to the training time (in seconds) required for 403 pairs of $160\times192\times224$ images (one epoch on the IXI dataset). LessNet achieves superior registration accuracy while reducing both training time and the number of parameters.

I. Introduction

MEDICAL image registration aims to establish the spatial correspondence between a moving image and a fixed image. It plays an important role in diverse healthcare applications [1]–[3], including disease diagnosis, disease progression monitoring, treatment planning, treatment guidance, etc.

Traditionally, unsupervised medical registration has been addressed through iterative optimization approaches [4]. These methods typically consist of three essential components. Firstly, a deformation model is defined, with options ranging from FFD [5], [6], LDDMM [7], DARTEL [8], to Demons [9] and others [4]. Secondly, an evaluation criterion is selected, which often incorporates a similarity constraint and a regularization term. Examples of similarity constraints include the mean squared error (MSE), normalized cross-correlation (NCC), mutual information (MI), and modality-independent neighborhood descriptor (MIND) [10]. Examples of regularization techniques encompass smoothness (e.g., diffusion regularizer, bending energy [11], or arbitrary order gradient [12]), inverse consistency [13], and diffeomorphism [14]. Finally, the deformation model under the given criterion is iteratively optimized using a specialized optimization technique, such as the Levenberg–Marquardt algorithm [8], deterministic or stochastic gradient descent [5], ADMM [15], etc. Such iterative registration approaches are designed to be explainable with clear mathematical derivations and can yield promising registration performance. A noteworthy aspect of these iterative approaches is their often slow optimization process and pairwise hyperparameter tuning, which restrict their application to large-scale imaging tasks. The emergence of recent data-driven approaches based on deep learning may have overcome these limitations [16].

In the context of deep learning-based unsupervised registration, neural networks replace conventional iterative processes and acquire registration knowledge through learning guided by an unsupervised loss function. Under this framework, there are two stages for registration: 1) training a network with a substantial amount of image pairs from a training set, and 2)

X. Jia, X. Cheng, and J. Duan are with the School of Computer Science, University of Birmingham, UK. W. Lu is with the Department of Computing and Mathematics, Manchester Metropolitan University, UK. We thank the computation support from Baskerville, which was funded by the EPSRC and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham. The corresponding author is Jinming Duan (j.duan@cs.bham.ac.uk).

deploying the trained network onto unseen image pairs from a test set to predict their displacement fields. More specifically, the registration is performed through a neural network $f(\boldsymbol{\Theta})$, parameterized by weights and biases $\boldsymbol{\Theta}$. Once trained, this network can efficiently predict a dense displacement field $\boldsymbol{u} = f(\boldsymbol{\Theta}; I_M, I_F)$ from the input image pair I_M and I_F. The network enables fast inference with a single forward pass, resembling a closed-form solution without iteration, thereby outperforming the time-consuming nature of iterative optimization. For instance, VoxelMorph [17] achieves comparable registration performance to iterative methods but runs orders of magnitude faster.

Following the success of VoxelMorph, which employs a U-Net architecture, numerous encoder-decoder style networks have been proposed for unsupervised image registration. Among them, some works, such as [18]–[21], leverage siamese- or dual-style networks that incorporate two identical encoders to capture features from moving and fixed images, respectively. Another line of research, represented by works from [22]–[28], entails the progressive composition of intermediate displacement fields to form the final displacement field. These approaches involve either cascading multiple U-Nets [23], [25] or integrating down-sampling techniques to construct multi-scale (pyramid) images or features for estimating multi-scale displacements or velocity fields in a coarse-to-fine manner [22], [24], [26]–[28]. The final deformation can be progressively refined or directly up-sampled from the down-scale deformations. Very recently, some works have explored vision-transformers to learn long-range information [29], [30]. While the debate between transformer- and convolution-based networks continues [31], a noticeable trend is the preference for larger networks with significantly more parameters and higher computational load. As an illustration, when comparing VoxelMorph to TransMorph, there is a respective increase of 136%, 17046%, and 216% in memory usage, the number of parameters, and the number of Mult-Adds. Consequently, training such large networks often demands substantially longer time. Fig. 1 further illustrates their detailed differences.

From VoxelMorph [17], [32] and Siamese-Net or Dual-Net [18]–[21] to the more recent TransMorph [30], the majority of registration networks have adopted the encoder-decoder (or contracting-expansion) style architecture. In this paradigm, input image pairs undergo a contracting path to encode high-level features, followed by an expansive path to decode these features into a dense, full-resolution displacement field. However, convolutions on full-resolution images or feature maps can entail intensive computations (Mult-Adds), especially when dealing with high-dimensional volumetric image data. Some works, such as DeepFlash [33], B-Spline [34], [35], Fourier-Net [36], [36], have recognized this drawback and proposed approaches to learn low-dimensional representations of the displacement field, which can significantly reduce the computational load resultant from convolutional operations in either the encoder or the decoder. Unfortunately, these networks exhibit limitations, as discussed in Sec. II.

To handle high-dimensional volumetric image data more effectively, we propose LessNet, which eliminates the entire learnable encoder and relies solely on a convolutional decoder to learn displacement fields from image pairs. A schematic comparison between the architectures of some popular registration networks as well as our LessNet is given in Fig. 2. The main contributions of this work are summarized as follows:

- While the majority of deep unsupervised registration networks adopted symmetric encoder-decoder (or contracting-expansion) style architectures, we demonstrated the presence of redundancy in the encoder and thereby proposed a simplified decoder-only architecture for medical image registration.
- As a proof of concept, we employed simple handcrafted features to replace the entire trainable encoder. These features comprise three distinct pooling operations, namely max pooling, average pooling, and min pooling. We showed in our experiments that these manually designed, multi-scale features are already effective for the decoder to learn dense, full-resolution displacement fields from image pairs. It is important to highlight that, beyond these pooling features, alternative choices such as incorporating other handcrafted features or features from large pre-trained networks (Sec. IV-G) may also be valid options.
- Evaluated on five registration tasks, our decoder-only LessNet demonstrated comparable accuracy to established encoder-decoder style networks, while reducing computational load significantly, as illustrated in Fig. 1.

II. Related Works

A. Encoder-Decoder Style Networks

Early efforts have been made to estimate dense displacement fields for 3D deformable image registration, as demonstrated in [37], where the authors introduced a voxel-to-voxel encoder-decoder style fully convolutional network (FCN). Such an FCN model was optimized using Adam, with NCC as the data term and total variation as the regularization term. Balakrishnan et al. [17], [32] further advanced the field by introducing VoxelMorph for unsupervised deformable image registration, which employs U-Net [38] as its backbone instead of the FCN [37]. In this framework, they explored the use of MSE and local normalized cross-correlation (LNCC) as the data term, along with a diffusion regularizer. Additionally, in [32] a Dice loss based on anatomical segmentation masks was incorporated during training to improve registration accuracy.

Qin et al. [18] presented a framework for jointly learning motion and segmentation in cardiac sequences. They utilized a Siamese-style recurrent network as the backbone, with the loss function incorporating the MSE between warped frames and the target frame as well as the regularization that induces both spatial and temporal smoothness of the displacement fields. Hu et al. introduced the dual-stream pyramid network (Dual-PRNet) [20] and later extended it to Dual-PRNet++ [21]. In contrast to VoxelMorph, which is a single-stream encoder-decoder network that predicts the full-resolution displacement field from the last convolutional layer, Dual-PRNet predicts multiple intermediate displacement fields in a coarse-to-fine manner. The final displacement field is integrated through up-samplings and warpings on intermediate displacement fields. Their loss function includes the NCC and diffusion regularizer.

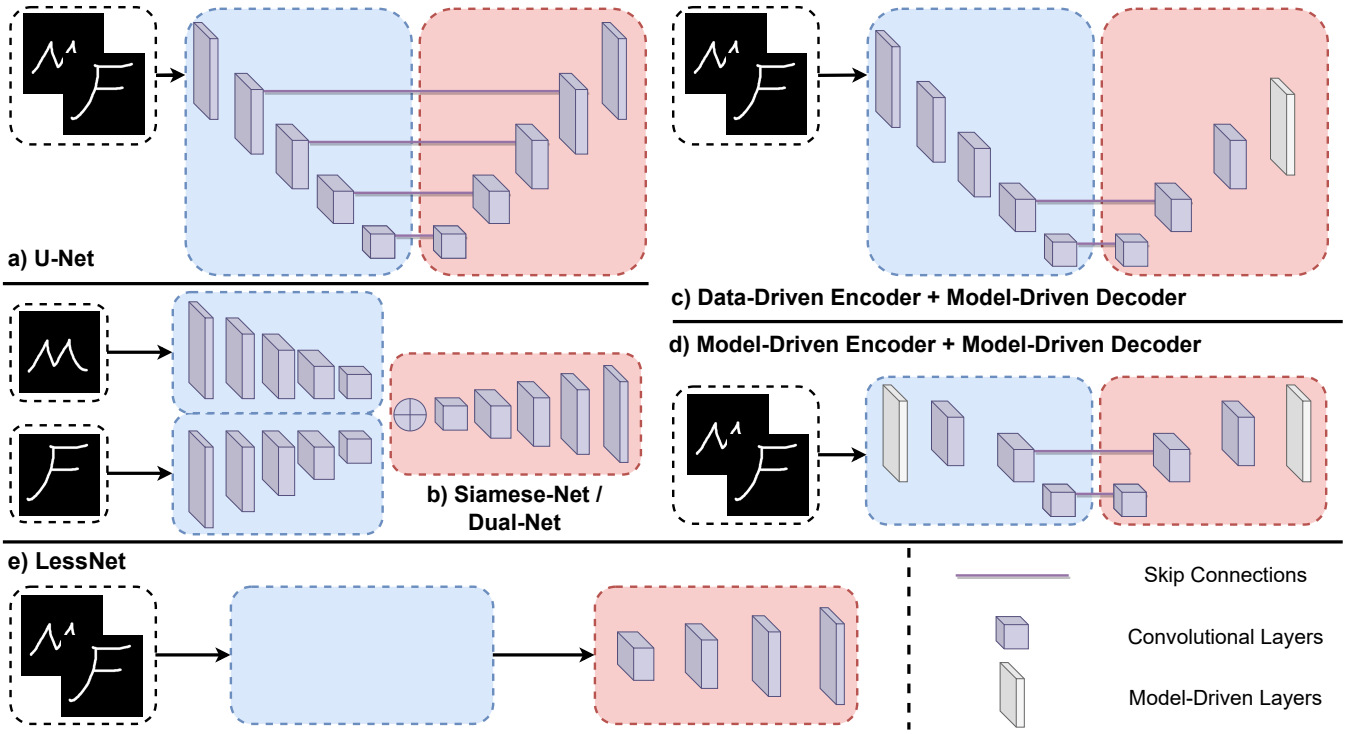

Fig. 2: Schematic overview of different encoder-decoder architectures used in medical image registration. a) U-Net style network: The moving and fixed images are stacked into a two-channel input (indicated by the black box). The encoder (blue box) and decoder (red box) follow a symmetric layout, with skip connections linking corresponding features between them. (b) Siamese-Net (Dual-Net): The moving and fixed images are processed separately by two parallel encoders. The decoder then integrates features from both encoders and maps them to a registration field. c) and d) Model-driven networks: Some learnable layers are replaced by model-driven layers which are pre-defined, knowledge-driven parameter-free blocks. These hybrid approaches often lead to a reduced number of network parameters, fewer Mult-Adds, and thereby faster training and inference speeds. e) LessNet: Our network stands out by not having a learnable encoder at all. Instead, the decoder learns a full-resolution registration field directly from the input images.

