



TransMorph: Transformer for unsupervised medical image registration

Junyu Chen ^{a,b,*}, Eric C. Frey ^{a,b}, Yufan He ^e, William P. Segars ^c, Ye Li ^d, Yong Du ^a

^a Russell H. Morgan Department of Radiology and Radiological Science, Johns Hopkins Medical Institutes, Baltimore, MD, USA

^b Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, USA

^c Carl E. Ravin Advanced Imaging Laboratories, Department of Radiology, Duke University Medical Center, Durham, NC, USA

^d Center for Advanced Medical Computing and Analysis, Massachusetts General Hospital and Harvard Medical School, Boston, MA, USA

^e NVIDIA Corporation, Bethesda, MD, USA

ARTICLE INFO

Keywords:

Image registration

Deep learning

Vision transformer

Computerized phantom

ABSTRACT

In the last decade, convolutional neural networks (ConvNets) have been a major focus of research in medical image analysis. However, the performances of ConvNets may be limited by a lack of explicit consideration of the long-range spatial relationships in an image. Recently, Vision Transformer architectures have been proposed to address the shortcomings of ConvNets and have produced state-of-the-art performances in many medical imaging applications. Transformers may be a strong candidate for image registration because their substantially larger receptive field enables a more precise comprehension of the spatial correspondence between moving and fixed images. Here, we present TransMorph, a hybrid Transformer-ConvNet model for volumetric medical image registration. This paper also presents diffeomorphic and Bayesian variants of TransMorph: the diffeomorphic variants ensure the topology-preserving deformations, and the Bayesian variant produces a well-calibrated registration uncertainty estimate. We extensively validated the proposed models using 3D medical images from three applications: inter-patient and atlas-to-patient brain MRI registration and phantom-to-CT registration. The proposed models are evaluated in comparison to a variety of existing registration methods and Transformer architectures. Qualitative and quantitative results demonstrate that the proposed Transformer-based model leads to a substantial performance improvement over the baseline methods, confirming the effectiveness of Transformers for medical image registration.

1. Introduction

Deformable image registration (DIR) is fundamental for many medical imaging analysis tasks. It functions by establishing spatial correspondence in order to minimize the differences between a pair of fixed and moving images. Traditional methods formulate image registration as a variational problem for estimating a smooth mapping between the points in one image and those in another (Avants et al., 2008; Beg et al., 2005; Vercauteren et al., 2009; Heinrich et al., 2013a; Modat et al., 2010). However, such methods are computationally expensive and usually slow in practice because the optimization problem needs to be solved de novo for each pair of unseen images.

Recently, deep neural networks (DNNs), especially convolutional neural networks (ConvNets), have demonstrated state-of-the-art performance in many computer vision tasks, including object detection (Redmon et al., 2016), image classification (He et al., 2016), and segmentation (Long et al., 2015). Ever since the success of U-Net in the ISBI cell tracking challenge of 2015 (Ronneberger et al., 2015), ConvNet-based

methods have become a major focus of attention in medical image analysis fields, such as tumor segmentation (Isensee et al., 2021; Zhou et al., 2019), image reconstruction (Zhu et al., 2018), and disease diagnostics (Lian et al., 2018). In medical image registration, ConvNet-based methods can produce significantly improved registration performance while operating orders of magnitudes faster (after training) compared to traditional methods. ConvNet-based methods replace the costly per-image optimization seen in traditional methods with a single global function optimization during a training phase. The ConvNets learn the common representation of image registration from training images, enabling rapid alignment of an unseen image pair after training. Initially, the supervision of ground-truth deformation fields (which are usually generated using traditional registration methods) is needed for training the neural networks (Onofrey et al., 2013; Yang et al., 2017b; Rohé et al., 2017). Recently, the focus has been shifted towards developing unsupervised methods that do not depend on ground-truth deformation fields (Balakrishnan et al., 2019; Dalca et al., 2019; Kim

* Corresponding author at: Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, USA.

E-mail addresses: jchen245@jhmi.edu (J. Chen), efrey@jhmi.edu (E.C. Frey), yufanh@nvidia.com (Y. He), paul.segars@duke.edu (W.P. Segars), gary.li@mgh.harvard.edu (Y. Li), duyong@jhmi.edu (Y. Du).

et al., 2021; de Vos et al., 2019, 2017; Lei et al., 2020; Chen et al., 2020; Zhang, 2018). Nearly all of the existing deep-learning-based methods mentioned above used U-Net (Ronneberger et al., 2015) or the simply modified versions of U-Net (e.g., tweaking the number of layers or changing down- and up-sampling schemes) as their ConvNet designs.

ConvNet architectures generally have limitations in modeling explicit long-range spatial relations (i.e., relations between two voxels that are far away from each other) present in an image due to the intrinsic locality (i.e., the limited receptive field) of convolution operations (Luo et al., 2016). The U-Net (or V-Net (Milletari et al., 2016)) was proposed to overcome this limitation by introducing down- and up-sampling operations into a ConvNet, which theoretically enlarges the receptive field of the ConvNet and, thus, encourages the network to consider long-range relationships between points in images. However, several problems remain: first, the receptive fields of the first several layers are still restricted by the convolution-kernel size, and the global information of an image can only be viewed at the deeper layers of the network; second, it has been shown that as the convolutional layers deepen, the impact from far-away voxels decays quickly (Li et al., 2021). Therefore, the effective receptive field of a U-Net is, in practice, much smaller than its theoretical receptive field, and it is only a portion of the typical size of a medical image. This limits the U-Net's ability to perceive semantic information and model long-range relationships between points. Yet, it is believed that the ability to comprehend semantic scene information is of great importance in coping large deformations (Ha et al., 2020). Many works in other fields (e.g., image segmentation) have addressed this limitation of U-Net (Zhou et al., 2019; Jha et al., 2019; Devalla et al., 2018; Alom et al., 2018). To allow for a better flow of multi-scale contextual information throughout the network, Zhou et al. (2019) proposed a nested U-Net (i.e., U-Net++), in which the complex up- and down-samplings along with multiple skip connections were used. Devalla et al. (2018) introduced dilated convolution to the U-Net architecture that enlarges the network's effective receptive field. A similar idea was proposed by Alom et al. (2018), where the network's effective receptive field was increased by deploying recurrent convolutional operations. Jha et al. proposed ResUNet++ (Jha et al., 2019) that incorporates the attention mechanisms into U-Net for modeling long-range spatial information. Despite these methods' promising performance in other medical imaging fields, there has been limiting work on using advanced network architectures for medical image registration.