Zhao et al. [23] introduced a recursive cascaded network architecture, which they termed RC-Net. This approach enables sequential warpings of the moving image, facilitating the estimation of large displacement fields. Within RC-Net, each cascade employs a U-Net style network to predict a small displacement field for an image pair, which consists of the warped moving image from the previous cascade and the original fixed image. The final displacement field is a composite of all small displacement fields predicted by each cascade. Using multiple cascaded networks (e.g., 10 cascades in their study), RC-Net demonstrated substantial improvement over VoxelMorph [17] in both liver and brain registration tasks. Jia et al. proposed VR-Net [25], which unrolls the mathematical structure of an iterative variational optimization through variable splitting and seamlessly integrating it into a deep neural network in a cascading fashion. However, optimal performance in VR-Net relies on the existence of an initial displacement field, which is predicted by an additional U-Net.

While algorithms such as [17], [23], [25], [32], [34] demonstrated efficacy in fast unsupervised registration, they lack a guarantee or promotion of inverse-consistency and topology preservation during the registration process. To address these limitations, Zhang [39] proposed an inverse-consistent U-Net style network (IC-Net), which enforces inverse consistency on both the forward and backward displacement fields. This ensures that the image pair is symmetrically deformed towards each other. To further mitigate folding issues in the deformation, they introduced an anti-folding constraint along with a local smoothness term. In the work by Dalca et al. [40] and its subsequent extension [41], a probabilistic diffeomorphic registration network was introduced to learn diffeomorphisms for deformation fields. Initially, they proposed to learn a distribution of stationary velocity fields using a variational 3D U-Net architecture. Diffeomorphism was achieved by applying scaling and squaring [8] to the stationary velocity field. Building upon IC-Net and the probabilistic diffeomorphic registration networks [39], [40], several works have been introduced to further enhance the accuracy of diffeomorphic registration, including SYM-Net [42], LapIRN [24], and CycleMorph [43].

Recently, architectures based on vision transformers [44] have drawn a lot of attention in the registration community due to their capacity to capture long-range dependencies [29], [30]. It is noteworthy that despite the replacement of basic convolutional blocks with more advanced attention blocks in these transformer-based architectures, they still follow the classical encoder-decoder style [31].

B. Model-Driven Networks

Instead of making an entire network learnable across all layers, model-driven networks replace some learnable layers with pre-defined, knowledge-driven parameter-free modules. This approach often leads to a reduced number of parameters, fewer Mult-Adds, and thereby faster training and inference speeds [33]–[36], [45], [46].

B-Spline networks such as DIR-Net [45] and DLIR [34] estimated a grid of control points. The full-resolution displacement field was then interpolated from these points using a cubic b-spline function, which serves as a mathematical model. Qiu et al. [35] introduced Diff-B-Spline, a diffeomorphic b-spline network designed for modality-invariant registration utilizing mutual information. In [35], they pruned the decoder by discarding several convolutional layers to predict b-spline parameterized velocities. These velocities, represented as regularly spaced control points, can then be interpolated into dense diffeomorphic deformation fields using scaling and squaring. B-Spline networks [34], [35], [45] typically require a local interpolation process. This characteristic often leads to a compromise in capturing global details, consequently impacting their overall registration performance.

The multi-step DeepFlash [33] performs registration by first predicting a band-limited velocity field and subsequently converting it to the full-resolution deformation field using a model-driven partial differential equation. Note that the multi-step process makes DeepFlash cumbersome and difficult to implement. It is also important to note that DeepFlash relies on supervision signals for training, and therefore its performance may be bounded by the effectiveness of the underlying iterative method [33] used to generate these supervision signals.

Fourier-Net [36] is an end-to-end and unsupervised approach capable of predicting a compact, low-dimensional representation of the displacement field in the band-limited Fourier space. Within Fourier-Net, a model-driven decoder effectively reconstructed the full-resolution displacement field from a few band-limited coefficients. Building upon Fourier-Net, the authors introduced Fourier-Net+ [46], which learns the band-limited displacement field from the band-limited representation of images. Fourier-Net+ further accelerated registration speed by constraining both the input and output of the network to low-dimensional representations, thereby reducing the need for repeated convolution operations. However, the process of learning the low-dimensional band-limited representation often results in the loss of high-frequency signals in displacement fields. To address this limitation, the use of multiple cascades, as demonstrated in Fourier-Net+ [46], became necessary to effectively handle complex and large displacement fields.

III. LessNet

A. Redundancy in Encoder

The use of encoder-decoder architectures has been playing an essential role in medical image registration. For this, we ask a question: do the encoder and the decoder equally contribute to the estimation of displacement fields? This question has not been answered by researchers in the field of image registration using deep learning. To answer it, we used the classical architecture of VoxelMorph as an example. More specifically, we employed VoxelMorph-1 in [17] and VoxelMorph-2 in [32] (their details have been given described in Section IV) and conducted the following experiments for each network. Notably, skip connections were still used.

TABLE I: The effects of enabling or disabling parameter updates in the encoder or decoder of VoxelMorph-1 and VoxelMorph-2. ✓ indicates learning is enabled, while ✗ denotes learning is disabled.

Methods	Encoder	Decoder	Dice↑	$\lvert J\rvert_{<0}\%$
Initial	-	-	0.544±0.089	-
VM-1	✓	✗	0.679±0.051	0.486±0.341
VM-2	✓	✗	0.690±0.051	0.537±0.350
VM-1	✗	✓	0.747±0.043	0.818±0.400
VM-2	✗	✓	0.747±0.043	0.826±0.390
VM-1	✓	✓	0.757±0.039	0.723±0.370
VM-2	✓	✓	0.757±0.040	0.793±0.405

- We randomly initialized all parameters in the encoder and the decoder and trained these parameters from scratch.
- We randomly initialized all parameters in the encoder and the decoder, then froze the decoder parameters (except the final output layer), and trained the encoder parameters (and the final output layer) only.
- We randomly initialized all parameters in the encoder and the decoder, then froze the encoder parameters and trained the decoder parameters only.

Analysis of the results in Table I indicates that with a randomly initialized encoder, the registration performance would not be significantly decreased, suggesting the learning of the encoder is less critical. Consequently, we argue that, for medical image registration, a learnable encoder may not be necessary if the decoder is appropriately designed. Furthermore, we note that even a randomly initialized encoder in Table I can still incur considerable convolutional computations and memory usage. This raises the question: can handcrafted features be employed as an alternative? The answer is Yes.

B. Pooling Features

As a proof of concept, we employed three parameters-free, computationally efficient pooling operations to substitute learnable layers in the conventional encoder. As illustrated in Fig. 3, the original 3D image pair comes with a resolution of $2 \times H \times W \times D$. Employing a max pooling operation with a kernel size of $2 \times 2 \times 2$ and a stride of 2, a feature map of dimensions $2 \times \frac{H}{2} \times \frac{W}{2} \times \frac{D}{2}$ was generated. Similarly, using average and min pooling operations, we obtained corresponding feature maps. By concatenating these pooling features, we created a six-channel feature map with a spatial resolution of $\frac{H}{2} \times \frac{W}{2} \times \frac{D}{2}$ in the first layer. Next, employing a kernel size of $4 \times 4 \times 4$ with a stride of 4 on the original input pair, a six-channel feature map was obtained with a spatial resolution of $\frac{H}{4} \times \frac{W}{4} \times \frac{D}{4}$ in the second layer. Subsequently, utilizing a kernel size of $8 \times 8 \times 8$ with a stride of 8 on the original input pair, a six-channel feature map was generated with a spatial resolution of $\frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$ in the third layer.

The $6 \times \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$ pooling feature map was used as input to the first decoding layer. Simultaneously, the remaining

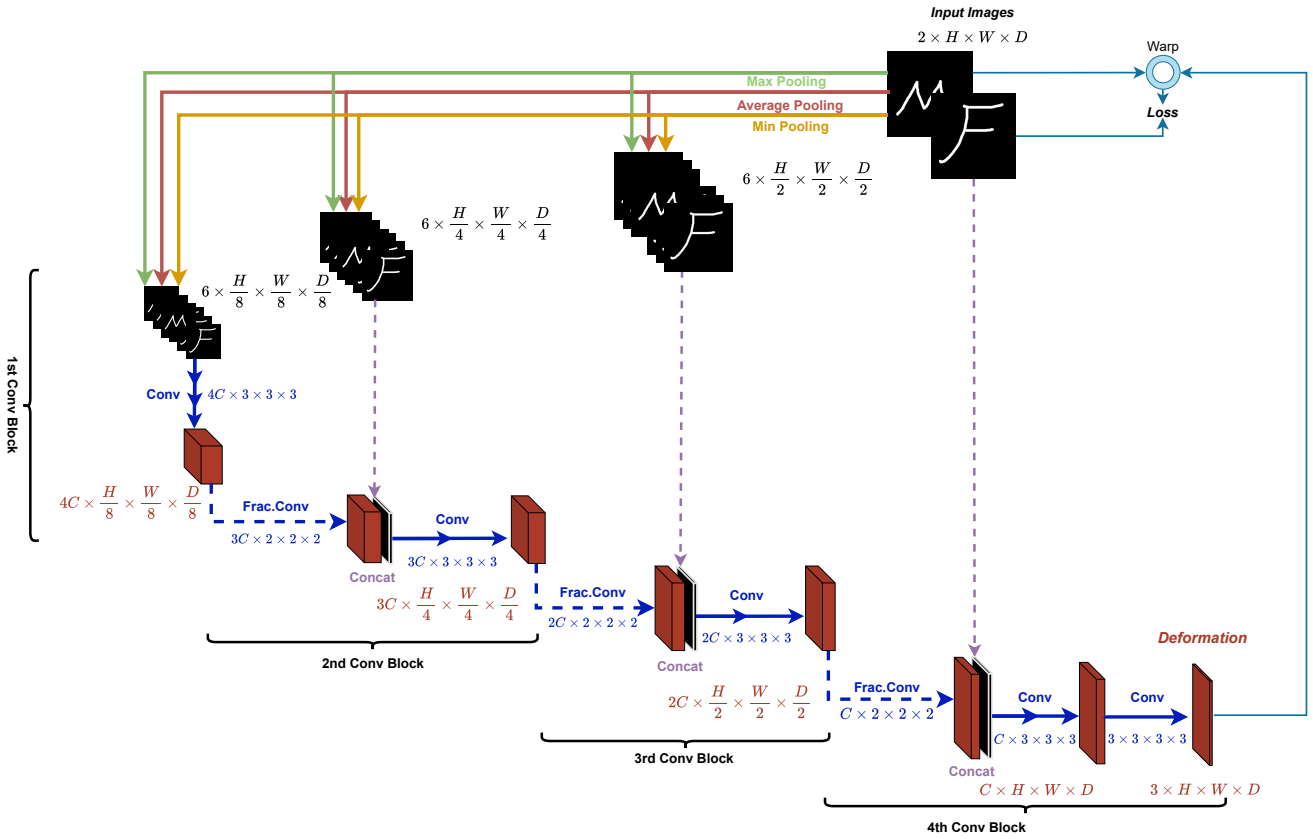

Fig. 3: The architecture of LessNet. The upper half panel demonstrates the generation of multi-scale pooling features, while the lower half panel showcases the input and output of the learnable decoder, which consists of four hierarchical convolutional blocks. The loss function is applied to the moving image, warped by the predicted displacement field, and the fixed image.

pooling features, such as $6 \times \frac{H}{4} \times \frac{W}{4} \times \frac{D}{4}$ and $6 \times \frac{H}{2} \times \frac{W}{2} \times \frac{D}{2}$, were concatenated into their corresponding decoder layers as supplementary cues for registration. Additionally, the original image pairs was inserted into an appropriate decoding layer to provide additional information.

In these instances, we effectively leveraged the information contained in the moving and fixed image pair without the necessity of a learnable encoder. It is important to mention that alternative handcrafted features or features extracted from pre-trained networks are also valid options. We encourage the reader to experiment with various handcrafted or pre-trained features that could potentially improve registration performance. However, for the purpose of this study, we limited our exploration to simple pooling features to showcase the feasibility of decoder-only architectures for image registration.

C. Decoder in LessNet

As shown in Fig. 3, the decoder in LessNet consists of four hierarchical convolutional blocks. In the following, we provide a detailed explanation of each convolutional block:

- **The first convolutional block**: The input to this block is a six-channel feature map with a spatial resolution of $\frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$. This feature map undergoes convolution with kernels of size $4C \times 3 \times 3 \times 3$. Finally, the output of this block is a $4C \times \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$ feature map.
- **The second convolutional block**: The input to this block is the $4C \times \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$ feature map from the last block. This feature map is firstly upsampled to $3C \times \frac{H}{4} \times \frac{W}{4} \times \frac{D}{4}$ using fractional convolution with kernels of size $3C \times 2 \times 2 \times 2$. The upsampled feature map is subsequently concatenated with the $6 \times \frac{H}{4} \times \frac{W}{4} \times \frac{D}{4}$ pooling features, yielding a new feature map of size $(3C+6) \times \frac{H}{4} \times \frac{W}{4} \times \frac{D}{4}$. This new feature map undergoes convolution with kernels of size $3C \times 3 \times 3 \times 3$. As a result, the output of this block is a $3C \times \frac{H}{4} \times \frac{W}{4} \times \frac{D}{4}$ feature map.
- **The third convolutional block**: The input to this block is the $3C \times \frac{H}{4} \times \frac{W}{4} \times \frac{D}{4}$ feature map from the last block. This feature map is firstly upsampled to $2C \times \frac{H}{2} \times \frac{W}{2} \times \frac{D}{2}$ using fractional convolution with kernels of size $2C \times 2 \times 2 \times 2$. The upsampled feature map is subsequently concatenated with the $6 \times \frac{H}{2} \times \frac{W}{2} \times \frac{D}{2}$ pooling features, yielding a new feature map of size $(2C+6) \times \frac{H}{2} \times \frac{W}{2} \times \frac{D}{2}$. Next, this new feature map undergoes convolution with kernels of size $2C \times 3 \times 3 \times 3$. As a result, the output of this block is a $2C \times \frac{H}{2} \times \frac{W}{2} \times \frac{D}{2}$ feature map.
- **The fourth convolutional block**: The input to this block is the $2C \times \frac{H}{2} \times \frac{W}{2} \times \frac{D}{2}$ feature from the last block. This feature map is firstly upsampled to $C \times H \times W \times D$ using fractional convolution with kernels of size $C \times 2 \times 2 \times 2$. The upsampled feature map is then concatenated with the original image pair, yielding a new feature map of size $(C+2) \times H \times W \times D$. Next, this new feature map undergoes convolution with kernels of size $C \times 3 \times 3 \times 3$, producing yet another feature map of size $C \times H \times W \times$

D. This feature map then undergoes a final convolution with kernels of size $3 \times 3 \times 3 \times 3$, outputting the dense displacement field of size $3 \times H \times W \times D$.

Note that, we used LeaklyReLU with a default negative slope of 0.01 as the activation function after each convolution apart from the final output layer, for which we used the SoftSign activation function. On the other hand, we crafted feature maps of size $\frac{H}{h} \times \frac{W}{w} \times \frac{D}{d}$ throughout the entire network, where $h = w = d$, and their values were either 2, 4, or 8. However, the values of h, w, and d do not necessarily have to be equal, and one can choose appropriate values with slight modifications to the proposed architecture in Fig. 3.

D. Loss Functions

The proposed LessNet is a general unsupervised learning framework, making it adaptable to typical registration loss functions. First, we assume that LessNet predicts directly displacement fields. In this case, the loss function $\mathcal{L}(\mathbf{\Theta})$ for LessNet is given by:

$$\mathbf{\Theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_S(I_{M_i} \circ (\boldsymbol{u}_i + \mathrm{Id}), I_{F_i}) + \frac{\lambda}{N} \sum_{i=1}^{N} \mathcal{L}_R(\boldsymbol{u}_i), \quad (1)$$

with

$$\boldsymbol{u}_i = f(\mathbf{\Theta}; I_{M_i}, I_{F_i}).$$

Here, $\boldsymbol{u}_i$ denotes the displacement field, predicted by the network f parameterized by $\mathbf{\Theta}$ on the input image pair I_{M_i} and I_{F_i}. Id is the identity grid, N is the number of training pairs, and $\circ$ is the warping operator. $\mathcal{L}_S$ represents the similarity (data) loss, while $\mathcal{L}_R$ denotes the regularization loss. The hyperparameter λ balances the two terms. In experiments, we explored the use of MSE and LNCC for $\mathcal{L}_S$, along with a first-order diffusion regularizer for $\mathcal{L}_R$.

On the other hand, LessNet can be directly applied to diffeomorphic image registration with only minor modifications to the network output and loss function. Computing a diffeomorphic deformation can be viewed as modeling a dynamical system [7] given by an ordinary differential equation (ODE): $\partial\phi/\partial t = \boldsymbol{v}_t(\phi_t)$, where $\phi_0 = \mathrm{Id}$ represents the identity transformation, and $\boldsymbol{v}_t$ signifies the velocity field at time t ($\in [0, 1]$). Alternatively, a diffeomorphic deformation can be modeled with the stationary velocity field [8] through: $\partial\phi/\partial t = \boldsymbol{v}(\phi_t)$, where the velocity field $\boldsymbol{v}$ is assumed constant over time. In this paper, we utilized the stationary velocity field parameterized implementation. In such a case, the loss function of LessNet for diffeomorphic registration becomes:

$$\mathbf{\Theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_S(I_{M_i} \circ \boldsymbol{Exp}(\boldsymbol{v}_i), I_{F_i}) + \frac{\lambda}{N} \sum_{i=1}^{N} \mathcal{L}_R(\boldsymbol{v}_i), \quad (2)$$

with

$$\boldsymbol{v}_i = f(\mathbf{\Theta}; I_{M_i}, I_{F_i}).$$

The differences between the two losses (1) and (2) are twofold. First, (2) requires the network f to predict a stationary velocity field $\boldsymbol{v}$ rather than a displacement field $\boldsymbol{u}$. Second, in (2), we need to exponentiate the predicted stationary velocity field (i.e., $\boldsymbol{Exp}(\boldsymbol{v})$) to attain a diffeomorphic deformation. To implement such an exponential function, we use seven scaling and squaring operations as in [8], [40]. All operations in LessNet are differentiable, thereby enabling optimization through standard backpropagation.

IV. Experiments

A. Datasets and Evaluation

OASIS-1 dataset [47]: We used a processed version of this dataset, provided by [48], [49], to perform subject-to-subject (inter-subject) brain registration. The dataset has 414 2D 160×192 slices and masks contain 24 anatomical structures extracted from their corresponding 3D $160 \times 192 \times 224$ volumes. The 2D slice was extracted from the middle of the coronal plane. Detailed information about the pre-processing protocol can be accessed at [1]. The 414 images were randomly split into three sets: 201 images for training, 12 images for validation, and 201 images for testing. We then constructed 40200 ([201-1]$\times$201), 22 ([12-1]$\times$2), and 400 ([201-1]$\times$2) image pairs for training, validation, and testing, respectively. All 24 anatomical structures are used to compare the registration performance.

IXI dataset[2]: We used a processed version of this dataset, provided by [30], to perform atlas-to-subject registration. Following their exact evaluation protocol, we employed 403 images for training, 58 for validation, and 115 for testing. The atlas was generated by [30] with the method presented in [43]. All volumes and label maps (depicting 30 anatomical structures) were cropped to the size of $160 \times 192 \times 224$.

ACDC dataset [50] comprises cardiac MRI images from 150 patients, evenly distributed across five pathological categories. For our experiments, we selected 100 subjects with available manual annotations. The three annotated structures include the left ventricle, right ventricle and myocardium. We randomly split these subjects into 40 for training, 10 for validation, and 50 for testing to perform intra-subject end-systole (ES) to end-diastole (ED) registration using mid-vertical slices. Given that the in-plane resolution varies between 1.34 mm and 1.68 mm, we resampled all images to a uniform resolution of 1.8 mm and center-cropped the 128×128 regions before conducting the experiments.

CHAOS Abdomen MR dataset [51]: We used a pre-processing version provided by the Learn2Reg challenge [3] to perform inter-subject registration. The 40 abdomen scans in this database have been normalized into the $192 \times 160 \times 192$. We randomly divided the 40 scans into 16 for training, 5 for validation, and 19 for testing. After pairing the scans, this resulted in 240 ([16-1]$\times$16) training pairs, 20 ([5-1]$\times$5) validation pairs, and 38 (19$\times$2) testing pairs. The registration performance is evaluated with the four annotations of the liver, left and right kidneys, and spleen, and there were no tumors or lesions at the borders of the annotated organs of interest.