Transformer, which originated from natural language processing tasks (Vaswani et al., 2017), has shown its potential in computer vision tasks. A Transformer deploys self-attention mechanisms to determine which parts of the input sequence (e.g., an image) are essential based on contextual information. Unlike convolution operations, whose effective receptive fields are limited by the size of convolution kernels, the self-attention mechanisms in a Transformer have large size effective receptive fields, making a Transformer capable of capturing long-range spatial information (Li et al., 2021). Dosovitskiy et al. (2020) proposed Vision Transformer (ViT) that applies the Transformer encoder from NLP directly to images. It was the first purely self-attention-based network for computer vision and achieved state-of-the-art performance in image recognition. Subsequent to their success, Swin Transformer (Liu et al., 2021a) and its variants (Dai et al., 2021; Dong et al., 2021) have demonstrated their superior performances in object detection, and semantic segmentation. Recently, Transformer-related methods have gained increased attention in medical imaging (Chen et al., 2021b; Xie et al., 2021; Wang et al., 2021a; Li et al., 2021; Wang et al., 2021b; Zhang et al., 2021); the major application has been the task of image segmentation.

Transformer can be a strong candidate for image registration because it can better comprehend the spatial correspondence between the moving and fixed images. Registration is the process of establishing such correspondence, and intuitively, by comparing different parts of the moving to the fixed image. A ConvNet has a narrow field of view: it

performs convolution locally, and its field of view grows in proportion to the ConvNet's depth; hence, the shallow layers have a relatively small receptive field, limiting the ConvNet's ability to associate the distant parts between two images. For example, if the left part of the moving image matches the right part of the fixed image, ConvNet will be unable to establish the proper spatial correspondence between the two parts if it cannot see both parts concurrently (i.e., when one of the parts falls outside of the ConvNet's field of view). However, Transformer is capable of handling such circumstances and rapidly focusing on the parts that need deformation, owing to its large receptive field and self-attention mechanism.

Our group has previously shown preliminary results that demonstrated the bridging of ViT and V-Net provided good performance in image registration (Chen et al., 2021a). In this work, we extended that preliminary work and investigated various Transformer models from other tasks (i.e., computer vision and medical imaging tasks). We present a hybrid Transformer-ConvNet framework, TransMorph, for volumetric medical image registration. In this method, the Swin Transformer (Liu et al., 2021a) was employed as the encoder to capture the spatial correspondence between the input moving and fixed images. Then, a ConvNet decoder processed the information provided by the Transformer encoder into a dense displacement field. Long skip connections were deployed to maintain the flow of localization information between the encoder and decoder stages. We also introduced diffeomorphic variations of TransMorph to ensure a smooth and topology-preserving deformation. Additionally, we applied variational inference on the parameters of TransMorph, resulting in a Bayesian model that predicts registration uncertainty based on the given image pair. Qualitative and quantitative evaluation of the experimental results demonstrate the robustness of the proposed method and confirm the efficacy of Transformers for image registration.

The main contributions of this work are summarized as follows:

- *Transformer-based model*: This paper presents the pioneering work on using Transformers for image registration. A novel Transformer-based neural network, TransMorph, was proposed for affine and deformable image registration.
- *Architecture analysis*: Experiments in this paper demonstrate that positional embedding, which is a commonly used element in Transformer by convention, is not required for the proposed hybrid Transformer-ConvNet model. Secondly, we show that Transformer-based models have larger effective receptive fields than ConvNets. Moreover, we demonstrated that TransMorph promotes a flatter registration loss landscape.
- *Diffeomorphic registration*: We demonstrate that TransMorph can be easily integrated into two existing frameworks as a registration backbone to provide diffeomorphic registration.
- *Uncertainty quantification*: This paper also provides a Bayesian uncertainty variant of TransMorph that yields transformer uncertainty and perfectly calibrated appearance uncertainty estimates.
- *State-of-the-art results*: We extensively validate the proposed registration models on two brain MRI registration applications (inter-patient and atlas-to-patient registration) and on a novel application of XCAT-to-CT registration with an aim to create a population of anatomically variable XCAT phantom. The datasets used in this study (which include a publicly available dataset, the IXI dataset¹) contain over 1000 image pairs for training and testing. The proposed models were compared with various registration methods and demonstrated state-of-the-art performance. Eight registration approaches were employed as baselines, including learning-based methods and widely used conventional methods. The performances of four recently proposed Transformer architectures from other tasks (e.g., semantic segmentation, classification, etc.) were also evaluated on the task of image registration.

¹ <https://brain-development.org/ixi-dataset/>

- *Open source:* We provide the community with a fast and accurate tool for deformable registration. The source code, the pre-trained models, and our preprocessed IXI dataset are publicly available at <https://bit.ly/37eJS6N>.

The paper is organized as follows. Section 2 discusses related work. Section 3 explains the proposed methodology. Section 4 discusses experimental setup, implementation details, and datasets used in this study. Section 5 presents experimental results. Section 6 discusses the findings based on the results, and Section 7 concludes the paper.