Multi-site Multi-study dataset: To further evaluate the generalization of our method, we constructed a large-scale,

[1] https://github.com/adalca/medical-datasets/blob/master/neurite-oasis.md

[2] https://brain-development.org/ixi-dataset/

[3] https://learn2reg.grand-challenge.org/Learn2Reg2021/

multi-site, multi-study dataset using the OpenBHB [52] and Mindboggle-101 [53] T1 brain data, resulting in 4046 3D brain scans from 13 public brain MR sources, including IXI, ABIDE 1[4], ABIDE 2[5], CoRR [54], GSP [55], LOCALIZER [56], MPI-Leipzig [57], NAR [58], NPC [59], and RBP [60], [61], NKI-RS-22[6], NKI-TRT-24[7], and OASIS-TRT-20 [47]. Standard preprocessing steps, including affine alignment, skull stripping, intensity normalization, and cropping ($160\times192\times160$) have been applied for each scan. We used the first 3984 images from the first 10 datasets of OpenBHB as the training set, we randomly split the 62 images from the 3 datasets (i.e., NKI-RS-22, NKI-TRT-24, and OASIS-TRT-20 of Mindboggle-101) into 6 atlas images, 6 validation images, and 50 testing images, resulting in 3983, 36, 300 pairs for training, validation, and testing. The registration performance is evaluated on 50 manual annotated structures, following [62], [63].

Evaluation metrics: Dice score was employed to assess the overlapping ratio between anatomical structures. The better the registration performs, the higher the score. In addition, the percentage of negative Jacobian determinants in the deformation, denoted as $|J|_{<0}\%$, was reported to assess whether the deformation is realistic and plausible. A higher percentage indicates more foldings in the deformation, while a lower percentage indicates fewer foldings. In the case of a perfect diffeomorphism, there will be no folding, and therefore the $|J|_{<0}\%$ is expected to be zero. RMSE (root mean square error) was also reported for a comprehensive analysis.

B. Implementation Details

As shown in Fig. 3, the overall model size of LessNet is controlled by the hyperparameter C, which we set to 16 by default. We implemented our proposed networks in PyTorch, where training was optimized using Adam with a learning rate of 0.0001 and a batch size of 1. To adapt LessNet to 2D, we changed 3D kernels to 2D, each with a size of 3×3. For training in both 2D and 3D, we tuned built-in hyperparameters on respective held-out validation sets. In terms of loss functions, we employed MSE to train our networks on OASIS-1 for 20 epochs, achieving optimal performance with $\lambda = 0.01$. On IXI, LessNet was trained with LNCC for 500 epochs with $\lambda = 5$, while the diffeomorphic LessNet (Diff-LessNet) was trained optimally with $\lambda = 2$. On the ACDC dataset, all models were trained for 500 epochs using MSE loss with $\lambda = 0.01$. On the CHAOS dataset, all models were trained for 100 epochs using MSE loss with $\lambda = 0.05$. On the multi-site, multi-study dataset, all models were trained for 500 epochs using LNCC loss with $\lambda = 1.0$. We note that the optimal λ was selected based on the highest Dice score obtained during tuning on the validation set. The training time depicted in Fig. 1 was calculated using an A100 GPU using the FastLNCC [64] [8].

[4]http://fcon_1000.projects.nitrc.org/indi/abide/abide_I.html

[5]http://fcon_1000.projects.nitrc.org/indi/abide/abide_II.html

[6]https://fcon_1000.projects.nitrc.org/indi/pro/nki.html

[7]https://fcon_1000.projects.nitrc.org/indi/pro/eNKI_RS_TRT/FrontPage.html

[8]https://github.com/xi-jia/FastLNCC

TABLE II: The impact of using multi-scale pooling features on the OASIS-1 dataset. For example, $\frac{1}{8}$ denotes the utilization of $6 \times \frac{H}{8} \times \frac{W}{8}$ pooling features, while I_M and I_F represent the use of the original images.

$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	I_M, I_F	Dice↑	RMSE	$\lvert J\rvert_{<0}\%$
✓	✗	✗	✗	0.706±0.039	0.045±0.004	0.487±0.363
✓	✓	✗	✗	0.748±0.038	0.034±0.004	0.644±0.362
✓	✓	✓	✗	0.759±0.040	0.031±0.004	0.783±0.391
✓	✓	✓	✓	0.761±0.039	0.031±0.004	0.742±0.353

The CPU and GPU runtimes presented in Tables VI and VII were obtained from one local machine equipped with an RTX 2080Ti GPU, a 3.80GHz Intel(R) Core(TM) i7-9800X CPU with 128GB of RAM. The computational time, including the loading cost of models and images, was averaged over the entire test set with a batch size of 1.

C. Impact of Pooling Features

First, we investigated the necessity of concatenating different resolutions of pooling feature maps as well as the original image pair to different blocks of the decoder of LessNet. To explore this, we conducted experiments with four different settings: 1) We disabled the concatenation of the original image pair (I_M and I_F) and the pooling features to the last three conv blocks, but only concatenated the $6 \times \frac{H}{8} \times \frac{W}{8}$ pooling features to the first block; 2) We disabled the concatenation of the original image pair and $6 \times \frac{H}{2} \times \frac{W}{2}$ pooling feature, but concatenated the $6\times \frac{H}{8} \times \frac{W}{8}$ and $6\times \frac{H}{4} \times \frac{W}{4}$ pooling features; 3) We disabled the concatenation of only the original image pair, but concatenated the three different resolutions of pooling features; and 4) we enabled all the concatenations.

As seen in Table II, the model achieved a Dice score of 0.706 under Setting 1. The Dice score was improved with the inclusion of more pooling features. In particular, there was a rapid increase in Dice to 0.748 under Setting 2, yielding a 4.2% performance gain. A further increase of 1.1% can be observed, achieving a Dice score of 0.759 under Setting 3. Finally, the highest performance was achieved under Setting 4, resulting in a Dice score of 0.761. These results underscore the necessity of utilizing multi-scale pooling features.

We randomly selected a test pair and generated visualizations, including deformation fields and $|J|$ maps for different scenarios in Table II. As shown in Fig. 4, using more levels of features not only enhances the Dice performance but also increases the $|J|_{<0}\%$, resulting in more folding areas (highlighted in green).

Next, we assessed the effectiveness of employing three different types of pooling operations: min, max, and average pooling. In Table III, the registration performances of min pooling and average pooling are comparable, achieving Dice scores of 0.757 and 0.758, respectively. Max pooling slightly lags behind at 0.753. Combining all three pooling operations resulted in the highest Dice score of 0.761.

D. Model Size

As shown in Fig. 3, the overall model size of LessNet is controlled by the hyperparameter C. Accordingly, we investigated

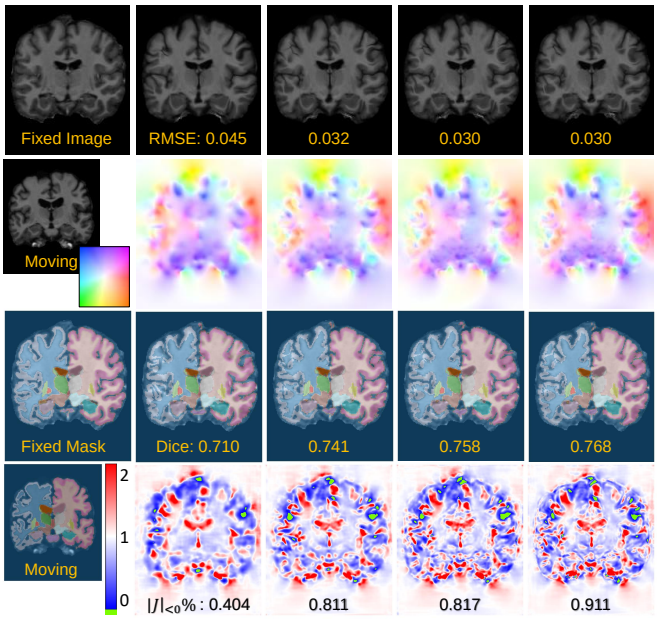

Fig. 4: Visual comparisons of models using different scales of pooling features. From left to right: models incorporating 1/8, 1/4, 1/2 pooling features and original images. The used color coding [65] for the displacement is shown in the right corner of the moving image. The color bar of $|J|$ map is shown in the right corner of moving masks. The detailed values of RMSE, Dice, and $|J|_{<0}\%$ are reported for a comprehensive analysis.

TABLE III: Comparison of registration performance between different types of pooling operations on OASIS-1.

Min	Average	Max	Dice↑	RMSE	$\lvert J\rvert_{<0}\%$
✓	✗	✗	0.757±0.040	0.033±0.004	0.868±0.393
✗	✓	✗	0.758±0.040	0.031±0.004	0.802±0.409
✗	✗	✓	0.753±0.041	0.032±0.004	0.856±0.401
✓	✓	✓	0.761±0.039	0.031±0.004	0.742±0.353

the performance of LessNet with various values of C, namely, 8, 12, 16, 24, and 32. The results in Table IV indicate that increasing the value of C consistently improves registration performance, albeit at the cost of additional computational resources. For instance, when C was set to 8, LessNet achieved a 0.749 Dice score, with 44,336 network parameters, 10.31MB of GPU memory with one forward and backward pass, and 152 million Mult-Adds. The Dice score increases by 0.8%, 1.2%, 1.7%, and 1.9%, respectively when C was set to be 12, 16, 24, and 32. However, setting C to 32 resulted in a 14.74-fold increase in parameters, a 3.93-fold increase in memory usage, and a 14.47-fold increase in Mult-Adds. Since the same λ is used for all models, the $|J|_{<0}\%$ shows minimal fluctuations and does not vary significantly across different values of C.

E. Diffeomorphism

Finally, we investigated whether the proposed LessNet supports diffeomorphic registration. In this case, the output from the last convolutional layer was a stationary velocity field, and the deformation field was integrated through 7 scaling and squaring operations [31], [36], [40]. In Table V, we observed that Diff-LessNet produced nearly zero $|J|_{<0}\%$.

TABLE IV: Comparison between different LessNet architectures, with model sizes controlled by the hyperparameter C. Mult-Adds and Memory are measured in millions (M) and megabytes (MB), respectively.

C	Parameter	Mult-Adds	Memory	Dice↑	$\lvert J\rvert_{<0}\%$
8	44,336	152	10.31	0.749±0.040	0.808±0.389
12	96,264	327	15.23	0.757±0.040	0.749±0.376
16	168,032	570	20.16	0.761±0.039	0.742±0.353
24	371,088	1250	30.00	0.766±0.039	0.852±0.392
32	653,504	2200	39.84	0.768±0.039	0.830±0.397

TABLE V: Comparison of registration performance between non-diffeomorphic and diffeomorphic LessNet (Diff-LessNet).

Scaling & Squaring	C	Dice↑	$\lvert J\rvert_{<0}\%$
✗	8	0.749±0.040	0.808±0.389
✗	12	0.757±0.040	0.749±0.376
✗	16	0.761±0.039	0.742±0.353
✗	24	0.766±0.039	0.852±0.392
✗	32	0.768±0.039	0.830±0.397
✓	8	0.747±0.039	<0.0001
✓	12	0.754±0.039	<0.0001
✓	16	0.758±0.038	<0.0001
✓	24	0.761±0.038	<0.0001
✓	32	0.762±0.039	<0.0001

However, on this specific dataset, the registration performance of different Diff-LessNets consistently lags behind their non-diffeomorphic counterparts in terms of Dice score. It is important to note that this phenomenon may not be universally valid for other datasets. For instance, Diff-LessNet achieved higher accuracy than LessNet on IXI, as shown in Table VII.

F. Performance Comparison

1) Candidate Methods for Comparison: In this section, we detail 10 related methods for comparison. For all five registration tasks, methods requiring training were optimized with the Adam optimizer, a learning rate of 0.0001, and a batch size of 1 to ensure a fair comparison. Other hyperparameters were optimally tuned using the validation set for each network, unless otherwise specified.

- VoxelMorph. Two non-diffeomorphic versions, namely VoxelMorph-1 [17] and VoxelMorph-2 [32], along with one diffeomorphic version [40], were adopted using the official implementations [9]. Such models were adapted to 2D registration by replacing 3D convolutional kernels with 2D counterparts. For the IXI dataset, we used the public weights trained by [30].
- TransMorph [30]. We adopted the official release of both 2D and 3D implementations [10]. For the IXI dataset, we used the official weights released by the authors.
- Diff-B-Spline [35]. The official implementation[11] was adopted. Specifically, for our 2D experiments, the optimal control spacing was set to 4, while for our 3D experiments, the control spacing was set to 3. For the IXI dataset, we used the public weights trained by [30].
- Flash [33], LKU-Net [31], and Fourier-Net [36]. Their results on OASIS-1 and IXI were directly copied from

[9] https://github.com/voxelmorph/voxelmorph
[10] https://github.com/junyuchen245/TransMorph_Transformer_for_Medical_Image_Registration
[11] https://github.com/qiuhuaqi/midir

TABLE VI: Comparison of registration performance between various methods on the OASIS-1 dataset.

Methods	Dice↑	RMSE	$\lvert J\rvert_{<0}\%$	Parameters	Mult-Adds (M)	Memory (MB)	CPU (s)	GPU (s)
Initial	0.544±0.089	0.083±0.009	-	-	-	-	-	-
Flash [33]	0.734±0.045	-	0.049±0.080	-	-	-	85.773	-
VoxelMorph-1 [17]	0.757±0.039	0.032±0.004	0.723±0.370	91,578	538.77	15.29	0.007	0.014
VoxelMorph-2 [32]	0.757±0.040	0.031±0.004	0.793±0.405	100,530	690.29	19.51	0.009	0.014
Diff-VoxelMorph [40]	0.740±0.044	0.038±0.005	0.019±0.082	102,532	294.28	9.90	0.008	0.015
TransMorph [30]	0.768±0.039	0.028±0.003	0.777±0.398	31,005,506	3000	30.53	0.036	0.021
Diff-TransMorph [30]	0.748±0.043	0.036±0.005	0.022±0.076	30,934,084	2170	15.29	0.034	0.021
Diff-B-Spline-TM [30]	0.759±0.038	0.033±0.004	<0.0001	31,017,090	2710	20.70	0.035	0.021
LKU-Net [31]	0.763±0.039	0.030±0.004	0.739±0.391	551,990	713.09	51.68	0.014	0.016
Diff-LKU-Net [31]	0.757±0.038	0.033±0.004	<0.0001	551,990	713.09	51.68	0.016	0.017
Diff-B-Spline [35]	0.737±0.038	0.040±0.005	0.015±0.069	88,690	139.40	7.49	0.012	0.015
Fourier-Net [36]	0.756±0.039	0.036±0.004	0.753±0.408	1,427,376	888.25	35.89	0.011	0.015
Diff-Fourier-Net [36]	0.756±0.037	0.037±0.004	<0.0001	1,427,376	888.25	35.89	0.015	0.015
LessNet_8	0.749±0.040	0.033±0.004	0.808±0.389	44,336	151.80	10.31	0.006	0.014
Diff-LessNet_8	0.747±0.039	0.035±0.004	<0.0001	44,336	151.80	10.31	0.009	0.014
LessNet_{12}	0.757±0.040	0.032±0.004	0.749±0.376	96,264	327.45	15.23	0.007	0.014
Diff-LessNet_{12}	0.754±0.039	0.034±0.004	<0.0001	96,264	327.45	15.23	0.010	0.015
LessNet_{16}	0.761±0.039	0.031±0.004	0.742±0.353	168,032	569.59	20.16	0.008	0.014
Diff-LessNet_{16}	0.758±0.039	0.033±0.004	<0.0001	168,032	569.59	20.16	0.012	0.015
LessNet_{24}	0.766±0.039	0.029±0.004	0.852±0.392	370,368	1250	30.00	0.013	0.015
Diff-LessNet_{24}	0.761±0.038	0.032±0.004	<0.0001	370,368	1250	30.00	0.017	0.015
LessNet_{32}	0.768±0.039	0.028±0.004	0.830±0.397	653,504	2200	39.84	0.015	0.015
Diff-LessNet_{32}	0.762±0.038	0.032±0.004	<0.0001	653,504	2200	39.84	0.019	0.015

TABLE VII: Comparison of registration performance between different methods on the 3D IXI dataset.

Methods	Dice↑	RMSE	$\lvert J\rvert_{<0}\%$	Parameters	Mult-Adds (G)	Memory (MB)	CPU (s)	GPU (s)
Affine	0.386±0.195	0.089±0.004	-	-	-	-	-	
SyN [66]	0.645±0.152	-	<0.0001	-	-	-	-	
NiftyReg [67]	0.645±0.167	-	<0.0001	-	-	-	-	
LDDMM [7]	0.680±0.135	-	<0.0001	-	-	-	-	
Flash [33]	0.692±0.140	-	0.0±0.0	-	-	-	1760	-
deedsBCV [68]	0.733±0.126	-	0.147±0.050	-	-	-	-	
VoxelMorph-1 [32]	0.728±0.129	0.048±0.005	1.590±0.339	274,387	304.05	2999.88	2.075	0.398
VoxelMorph-2 [32]	0.732±0.123	0.047±0.005	1.522±0.336	301,411	398.81	3892.38	2.321	0.408
Diff-VoxelMorph [40]	0.580±0.165	0.040±0.003	<0.0001	307,878	89.67	1464.26	1.422	0.398
TransMorph [30]	0.754±0.124	0.043±0.005	1.579±0.328	46,771,251	657.64	4090.31	4.094	0.516
Diff-TransMorph [30]	0.594±0.163	0.038±0.002	<0.0001	46,557,414	252.61	1033.18	2.797	0.419
Diff-B-Spline-TM [30]	0.761±0.122	0.049±0.006	<0.0001	46,806,307	425.95	1563.41	7.582	0.417
LKU-Net [31]	0.765±0.129	0.055±0.006	0.109±0.054	2,086,342	272.09	8713.36	2.304	0.398
Diff-LKU-Net [31]	0.760±0.132	0.058±0.006	0.0±0.0	2,086,342	272.09	8713.36	5.914	0.390
Diff-B-Spline [35]	0.742±0.128	0.055±0.005	<0.0001	266,387	47.05	1233.23	5.649	0.378
Fourier-Net [36]	0.763±0.129	0.058±0.006	0.024±0.019	4,198,352	169.07	4802.93	1.029	0.384
Diff-Fourier-Net [36]	0.761±0.131	0.059±0.006	0.0±0.0	4,198,352	169.07	4802.93	4.668	0.384
LessNet_8	0.757±0.131	0.055±0.006	0.083±0.045	126,904	60.26	1801.41	1.235	0.377
Diff-LessNet_8	0.760±0.129	0.053±0.006	0.0±0.0	126,904	60.26	1801.41	4.771	0.378
LessNet_{12}	0.762±0.130	0.054±0.006	0.131±0.065	275,796	127.97	2623.36	1.715	0.378
Diff-LessNet_{12}	0.765±0.128	0.053±0.006	0.0±0.0	275,796	127.97	2623.36	5.319	0.385
LessNet_{16}	0.765±0.129	0.053±0.006	0.148±0.076	481,648	220.74	3445.31	2.124	0.385
Diff-LessNet_{16}	0.767±0.128	0.052±0.006	0.0±0.0	481,648	220.74	3445.31	5.690	0.387
LessNet_{24}	0.768±0.130	0.052±0.006	0.219±0.101	1,064,232	481.43	5089.22	3.511	0.386
Diff-LessNet_{24}	0.768±0.127	0.050±0.006	0.0±0.0	1,064,232	481.43	5089.22	7.054	0.391
LessNet_{32}	0.768±0.126	0.051±0.006	0.214±0.103	1,874,656	842.35	6733.12	4.411	0.390
Diff-LessNet_{32}	0.768±0.128	0.050±0.006	0.0±0.0	1,874,656	842.35	6733.12	7.951	0.398

our previous publications [31], [36], [46], as we used the exact same experimental settings on the two datasets.

- The results of SyN [66], NiftyReg [67], LDDMM [7], deedsBCV [68] on the 3D IXI dataset were from [30].

2) Subject-to-Subject Registration: First, we note that we defined different versions of LessNet by appending a subscript denoting the parameter C. For example, LessNet_8 represents the architecture configured with C=8. As shown in Table VI, with similar parameters and memory usage, our LessNet_{12} has achieved comparable results to VoxelMorph-1 and VoxelMorph-2, using only 60.78% and 47.44% of their Mult-Adds. Furthermore, our LessNet_{32} achieved comparable results with TransMorph while using only 2.11% of its parameters and 73.33% of its Mult-Adds. Moreover, our Diff-LessNet_{12} outperformed both Diff-VoxelMorph and Diff-TransMorph by a large margin. These results show the superiority of LessNet over other encoder-decoder-style networks.

On the other hand, our Diff-LessNet_8 is faster in CPU runtime and achieved a 1% higher Dice score than Diff-B-Spline. Additionally, both our LessNet_{16} and Diff-LessNet_{16} outperformed Fourier-Net and Diff-Fourier-Net in terms of Dice and CPU runtime, further demonstrating its efficiency.