2. Related work

This section reviews the relevant literature and provides fundamental knowledge for the proposed method.

2.1. Image registration

Deformable image registration (DIR) establishes spatial correspondence between two images by optimizing an energy function:

$$E(I_m, I_f, \phi) = E_{sim}(I_m \circ \phi, I_f) + \lambda R(\phi), \quad (1)$$

where I_m and I_f denote, respectively, the moving and fixed image, ϕ denotes the deformation field that warps the moving image (i.e., $I_m \circ \phi$), $R(\phi)$ imposes smoothness of the deformation field, and λ is the regularization hyper-parameter that determines the trade-off between image similarity and deformation field regularity. The optimal warping, $\hat{\phi}$ is given by minimizing this energy function:

$$\hat{\phi} = \arg \min_{\phi} E(I_m, I_f, \phi). \quad (2)$$

In the energy function, E_{sim} measures the level of alignment between the deformed moving image, $I_m \circ \phi$, and the fixed image, I_f . Some common choices for E_{sim} are mean squared error (MSE) (Beg et al., 2005; Wolberg and Zokai, 2000), normalized cross-correlation (NCC) (Avants et al., 2008), structural similarity index (SSIM) (Chen et al., 2020), and mutual information (MI) (Viola and Wells III, 1997). The regularization term, $R(\phi)$, imposes spatial smoothness on the deformation field. A common assumption in most applications is that similar structures exist in both moving and fixed images. As a result, a continuous and invertible deformation field (i.e., a diffeomorphism) is needed to preserve topology, and the regularization, $R(\phi)$ is meant to enforce or encourage this. Isotropic diffusion (equivalent to Gaussian smoothing) (Balakrishnan et al., 2019), anisotropic diffusion (Pace et al., 2013), total variation (Vishnevskiy et al., 2016), and bending energy (Johnson and Christensen, 2002) are popular options for $R(\phi)$.

2.1.1. Image registration via deep neural networks

While traditional image registration methods iteratively minimize the energy function in (1) for each pair of moving and fixed images, DNN-based methods optimize the energy function for a training dataset, thereby learning a global representation of image registration that enables alignment of an unseen pair of volumes. DNN methods are often categorized as supervised or unsupervised, with the former requiring a ground truth deformation field for training and the latter relying only on the image datasets.

In supervised DNN methods, the ground-truth deformation fields are either produced synthetically or generated by traditional registration methods (Yang et al., 2017b; Sokooti et al., 2017; Cao et al., 2018). Yang et al. (2017b) proposed a supervised ConvNet that predicts the LDDMM (Beg et al., 2005) momentum from image patches. Sokooti et al. (2017) trained a registration ConvNet with synthetic displacement fields. The ground-truth deformation fields are often computationally expensive to generate, and the registration accuracy of these methods is highly dependent on the quality of the ground truth.

Due to the limitations of supervised methods, the focus of research has switched to unsupervised DNN methods that do not need

ground-truth deformation fields. Unsupervised DNNs optimize an energy function on the input images, similar to traditional methods. However, DNN-based methods learn a common registration representation from a training set and then apply it to unseen images. Note that the term “unsupervised” refers to the absence of ground-truth deformation fields, but the network still needs training (this is also known as “self-supervised”). de Vos et al. (2019), Balakrishnan et al. (2018, 2019) are representative of unsupervised DNN-based methods.

More recently, diffeomorphic deformation representations have been developed to address the issue of non-smooth deformations in DNN-based methods. We briefly introduce its concepts in the next subsection.

2.1.2. Diffeomorphic image registration

Diffeomorphic deformable image registration is important in many medical image applications, owing to its special properties including topology preservation and transformation invertibility. A diffeomorphic transformation is a smooth and continuous one-to-one mapping with invertible derivatives (i.e., non-zero Jacobian determinant). Such a transformation can be achieved via the time-integration of time-dependent (Beg et al., 2005; Avants et al., 2008) or time-stationary velocity fields (SVFs) (Arsigny et al., 2006; Ashburner, 2007; Vercauteren et al., 2009; Hernandez et al., 2009). In the time-dependent setting (e.g., LDDMM Beg et al., 2005) and SyN (Avants et al., 2008), a diffeomorphic transformation ϕ is obtained via integrating the sufficiently smooth time-varying velocity fields $v^{(t)}$, i.e., $\frac{d}{dt} \phi^{(t)} = v^{(t)}(\phi^{(t)})$, where $\phi^{(t)} = id$ is the identity transform. On the other hand, in the stationary velocity fields (SVFs) setting (e.g., DARTEL (Ashburner, 2007) and diffeomorphic Demons (Vercauteren et al., 2009)), the velocity fields are assumed to be stationary over time, i.e., $\frac{d}{dt} \phi^{(t)} = v(\phi^{(t)})$. Dalca et al. (2019) first adopt the diffeomorphism formulation in a deep learning model, using the SVFs setting with an efficient scaling-and-squaring approach (Arsigny et al., 2006). In the scaling-and-squaring approach, the deformation field is represented as a Lie algebra member that is exponentiated to generate a time 1 deformation $\phi^{(1)}$, which is a member of the Lie group: $\phi^{(1)} = \exp(v)$. This means that the exponentiated flow field compels the mapping to be diffeomorphic and invertible using the same flow field. Starting from an initial deformation field:

$$\phi^{(1/2^T)} = p + \frac{v(p)}{2^T}, \quad (3)$$

where p denotes the spatial locations. The $\phi^{(1)}$ can be obtained using the recurrence:

$$\phi^{(1/2^{t-1})} = \phi^{(1/2^t)} \circ \phi^{(1/2^t)}. \quad (4)$$

Thus, $\phi^{(1)} = \phi^{(1/2)} \circ \phi^{(1/2)}$.

In practice, a neural network first generates a displacement field, which is then scaled by $1/2^T$ to produce an initial deformation field $\phi^{(1/2^T)}$. Subsequently, the squaring technique (i.e., Eq. (4)) is applied recursively to $\phi^{(1/2^T)}$ T times via a spatial transformation function, resulting in a final diffeomorphic deformation field $\phi^{(1)}$. Despite the fact that diffeomorphisms are theoretically guaranteed to be invertible, interpolation errors can lead to invertibility errors that increase linearly with the number of interpolation steps (Avants et al., 2008; Mok and Chung, 2020).

2.2. Self-attention mechanism and transformer

Transformer makes use of a self-attention mechanism that estimates the relevance of one input sequence to another via the Query-Key-Value (QKV) model (Vaswani et al., 2017; Dosovitskiy et al., 2020). The input sequences often originate from the flattened patches of an image. Let x be an image volume defined over a 3D spatial domain (i.e., $x \in \mathbb{R}^{H \times W \times L}$). The image is first divided into N flattened 3D patches $x_p \in \mathbb{R}^{N \times P^3}$, where (H, W, L) is the size of the original image, (P, P, P) is the size of each image patch, and $N = \frac{HWL}{P^3}$. Then, a

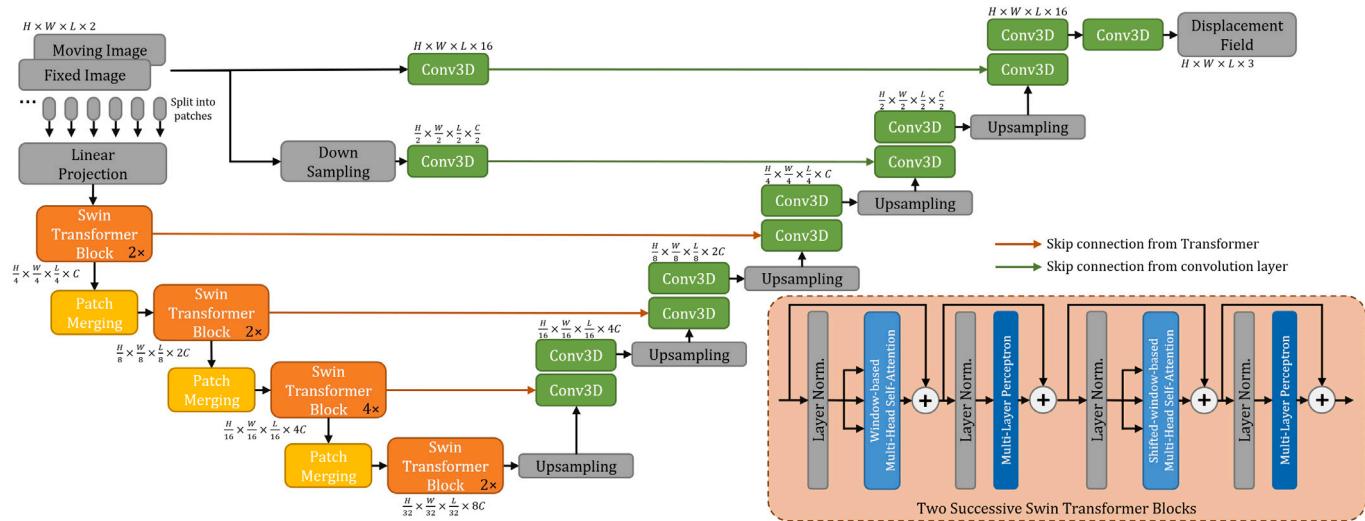


Fig. 1. The architecture of the proposed TransMorph registration network.

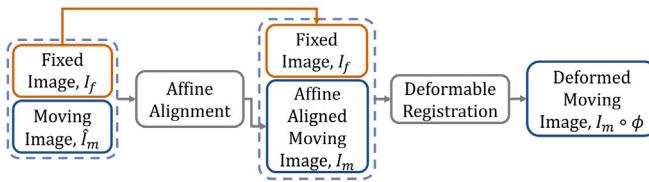


Fig. 2. The conventional paradigm of image registration.

learnable linear embedding \mathbf{E} is applied to \mathbf{x}_p , which projects each patch into a $D \times 1$ vector representation:

$$\hat{\mathbf{x}}_e = [\mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \dots; \mathbf{x}_p^N \mathbf{E}], \quad \mathbf{E} \in \mathbb{R}^{P^3 \times D} \quad (5)$$

where the dimension D is a user-defined hyperparameter. Then, a learnable positional embedding is added to $\hat{\mathbf{x}}_e$ so that the patches can retain their positional information, i.e., $\mathbf{x}_e = \hat{\mathbf{x}}_e + \mathbf{E}_{pos}$, where $\mathbf{E}_{pos} \in \mathbb{R}^{N \times D}$. These vector representations, often known as tokens, are subsequently used as inputs for self-attention computations.

Self-attention. To compute self-attention (SA), $\mathbf{x}_e \in \mathbb{R}^{N \times D}$ is encoded by \mathbf{U} (i.e., a linear layer) to three matrix representations: Queries $\mathbf{Q} \in \mathbb{R}^{N \times D_k}$, Keys $\mathbf{K} \in \mathbb{R}^{N \times D_k}$, and Values $\mathbf{V} \in \mathbb{R}^{N \times D_v}$. The scaled dot-product attention is given by:

$$\begin{aligned} [\mathbf{Q}, \mathbf{K}, \mathbf{V}] &= \mathbf{x}_e \mathbf{U}_{q,k,v} \quad \mathbf{U}_{q,k,v} \in \mathbb{R}^{D \times D_{q,k,v}}, \\ \mathbf{A} &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D_k}}\right) \quad \mathbf{A} \in \mathbb{R}^{N \times N}, \\ \text{SA}(\mathbf{x}_e) &= \mathbf{AV}, \end{aligned} \quad (6)$$

where \mathbf{A} is the attention weight matrix, each element of \mathbf{A} represents the pairwise similarity between two elements of the input sequence \mathbf{x}_e and their respective query and key representations. In general, SA computes a normalized score for each input token based on the dot product of the Query and Key representations. The score is subsequently applied to the Value representation of the token, signifying to the network whether or not to focus on this token.