3) Atlas-to-Subject Registration: Compared with encoder-decoder style networks that estimate dense full-resolution displacement fields, such as VoxelMorph-1, VoxelMorph-2,

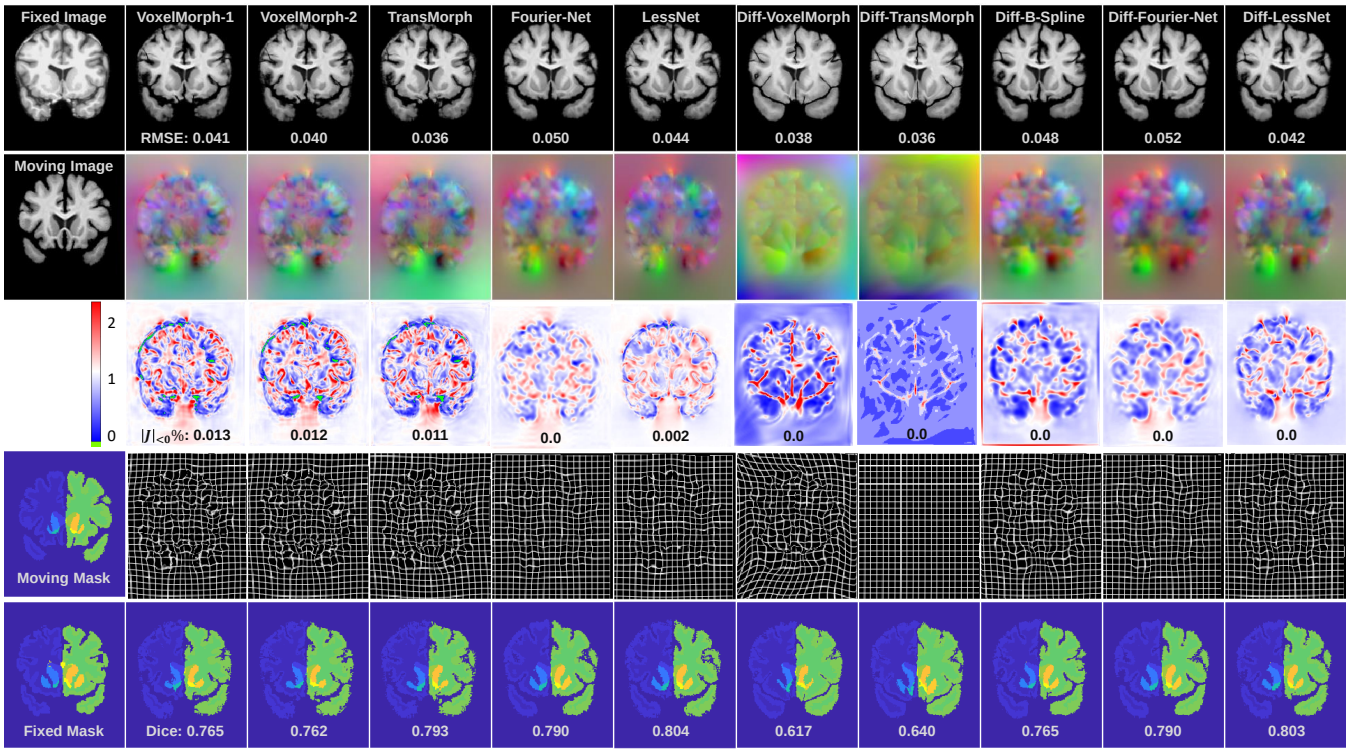

Fig. 5: Comparison of registration performance qualitatively. From top to bottom (apart from 1st column) are warped images, displacement fields, $|J|$ maps, deformation grids, and warped moving masks. The color bar of the $|J|$ maps is illustrated on the left side (with folding areas highlighted green).

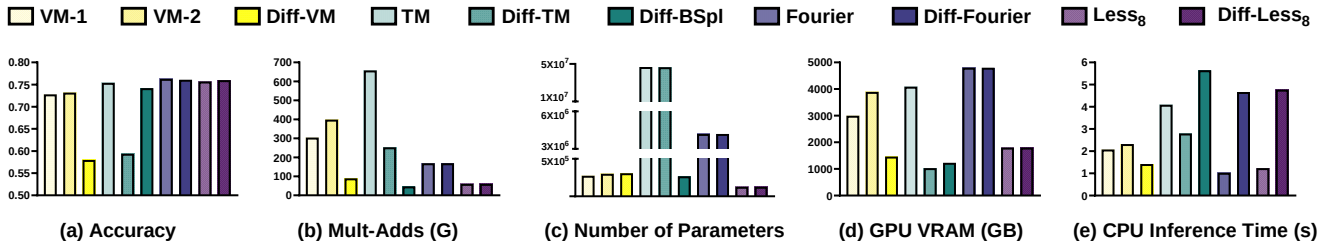

Fig. 6: Comparison between 10 different methods on various metrics such as registration accuracy, GPU memory usage, etc.

TransMorph, and LKU-Net, our LessNet achieved similar registration accuracy with lower computational cost and memory usage. Compared with model-driven networks that estimate low-dimensional representations of displacement fields, Fourier-Net+ and Diff-B-Spline demonstrated greater computational efficiency and lower memory usage than our LessNet. However, both our LessNet and Diff-LessNet exhibited comparable registration accuracy to these methods. To provide a comprehensive comparison, we have presented the qualitative registration results and quantitative plots of various methods in Figs. 5 and 6.

The Appendix includes further results on the ACDC cardiac ES-ED registration, CHAOS abdomen registration, and multi-site, multi-study brain registration.

G. Discussion

1) Is the finding in Table I applicable to other datasets?: To investigate this, we expanded the experiments in Table I to four other tasks. In all four tasks, the decoder-only VoxelMorph achieves comparable registration performance to the fully learned VoxelMorph, suggesting that this finding is consistent across multiple datasets. Detailed numerical results are provided in Table X. Note that the number of learnable parameters in the decoder-only VoxelMorph is significantly fewer than in the fully learned VoxelMorph.

2) Do the decoder-only and full-learned methods in Table I exhibit the same generalization capabilities?: To evaluate the generalization capabilities of the methods listed in Table I, we directly applied the trained brain registration models to the ACDC cardiac ES-ED registration task. The results are presented in Table VIII. The performance of models trained on the ACDC dataset is shown in the 'Train_ACDC Test_ACDC' column. Notably, both the fully-learned VoxelMorph and the decoder-only VoxelMorph (bottom four rows) achieved comparable registration performance on the ACDC dataset. The results for models trained on the OASIS dataset are presented under the 'Train_OASIS Test_ACDC' column. We observe:

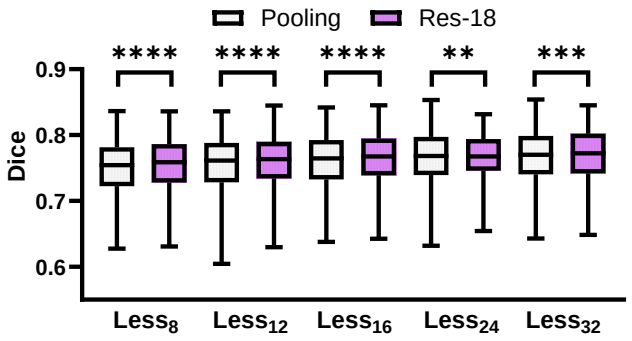

Fig. 7: Replacing the pooling features with pre-trained ResNet-18 features. The first three p-values are all less than 0.0001, while the last two are 0.0088 and 0.0003, respectively.

1) minimal performance difference between the fully-learned and decoder-only VoxelMorph models, and 2) marginal performance gaps between models trained on ACDC and those trained on OASIS, indicating the generalization capabilities of the decoder-only methods in Table I.

3) Comparison to an 'Encoder-Only' Network: The Slicer Network [69] incorporates a feature encoding module (neural backbones) and a slicer module (utilizing a learnable bilateral grid) to progressively refine and upsample feature maps through a splatting-blurring-slicing process. We train and evaluate the 3D Slicer Network on the 3D IXI dataset and found that 1) the Slicer Network can achieve a comparable Dice score of 0.752±0.129 with a zero $|J|_{<0}\%$, while 2) the current official implementation of Slicer Network is memory-hungry, it requires nearly all VRAM of an Nvidia A100-40GB GPU during training, and 3) the training/inference time of Slicer Network is slower than our LessNet due to the multi-scale splatting-blurring-slicing process, specifically, the training time of each epoch of the Slicer network takes more than 5 times of our largest LessNet_{32}.

4) Can the pre-trained CNN features offer further improvements compared to using only pooling features?: We use the classic ImageNet pre-trained ResNet-18 as a feature extractor to validate whether a pre-trained CNN encoder can further enhance performance. Our experiments are conducted on the OASIS inter-subject registration experiments, for each 160×192 image, we extract three sets of features, i.e., 64×80×96, 64×40×48, and 128×20×24 features. We note that the parameters of ResNet-18 are frozen. To ensure a fair comparison with our previous experiments, the moving and fixed features are transformed into 6 channels to match the dimensions of the previous pooling features. We report the registration performance with the Wilcoxon signed rank test in Figure 7. Replacing the pooling features consistently improves Dice scores, with statistically significant improvements across all compared models.

5) Limitations and Applications of LessNet: LessNet includes a tuned parameter C which controls the model size. Using a smaller C may result in suboptimal registration performance, while a larger C will boost the registration performance but significantly increase computational costs. Therefore, selecting an appropriate C for different registration tasks is an area worth exploring. Additionally, we have demonstrated the feasibility of a decoder-only model using simple pooling and ResNet-18 features. Investigating how to select representative features to enhance overall registration performance in LessNet would also be an interesting direction for future research.

TABLE VIII: Registration performance of models trained and tested on different datasets, the 'Train_ACDC Test_ACDC' column shows results for models trained and tested on ACDC, while the 'Train_OASIS Test_ACDC' column presents results for models trained on OASIS and tested on ACDC.

Methods	Encoder	Decoder	Train_ACDC Test_ACDC	Train_OASIS Test_ACDC
VM-1	✓	✗	0.806±0.074	0.792±0.079
VM-2	✓	✗	0.809±0.071	0.788±0.079
VM-1	✗	✓	0.853±0.054	0.840±0.073
VM-2	✗	✓	0.854±0.057	0.846±0.071
VM-1	✓	✓	0.849±0.058	0.848±0.077
VM-2	✓	✓	0.853±0.053	0.849±0.072

V. Conclusion

In the context of unsupervised medical image registration using deep learning, there is a growing trend that larger encoder-decoder networks are proposed in pursuit of improved performance. We however identified redundancy within the encoder of the classical VoxelMorph. Building upon this insight, we introduced LessNet to prove that a learnable decoder alone can suffice for image registration. While we envision that LessNet can be extended to incorporate other handcrafted or pre-trained features, our current proof of concept focuses on multi-scale pooling features. Our overall message from this paper is that for certain registration tasks, excessively large neural networks may not be imperative. Instead, the emphasis could be on designing more compact and efficient networks.

Appendix

A. More Results From The Cardiac, Abdomen, and Multi-Site, Multi-Study Brain Registration

We report the registration performance between different methods on the ACDC, CHAOS, and Multi-Site, Multi-Study datasets in Table IX. On the ACDC dataset, most of the compared methods, such as LKU-Net and Fourier-Net, are outperformed by VoxelMorph-1 and VoxelMorph-2, possibly due to the limited number of training images. Nevertheless, the registration performance of our LessNet consistently improves as the model size increases from LessNet_8 to LessNet_{32}. Even with LessNet_8, we achieve comparable registration performance to VoxelMorph. With a larger model, (Diff-)LessNet_{16} can outperform TransMorph.

On the Multi-Site, Multi-Study dataset, our LessNet_{12} achieves results comparable to both VoxelMorph-1 and VoxelMorph-2. LessNet_{24} outperforms most of the compared methods, including LKU-Net and Diff-B-Spline-TransMorph. However, LessNet_{32} falls short by 0.2% in Dice score compared to the best-performing TransMorph.