Multi-head self-attention. A Transformer employs multi-head self-attention (MSA) rather than a single attention function. MSA is an extension of self-attention in which h self-attention operations (i.e., ‘heads’) are processed in parallel, thereby effectively increasing the number of trainable parameters. Then, the outputs of the SA operations are concatenated then projected onto a D -dimensional representation:

$$MSA(\mathbf{x}_e) = [SA_1(\mathbf{x}_e); SA_2(\mathbf{x}_e); \dots; SA_h(\mathbf{x}_e)] \mathbf{U}_{MSA} \quad (7)$$

where $\mathbf{U}_{MSA} \in \mathbb{R}^{h \cdot D_h \times D}$, and D_h is typically set to D/h in order to keep the number of parameters constant before and after the MSA operation.

2.3. Bayesian deep learning

Uncertainty estimates help comprehend what a machine learning model does not know. They indicate the likelihood that a neural network may make an incorrect prediction. Because most deep neural networks are incapable of providing an estimate of the uncertainty in their output values, their predictions are frequently taken at face value and thought to be correct. *Bayesian deep learning* estimates predictive uncertainty, providing a realistic paradigm for understanding uncertainty within deep neural networks (Gal and Ghahramani, 2016). The uncertainty caused by the parameters in a neural network is known as epistemic uncertainty, which is modeled by placing a prior distribution (e.g., a Gaussian prior distribution: $\mathbf{W} \sim \mathcal{N}(0, I)$) on the parameters of a network and then attempting to capture how much these weights vary given specific data. Recent efforts in this area include the *Bayes by Back-prop* (Blundell et al., 2015), its closely related mean-field variational inference by assuming a Gaussian prior distribution (Tölle et al., 2021), stochastic batch normalization (Atanov et al., 2018), and Monte-Carlo (MC) dropout (Gal and Ghahramani, 2016; Kendall and Gal, 2017). The applications of Bayesian deep learning in medical imaging expands on image denoising (Tölle et al., 2021; Laves et al., 2020b) and image segmentation (DeVries and Taylor, 2018; Baumgartner et al., 2019; Mehrtash et al., 2020). In deep-learning-based image registration, the majority of methods provide a single, deterministic solution of the unknown geometric transformation. Knowing about epistemic uncertainty helps determine if and to what degree the registration results can be trusted and whether the input data is appropriate for the neural network.

In general, two categories of registration uncertainty may be modeled using the epistemic uncertainty of a deep learning model: transformation uncertainty and appearance uncertainty (Luo et al., 2019; Xu et al., 2022). Transformation uncertainty measures the local ambiguity of the spatial transformation (i.e., the deformation), whereas appearance uncertainty quantifies the uncertainty in the intensity values of registered voxels or the volumes of the registered organs. Transformation uncertainty estimates may be used for uncertainty-weighted registration (Simpson et al., 2011; Kybic, 2009), surgical treatment planning, or directly visualized for qualitative evaluations (Yang et al., 2017b). Appearance uncertainty may be translated into dose uncertainties in cumulative dose for radiation or radiopharmaceutical therapy (Risholm et al., 2011; Vickress et al., 2017; Chetty and Rosu-Bubulac, 2019; Gear et al., 2018). These registration uncertainty estimates also enable the assessment of operative risks and leads to

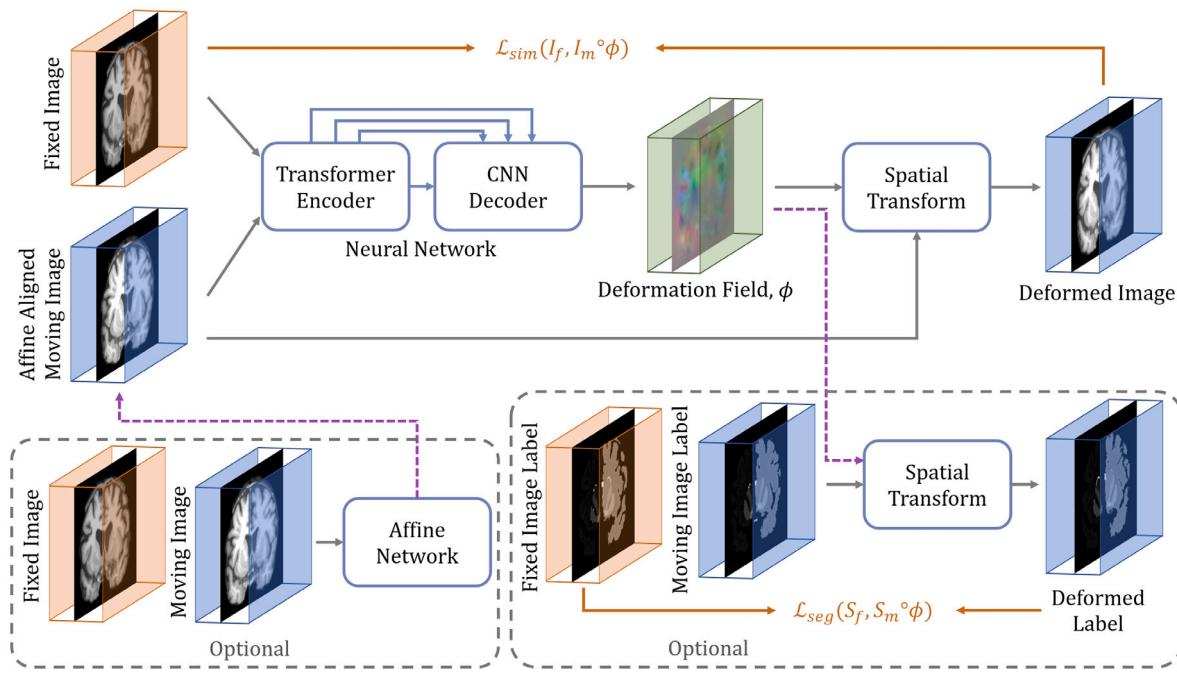


Fig. 3. The overall framework of the proposed Transformer-based image registration model, TransMorph. The proposed hybrid Transformer-ConvNet network takes two inputs: a fixed image and a moving image that is affinely aligned with the fixed image. The network generates a nonlinear warping function, which is then applied to the moving image through a spatial transformation function. If an image pair has not been affinely aligned, an affine Transformer may be used prior to the deformable registration (left dashed box). Additionally, auxiliary anatomical segmentations may be leveraged during training the proposed network (right dashed box).

better-informed clinical decisions (Luo et al., 2019). Cui et al. (2021) and Yang et al. (2017b) incorporated MC dropout layers in their registration network designs, allowing for the estimation of transformation uncertainty by sampling multiple deformation field predictions from the network.

The proposed image registration framework expands on these ideas. In particular, a new registration framework is presented that leverages a Transformer in the network design. We demonstrate that this framework can be readily adapted to several existing techniques to allow diffeomorphism for image registration, and incorporate Bayesian deep learning to estimate registration uncertainty.

3. Methods

The conventional paradigm of image registration is shown in Fig. 2. The moving and fixed images, denoted respectively as \hat{I}_m and I_f , are first affinely transformed into a single coordinate system. The resulting affine-aligned moving image is denoted as I_m . Subsequently, I_m is warped to I_f using a deformation field, ϕ , generated by a DIR algorithm (i.e., $I_m \circ \phi$). Fig. 3 presents an overview of the proposed method. Here, both the affine transformation and the deformable registration are performed using Transformer-based neural networks. The affine Transformer takes \hat{I}_m and I_f as inputs and computes a set of affine transformation parameters (e.g., rotation angle, translation, etc.). These parameters are used to affinely align \hat{I}_m with I_f via an affine transformation function, yielding an aligned image I_m . Then, a DIR network computes a deformation field ϕ given I_m and I_f , which warps I_m using a spatial transformation function (i.e., $\hat{I}_f = I_m \circ \phi$). During training, the DIR network may optionally include supplementary information (e.g., anatomical segmentation). The network architectures, the loss and regularization functions, and the variants of the method are described in detail in the following sections.

3.1. Affine transformation network

Affine transformation is often used as the initial stage in image registration because it facilitates the optimization of the following more

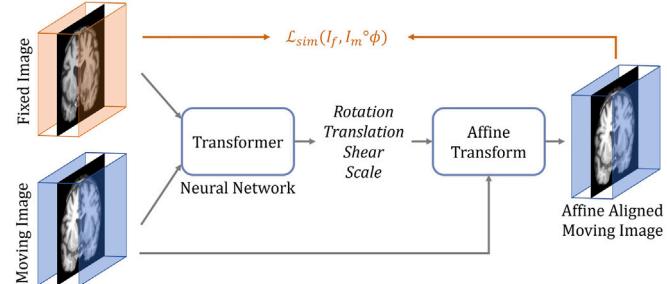


Fig. 4. The framework of the proposed Transformer-based affine model.

complicated DIR processes (de Vos et al., 2019). An affine network examines a pair of moving and fixed images globally and produces a set of transformation parameters that aligns the moving image with the fixed image. Here (see Fig. 4), the architecture of the proposed Transformer-based affine network is a modified Swin Transformer (Liu et al., 2021a) that takes two 3D volumes as the inputs (i.e., I_f and \hat{I}_m) and generates 12 affine parameters: three rotation angles, three translation parameters, three scaling parameters, and three shearing parameters. The details and a visualization of the architecture are shown in Fig. A.19 in Appendix A. We reduced the number of parameters in the original Swin Transformer due to the relative simplicity of affine registration. The specifics of the Transformer's architecture and parameter settings are covered in a subsequent section.

3.2. Deformable registration network

Fig. 1 shows the network architecture of the proposed TransMorph. The encoder of the network first splits the input moving and fixed volumes into non-overlapping 3D patches, each of size $2 \times P \times P \times P$, where P is typically set to 4 (Dosovitskiy et al., 2020; Liu et al., 2021a; Dong et al., 2021). We denote the i th patch as x_p^i , where $i \in \{1, \dots, N\}$ and $N = \frac{H}{P} \times \frac{W}{P} \times \frac{L}{P}$ is the total number of patches. Each patch is