On the CHAOS dataset, it is clear that our Diff-LessNet_8 outperformed VoxelMorph-1, VoxelMorph-2, and

TABLE IX: Comparison of registration performance between different methods on the ACDC, CHAOS, and Multi-Site, Multi-Study datasets.

Methods	ACDC			Multi-Site Multi-Study			CHAOS		
	Dice↑	RMSE	$\lvert J\rvert_{<0}\%$	Dice↑	RMSE	$\lvert J\rvert_{<0}\%$	Dice↑	RMSE	$\lvert J\rvert_{<0}\%$
Initial	0.644±0.112	0.079±0.024	–	0.336±0.052	0.195±0.010	–	0.454±0.120	0.094±0.021	–
VoxelMorph-1	0.849±0.058	0.053±0.018	0.296±0.283	0.544±0.081	0.081±0.013	1.097±0.199	0.633±0.149	0.057±0.015	0.854±0.768
VoxelMorph-2	0.853±0.053	0.049±0.017	0.442±0.387	0.555±0.082	0.077±0.013	1.087±0.210	0.632±0.148	0.056±0.014	0.634±0.609
Diff-VoxelMorph	0.832±0.059	0.055±0.019	0.0	0.480±0.072	0.115±0.011	<0.0001	0.631±0.144	0.063±0.016	0.0
TransMorph	0.860±0.052	0.052±0.017	0.340±0.304	0.589±0.086	0.072±0.014	0.823±0.211	0.658±0.151	0.054±0.013	0.603±0.374
Diff-TransMorph	0.845±0.053	0.056±0.018	0.002±0.009	0.544±0.080	0.097±0.013	<0.0004	0.655±0.148	0.060±0.015	<0.0001
Diff-B-Spline-TM	0.852±0.053	0.057±0.018	0.0	0.566±0.083	0.090±0.013	0.0	0.670±0.148	0.056±0.011	<0.0005
LKU-Net	0.833±0.057	0.058±0.019	0.298±0.315	0.573±0.084	0.075±0.013	0.874±0.192	0.646±0.155	0.056±0.013	0.585±0.614
Diff-LKU-Net	0.828±0.062	0.057±0.019	0.0	0.555±0.082	0.083±0.013	<0.0001	0.655±0.148	0.057±0.013	<0.0004
Diff-B-Spline	0.804±0.070	0.062±0.020	0.0	0.501±0.074	0.110±0.011	0.0	0.636±0.149	0.065±0.023	0.008±0.017
Fourier-Net	0.833±0.056	0.063±0.020	0.454±0.452	0.535±0.079	0.098±0.012	0.473±0.138	0.655±0.149	0.058±0.013	0.744±0.722
Diff-Fourier-Net	0.826±0.061	0.062±0.020	0.0	0.531±0.078	0.099±0.011	0.0	0.657±0.149	0.057±0.011	0.001+-0.004
LessNet_8	0.853±0.052	0.052±0.018	0.052±0.018	0.528±0.078	0.087±0.012	1.423±0.222	0.627±0.147	0.057±0.013	0.709±0.594
Diff-LessNet_8	0.857±0.048	0.050±0.016	0.0	0.538±0.079	0.087±0.012	<0.0001	0.645±0.146	0.056±0.011	0.002±0.007
LessNet_{12}	0.857±0.050	0.050±0.017	0.503±0.459	0.557±0.082	0.079±0.013	1.097±0.221	0.637±0.148	0.575±0.215	0.655±0.603
Diff-LessNet_{12}	0.859±0.047	0.048±0.016	0.0	0.548±0.081	0.084±0.012	<0.0001	0.651±0.147	0.055±0.011	0.001±0.006
LessNet_{16}	0.861±0.048	0.048±0.017	0.484±0.483	0.571±0.084	0.076±0.013	0.995±0.204	0.642±0.153	0.054±0.013	0.702±0.644
Diff-LessNet_{16}	0.863±0.047	0.046±0.015	0.0	0.558±0.082	0.080±0.013	<0.0001	0.652±0.150	0.054±0.011	0.002±0.009
LessNet_{24}	0.865±0.039	0.048±0.016	0.439±0.439	0.581±0.085	0.072±0.013	0.977±0.219	0.653±0.154	0.053±0.013	0.673±0.694
Diff-LessNet_{24}	0.865±0.042	0.045±0.015	0.0	0.569±0.084	0.077±0.013	<0.0001	0.661±0.151	0.054±0.011	0.001±0.006
LessNet_{32}	0.864±0.048	0.048±0.017	0.481±0.513	0.587±0.086	0.069±0.014	1.065±0.238	0.658±0.152	0.052±0.013	0.705±0.647
Diff-LessNet_{32}	0.866±0.043	0.045±0.015	<0.0002	0.572±0.084	0.077±0.013	<0.0001	0.673±0.167	0.054±0.015	0.011±0.049

TABLE X: Expanding the experiments in Table I to four other tasks.

Methods	Enc	Dec	IXI		ACDC		CHAOS		Multi-Site Multi-Study	
			Dice↑	$\lvert J\rvert_{<0}\%$	Dice↑	$\lvert J\rvert_{<0}\%$	Dice↑	$\lvert J\rvert_{<0}\%$	Dice↑	$\lvert J\rvert_{<0}\%$
Initial	-	-	0.386±0.195	-	0.644±0.112	-	0.454±0.110	–	0.336±0.052	-
VM-1	✓	✗	0.682±0.153	0.092±0.042	0.806±0.074	0.195±0.246	0.542±0.130	0.373±0.321	0.400±0.060	0.155±0.035
VM-2	✓	✗	0.715±0.139	0.256±0.086	0.809±0.071	0.190±0.215	0.550±0.128	0.521±0.446	0.434±0.065	0.251±0.048
VM-1	✗	✓	0.728±0.135	1.892±0.373	0.853±0.054	0.455±0.405	0.605±0.140	0.806±0.613	0.526±0.078	1.412±0.205
VM-2	✗	✓	0.729±0.137	1.878±0.363	0.854±0.057	0.427±0.404	0.602±0.142	0.766±0.583	0.534±0.079	1.143±0.189
VM-1	✓	✓	0.728±0.129	1.590±0.339	0.849±0.058	0.296±0.283	0.633±0.149	0.853±0.768	0.544±0.081	1.097±0.199
VM-2	✓	✓	0.732±0.123	1.522±0.336	0.853±0.053	0.442±0.397	0.632±0.148	0.634±0.609	0.555±0.082	1.087±0.210

Diff-VoxelMorph in terms of Dice. Our LessNet_{24} and Diff-LessNet_{24} can achieve comparable performance with TransMorph and Diff-TransMorph. The second best-performing method, Diff-B-Spline-TransMorph, can be outperformed by our Diff-LessNet_{32}.

B. More Results for 'Redundancy in Encoder'

We have extended the experiments in Table I into the rest four tasks, as reported in Table X.

- On the IXI dataset, the decoder-only VoxelMorph-1 and VoxelMorph-2 achieve comparable Dice scores to their fully-learned counterparts.
- On the ACDC dataset, the decoder-only VoxelMorph-2 outperforms the fully-learned VoxelMorph-2, achieving a higher Dice score, lower RMSE, and lower $|J|< 0\%$. The decoder-only VoxelMorph-1 achieves a higher Dice score and lower RMSE than the fully-learned VoxelMorph-1, though it exhibits a slight increase in $|J|< 0\%$.
- On the CHAOS dataset, the fully-learned networks perform about 3% higher in Dice score compared to the decoder-only networks.
- On the multi-site, multi-study dataset, the performance gap between the decoder-only and fully-learned networks is about 2% in terms of Dice score.
- We observe that the decoder-only networks often have a higher $|J|_{<0}\%$ than the fully-learned networks. However, there are exceptions, such as VoxelMorph-2 on the ACDC dataset and VoxelMorph-1 on the CHAOS dataset.

C. Statistical Analysis

For Table VI, we compare our method with the three top-performing approaches: TransMorph, LKU-Net, and Diff-B-Spline-TransMorph. The p-value between our LessNet_{32} and TransMorph is 0.6589, indicating no significant difference. However, the p-values between LessNet_{32} and both LKU-Net and Diff-B-Spline-TransMorph are <0.0001, indicating statistically significant improvements. Similarly, in Table VII, we compare LessNet_{32} with the three best-performing methods (excluding ours): LKU-Net, Fourier-Net, and Diff-B-Spline-TransMorph. The p-values for these comparisons are all below 0.0001, further confirming the significance of the improvements achieved by LessNet_{32}.

REFERENCES

[1] R. Marti, R. Zwiggelaar, and C. Rubin, "Automatic mammographic registration: towards the detection of abnormalities," in *S T1U Conference on Medical Image Understanding and Analysis*, 2001, pp. 149–152.

[2] C. Dean, J. Sykes, R. Cooper, P. Hatfield, B. Carey, S. Swift, S. Bacon, D. Thwaites, D. Sebag-Montefiore, and A. Morgan, "An evaluation of four ct–mri co-registration techniques for radiotherapy treatment planning of prone rectal cancer patients," *The British journal of radiology*, vol. 85, no. 1009, pp. 61–68, 2012.

[3] D. L. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes, "Medical image registration," *Physics in Medicine & Biology*, vol. 46, no. 3, p. R1, 2001.

[4] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE Transactions on Medical Imaging*, vol. 32, no. 7, pp. 1153–1190, 2013.

[5] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: application to breast mr images," *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 712–721, 1999.

[6] D. Rueckert, P. Aljabar, R. A. Heckemann, J. V. Hajnal, and A. Hammers, "Diffeomorphic registration using b-splines," in *International Conference on Medical Image Computing and Computer-Assisted Intervention. Proceedings, Part II 9*. Springer, 2006, pp. 702–709.

[7] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," *International Journal of Computer Vision*, vol. 61, no. 2, pp. 139–157, 2005.

[8] J. Ashburner, "A fast diffeomorphic image registration algorithm," *NeuroImage*, vol. 38, no. 1, pp. 95–113, 2007.

[9] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Diffeomorphic demons: Efficient non-parametric image registration," *NeuroImage*, vol. 45, no. 1, pp. S61–S72, 2009.

[10] M. P. Heinrich, M. Jenkinson, M. Bhushan, T. Matin, F. V. Gleeson, M. Brady, and J. A. Schnabel, "Mind: Modality independent neighbourhood descriptor for multi-modal deformable registration," *Medical Image Analysis*, vol. 16, no. 7, pp. 1423–1435, 2012.

[11] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[12] J. Duan, X. Jia, J. Bartlett, W. Lu, and Z. Qiu, "Arbitrary order total variation for deformable image registration," *Pattern Recognition*, p. 109318, 2023.

[13] G. E. Christensen and H. J. Johnson, "Consistent image registration," *IEEE Transactions on Medical Imaging*, vol. 20, no. 7, pp. 568–582, 2001.

[14] G. E. Christensen, R. D. Rabbitt, M. I. Miller, S. C. Joshi, U. Grenander, T. A. Coogan, and D. C. VANESSEN, "Topological properties of smooth anatomic maps," in *Information processing in medical imaging*, vol. 3, 1995, pp. 101–112.