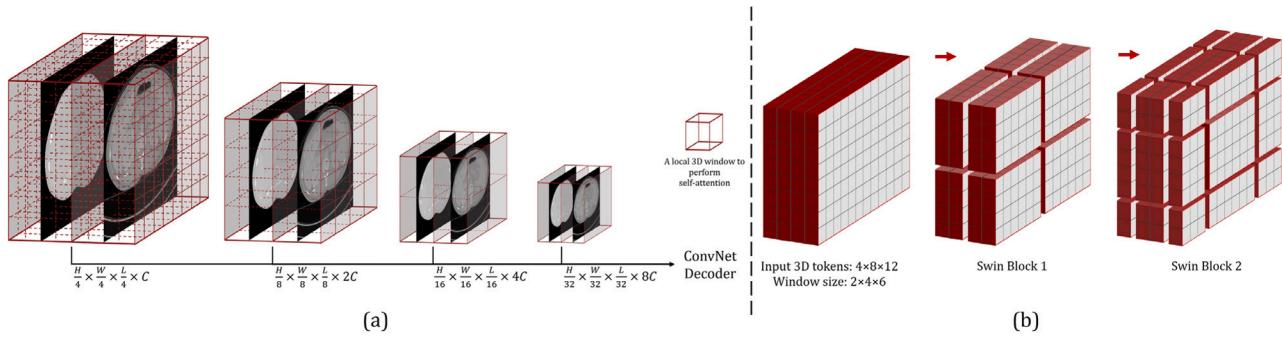


Fig. 5. (a): Swin Transformer creates hierarchical feature maps by merging image patches. The self-attention is computed within each local 3D window (the red box). The feature maps generated at each resolution are sent into a ConvNet decoder to produce an output. (b): The 3D cyclic shift of local windows for shifted-window-based self-attention computation.

flattened and regarded as a “token”, and then a linear projection layer is used to project each token to a feature representation of an arbitrary dimension (denoted as C):

$$\mathbf{z}_0 = [x_p^1 \mathbf{E}; x_p^2 \mathbf{E}; \dots; x_p^N \mathbf{E}], \quad (8)$$

where $\mathbf{E} \in \mathbb{R}^{2P^3 \times C}$ denotes the linear projection, and the output \mathbf{z}_0 has a dimension of $N \times C$.

Because the linear projection operates on image patches and does not keep the token’s location relative to the image as a whole, previous Transformer-based models often added a positional embedding to the linear projections in order to integrate the positional information into tokens, i.e. $\mathbf{z}_0 + \mathbf{E}_{pos}$ (Vaswani et al., 2017; Dosovitskiy et al., 2020; Liu et al., 2021a; Dong et al., 2021). Such Transformers were primarily designed for image classification, where the output is often a vector describing the likelihood of an input image being classified as a certain class. Thus, if the positional embedding is not employed, the Transformer may lose the positional information. However, for pixel-level tasks such as image registration, the network often includes a decoder that generates a dense prediction with the same resolution as the input or target image. The spatial correspondence between voxels in the output image is enforced by comparing the output with the target image using a loss function. Any spatial mismatches between output and target would contribute to the loss and be backpropagated into the Transformer encoder. The Transformer should thereby inherently capture the tokens’ positional information. In this work, we observed, as will be shown in Section 6.1.2, that positional embedding is not necessary for image registration, and it only adds extra parameters to the network without improving performance.

Following the linear projection layer, several consecutive stages of patch merging and Swin Transformer blocks (Liu et al., 2021a) are applied on the tokens \mathbf{z}_0 . The Swin Transformer blocks outputs the same number of tokens as the input, while the patch merging layers concatenate the features of each group of $2 \times 2 \times 2$ neighboring tokens, thus they reduce the number of tokens by a factor of $2 \times 2 \times 2 = 8$ (e.g., $H \times W \times L \times C \rightarrow \frac{H}{2} \times \frac{W}{2} \times \frac{L}{2} \times 8C$). Then, a linear layer is applied on the $8C$ -dimensional concatenated features to produce features each of $2C$ -dimension. After four stages of Swin Transformer blocks and three stages of patch merging in between the Transformer stages (i.e., orange boxes in Fig. 1), the output dimension at the last stage of the encoder is $\frac{H}{32} \times \frac{W}{32} \times \frac{L}{32} \times 8C$. The decoder consists of successive upsampling and convolutional layers with the kernel size of 3×3 . Each of the upsampled feature maps in the decoding stage was concatenated with the corresponding feature map from the encoding path via skip connections, then followed by two consecutive convolutional layers. As shown in Fig. 1, the Transformer encoder can only provide feature maps up to a resolution of $\frac{H}{P} \times \frac{W}{P} \times \frac{L}{P}$ owing to the nature of patch operation (denoted by the orange arrows). Hence, Transformer may fall short of delivering high-resolution feature maps and aggregating local information at lower layers (Raghu et al., 2021). To address this

shortcoming, we employed two convolutional layers using the original and downsampled image pair as inputs to capture local information and generate high-resolution feature maps. The outputs of these layers were concatenated with the feature maps in the decoder to produce a deformation field. The output deformation field, ϕ , was generated by the application of sixteen 3×3 convolutions. Except for the last convolutional layer, each convolutional layer is followed by a Leaky Rectified Linear Unit (Maas et al., 2013) activation. Finally, the spatial transformation function (Jaderberg et al., 2015) is used to apply a nonlinear warp to the moving image I_m with the deformation field ϕ (or the displacement field \mathbf{u}) provided by the network.

In the next subsections, we discuss the Swin Transformer block, the spatial transformation function, and the loss functions in detail.