[15] A. Thorley, X. Jia, H. J. Chang, B. Liu, K. Bunting, V. Stoll, A. de Marvao, D. P. O'Regan, G. Gkoutos, D. Kotecha *et al.*, "Nesterov accelerated admm for fast diffeomorphic image registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, pp. 150–160.

[16] D. Rueckert and J. A. Schnabel, "Model-based and data-driven strategies in medical image computing," *Proceedings of the IEEE*, vol. 108, no. 1, pp. 110–124, Jan 2020.

[17] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9252–9260.

[18] C. Qin, W. Bai, J. Schlemper, S. E. Petersen, S. K. Piechnik, S. Neubauer, and D. Rueckert, "Joint learning of motion estimation and segmentation for cardiac mr image sequences," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 472–480.

[19] H. Qiu, C. Qin, L. Le Folgoc, B. Hou, J. Schlemper, and D. Rueckert, "Deep learning for cardiac motion estimation: supervised vs. unsupervised training," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2019, pp. 186–194.

[20] X. Hu, M. Kang, W. Huang, M. R. Scott, R. Wiest, and M. Reyes, "Dual-stream pyramid registration network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 382–390.

[21] M. Kang, X. Hu, W. Huang, M. R. Scott, and M. Reyes, "Dual-stream pyramid registration network," *Medical Image Analysis*, vol. 78, p. 102379, 2022.

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

[23] S. Zhao, Y. Dong, E. I.-C. Chang, and Y. Xu, "Recursive cascaded networks for unsupervised medical image registration," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[24] T. C. Mok and A. C. Chung, "Large deformation diffeomorphic image registration with laplacian pyramid networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention, Proceedings, Part III 23*. Springer, 2020, pp. 211–221.

[25] X. Jia, A. Thorley, W. Chen, H. Qiu, L. Shen, I. B. Styles, H. J. Chang, A. Leonardis, A. De Marvao, D. P. O'Regan *et al.*, "Learning a model-driven variational network for deformable image registration," *IEEE Transactions on Medical Imaging*, vol. 41, no. 1, pp. 199–212, 2021.

[26] R. Liu, Z. Li, X. Fan, C. Zhao, H. Huang, and Z. Luo, "Learning deformable image registration from optimization: perspective, modules, bilevel training and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7688–7704, 2021.

[27] T. Ma, X. Dai, S. Zhang, and Y. Wen, "Pivit: Large deformation image registration with pyramid-iterative vision transformer," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2023, pp. 602–612.

[28] T. Ma, S. Zhang, J. Li, and Y. Wen, "Iirp-net: iterative inference residual pyramid network for enhanced image registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 11 546–11 555.

[29] J. Chen, Y. He, E. C. Frey, Y. Li, and Y. Du, "Vit-v-net: Vision transformer for unsupervised volumetric medical image registration," *arXiv preprint arXiv:2104.06468*, 2021.

[30] J. Chen, E. C. Frey, Y. He, W. P. Segars, Y. Li, and Y. Du, "Transmorph: Transformer for unsupervised medical image registration," *Medical Image Analysis*, vol. 82, p. 102615, 2022.

[31] X. Jia, J. Bartlett, T. Zhang, W. Lu, Z. Qiu, and J. Duan, "U-net vs transformer: Is u-net outdated in medical image registration?" *arXiv preprint arXiv:2208.04939*, 2022.

[32] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "Voxelmorph: A learning framework for deformable medical image registration," *IEEE Transactions on Medical Imaging*, vol. 38, no. 8, pp. 1788–1800, Aug 2019.

[33] M. Zhang and P. T. Fletcher, "Fast diffeomorphic image registration via fourier-approximated lie algebras," *International Journal of Computer Vision*, vol. 127, no. 1, pp. 61–73, 2019.

[34] 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," *Medical Image Analysis*, vol. 52, pp. 128–143, 2019.

[35] H. Qiu, C. Qin, A. Schuh, K. Hammernik, and D. Rueckert, "Learning diffeomorphic and modality-invariant registration using b-splines," in *Medical Imaging with Deep Learning*, 2021.

[36] X. Jia, J. Bartlett, W. Chen, S. Song, T. Zhang, X. Cheng, W. Lu, Z. Qiu, and J. Duan, "Fourier-net: Fast image registration with band-limited deformation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 1015–1023.

[37] H. Li and Y. Fan, "Non-rigid image registration using self-supervised fully convolutional networks without training data," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, 2018, pp. 1075–1078.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[39] J. Zhang, "Inverse-consistent deep networks for unsupervised deformable image registration," *arXiv preprint arXiv:1809.03443*, 2018.

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

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

[42] T. C. Mok and A. C. Chung, "Fast symmetric diffeomorphic image registration with convolutional neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[43] B. Kim, D. H. Kim, S. H. Park, J. Kim, J.-G. Lee, and J. C. Ye, "Cyclemorph: cycle consistent unsupervised deformable image registration," *Medical Image Analysis*, vol. 71, p. 102036, 2021.

[44] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[45] 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 *DLMIA 2017, and ML-CDS 2017*. Springer, 2017, pp. 204–212.

[46] X. Jia, A. Thorley, A. Gomez, W. Lu, D. Kotecha, and J. Duan, "Fourier-net+: Leveraging band-limited representation for efficient 3d medical image registration," *arXiv preprint arXiv:2307.02997*, 2023.

[47] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults," *Journal of Cognitive Neuroscience*, vol. 19, no. 9, pp. 1498–1507, 2007.

[48] A. Hoopes, M. Hoffmann, B. Fischl, J. Guttag, and A. V. Dalca, "Hypermorph: Amortized hyperparameter learning for image registra-

tion," in *Information Processing in Medical Imaging: 27th International Conference, IPMI 2021, Proceedings 27*. Springer, 2021, pp. 3–17.
[49] A. Hoopes, M. Hoffmann, D. N. Greve, B. Fischl, J. Guttag, and A. V. Dalca, "Learning the effect of registration hyperparameters with hypermorph," *The journal of machine learning for biomedical imaging*, vol. 1, 2022.
[50] O. Bernard, A. Lalande, C. Zotti, F. Cervenansky, X. Yang, P.-A. Heng, I. Cetin, K. Lekadir, O. Camara, M. A. G. Ballester *et al.*, "Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved?" *IEEE Transactions on Medical Imaging*, vol. 37, no. 11, pp. 2514–2525, 2018.
[51] A. E. Kavur, N. S. Gezer, M. Barış, S. Aslan, P.-H. Conze, V. Groza, D. D. Pham, S. Chatterjee, P. Ernst, S. Özkan *et al.*, "Chaos challenge-combined (ct-mr) healthy abdominal organ segmentation," *Medical Image Analysis*, vol. 69, p. 101950, 2021.
[52] B. Dufumier, A. Grigis, J. Victor, C. Ambroise, V. Frouin, and E. Duchesnay, "Openbhb: a large-scale multi-site brain mri data-set for age prediction and debiasing," *NeuroImage*, vol. 263, p. 119637, 2022.
[53] A. Klein and J. Tourville, "101 labeled brain images and a consistent human cortical labeling protocol," *Frontiers in neuroscience*, vol. 6, p. 171, 2012.
[54] X.-N. Zuo, J. S. Anderson, P. Bellec, R. M. Birn, B. B. Biswal, J. Blautzik, J. Breitner, R. L. Buckner, V. D. Calhoun, F. X. Castellanos *et al.*, "An open science resource for establishing reliability and reproducibility in functional connectomics," *Scientific data*, vol. 1, no. 1, pp. 1–13, 2014.
[55] R. L. Buckner, J. L. Roffman, and J. W. Smoller, "Brain genomics superstruct project (gsp)," 01 2014. [Online]. Available: https://cir.nii.ac.jp/crid/1881991017814738176
[56] D. P. Orfanos, V. Michel, Y. Schwartz, P. Pinel, A. Moreno, D. Le Bihan, and V. Frouin, "The brainomics/localizer database," *NeuroImage*, vol. 144, pp. 309–314, 2017.
[57] A. Babayan, M. Erbey, D. Kumral, J. D. Reinelt, A. M. Reiter, J. Röbbig, H. L. Schaare, M. Uhlig, A. Anwander, P.-L. Bazin *et al.*, "A mind-brain-body dataset of mri, eeg, cognition, emotion, and peripheral physiology in young and old adults," *Scientific data*, vol. 6, no. 1, pp. 1–21, 2019.
[58] S. A. Nastase, Y.-F. Liu, H. Hillman, A. Zadbood, L. Hasenfratz, N. Keshavarzian, J. Chen, C. J. Honey, Y. Yeshurun, M. Regev *et al.*, "The "narratives" fmri dataset for evaluating models of naturalistic language comprehension," *Scientific data*, vol. 8, no. 1, p. 250, 2021.
[59] A. Sunavsky and J. Poppenk, "Neuroimaging predictors of creativity in healthy adults," *Neuroimage*, vol. 206, p. 116292, 2020.
[60] D. J. Follmer, S.-Y. Fang, R. B. Clariana, B. J. Meyer, and P. Li, "What predicts adult readers' understanding of stem texts?" *Reading and Writing*, vol. 31, pp. 185–214, 2018.
[61] P. Li and R. B. Clariana, "Reading comprehension in l1 and l2: An integrative approach," *Journal of Neurolinguistics*, vol. 50, pp. 94–105, 2019.
[62] D. Kuang and T. Schmah, "Faim–a convnet method for unsupervised 3d medical image registration," in *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 10*. Springer, 2019, pp. 646–654.
[63] L. Liu, A. I. Aviles-Rivero, and C.-B. Schönlieb, "Contrastive registration for unsupervised medical image segmentation," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
[64] X. Jia, X. Cheng, J. Duan, and B. W. Papież, "A naive trick to accelerate training of lncc-based deep image registration models," *Preprints*, February 2025. [Online]. Available: https://doi.org/10.20944/preprints202502.2200.v1
[65] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International journal of computer vision*, vol. 92, pp. 1–31, 2011.
[66] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, "A reproducible evaluation of ants similarity metric performance in brain image registration," *NeuroImage*, vol. 54, no. 3, pp. 2033–2044, 2011.
[67] M. Modat, G. R. Ridgway, Z. A. Taylor, M. Lehmann, J. Barnes, D. J. Hawkes, N. C. Fox, and S. Ourselin, "Fast free-form deformation using graphics processing units," *Computer Methods and Programs in Biomedicine*, vol. 98, no. 3, pp. 278–284, 2010.
[68] M. P. Heinrich, O. Maier, and H. Handels, "Multi-modal multi-atlas segmentation using discrete optimisation and self-similarities." *VISCERAL Challenge@ ISBI*, vol. 1390, p. 27, 2015.
[69] H. Zhang, X. Chen, R. Wang, R. Hu, D. Liu, and G. Li, "Slicer networks," *arXiv preprint arXiv:2401.09833*, 2024.