3.2.1. 3D Swin Transformer block

Swin Transformer (Liu et al., 2021a) can generate hierarchical feature maps at various resolutions by using patch merging layers, making it ideal for usage as a general-purpose backbone for pixel-level tasks like image registration and segmentation. Swin Transformer’s most significant component, apart from patch merging layers, is the shifted window-based self-attention mechanism. Unlike ViT (Dosovitskiy et al., 2020), which computes the relationships between a token and all other tokens at each step of the self-attention modules. Swin Transformer computes self-attention within the evenly partitioned non-overlapping local windows of the original and the lower resolution feature maps (as shown in Fig. 5(a)). In contrast to the original Swin Transformer, this work uses rectangular-parallelepiped windows to accommodate non-square images, and each has a shape of $M_x \times M_y \times M_z$. At each resolution, the first Swin Transformer block employs a regular window partitioning method, beginning with the top-left voxel, and the feature maps are evenly partitioned into non-overlapping windows of size $M_x \times M_y \times M_z$. The self-attention is then calculated locally within each window. To introduce connections between neighboring windows, the Swin Transformer uses a shifted window design: in the successive Swin Transformer blocks, the windowing configuration shifts from that of the preceding block, by displacing the windows in the preceding block by $(\lfloor \frac{M_x}{2} \rfloor \times \lfloor \frac{M_y}{2} \rfloor \times \lfloor \frac{M_z}{2} \rfloor)$ voxels. As illustrated by an example in Fig. 5 (b), the input feature map has $4 \times 8 \times 12$ voxels. With a window size of $2 \times 4 \times 6$, the feature map is evenly partitioned into $2 \times 2 \times 2 = 8$ windows in the first Swin Transformer block (“Swin Block 1” in Fig. 5 (b)). Then, in the next block, the windows are shifted by $(\lfloor \frac{M_x}{2} \rfloor \times \lfloor \frac{M_y}{2} \rfloor \times \lfloor \frac{M_z}{2} \rfloor) = (1 \times 2 \times 3)$, and the number of windows becomes $3 \times 3 \times 3 = 27$. We extended the original 2D efficient batch computation (i.e., cyclic shift) (Liu et al., 2021a,b) to 3D and applied it to the 27 shifted windows, keeping the final number of windows for attention computation at 8. With the windowing-based attention, two

consecutive Swin Transformer blocks can be computed as:

$$\begin{aligned}\hat{\mathbf{z}}_\ell &= \text{W-MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \\ \mathbf{z}_\ell &= \text{MLP}(\text{LN}(\hat{\mathbf{z}}_\ell)) + \hat{\mathbf{z}}_\ell, \\ \hat{\mathbf{z}}_{\ell+1} &= \text{SW-MSA}(\text{LN}(\mathbf{z}_\ell)) + \mathbf{z}_\ell,\end{aligned}\quad (9)$$

$\mathbf{z}_{\ell+1}$ = MLP(LN($\hat{\mathbf{z}}_{\ell+1}$)) + $\hat{\mathbf{z}}_{\ell+1}$, where W-MSA and SW-MSA denote, respectively, window-based multi-head self-attention and shifted-window-based multi-head self-attention modules; MLP denotes the multi-layer perceptron module (Vaswani et al., 2017); $\hat{\mathbf{z}}_\ell$ and \mathbf{z}_ℓ denote the output features of the (S)W-MSA and the MLP module for block ℓ , respectively. The self-attention is computed as:

$$\mathbf{A}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}} + B\right)V, \quad (10)$$

where $Q, K, V \in \mathbb{R}^{M_x M_y M_z \times d}$ are query, key, value matrices, d denotes the dimension of query and key features, $M_x M_y M_z$ is the number of tokens in a 3D window, and B represents the relative position of tokens in each window. Since the relative position between tokens along each axis (i.e., x, y, z) can only take values from $[-M_{x,y,z} + 1, M_{x,y,z} - 1]$, the values in B are taken from a smaller bias matrix $\hat{B} \in \mathbb{R}^{(2M_x-1) \times (2M_y-1) \times (2M_z-1)}$. For the reasons given previously, we will show in Section 6.1.2 that positional bias B is not needed for the proposed network and that it just adds extra parameters without improving registration performance.

3.2.2. Loss functions

The overall loss function for network training derives from the energy function of traditional image registration algorithms (i.e., Eq. (1)). The loss function consists of two parts: one computes the similarity between the deformed moving and the fixed images, and another one regularizes the deformation field so that it is smooth:

$$\mathcal{L}(I_f, I_m, \phi) = \mathcal{L}_{sim}(I_f, I_m, \phi) + \lambda \mathcal{R}(\phi), \quad (11)$$

where \mathcal{L}_{sim} denotes the image fidelity measure, and \mathcal{R} denotes the deformation field regularization.

Image similarity measure. In this work, we experimented with two widely-used similarity metric for \mathcal{L}_{sim} . The first was the mean squared error, which was the mean of the squared difference in voxel values between I_f and I_m :

$$MSE(I_f, I_m, \phi) = \frac{1}{\Omega} \sum_{\mathbf{p} \in \Omega} |I_f(\mathbf{p}) - [I_m \circ \phi](\mathbf{p})|^2, \quad (12)$$

where \mathbf{p} denotes the voxel location, and Ω represents the image domain.

Another similarity metric used was the local normalized cross-correlation between I_f and I_m :

$$\begin{aligned}LNCC(I_f, I_m, \phi) &= \\ \sum_{\mathbf{p} \in \Omega} \frac{\left(\sum_{\mathbf{p}_i} (f(\mathbf{p}_i) - \bar{f}(\mathbf{p}))([I_m \circ \phi](\mathbf{p}_i) - [\bar{I}_m \circ \phi](\mathbf{p})) \right)^2}{\left(\sum_{\mathbf{p}_i} (f(\mathbf{p}_i) - \bar{f}(\mathbf{p}))^2 \right) \left(\sum_{\mathbf{p}_i} ([I_m \circ \phi](\mathbf{p}_i) - [\bar{I}_m \circ \phi](\mathbf{p}))^2 \right)},\end{aligned}\quad (13)$$

where $\bar{I}_f(\mathbf{p})$ and $\bar{I}_m(\mathbf{p})$ denotes the mean voxel value within the local window of size n^3 centered at voxel \mathbf{p} . We used $n = 9$ in the experiments.

Deformation field regularization. Optimizing the similarity metric alone would encourage $I_m \circ \phi$ to be visually as close as possible to I_f . The resulting deformation field ϕ , however, might not be smooth or realistic. To impose smoothness in the deformation field, a regularizer $\mathcal{R}(\phi)$ was added to the loss function. $\mathcal{R}(\phi)$ encourages the displacement value in a location to be similar to the values in its neighboring locations. Here,

we experimented with two regularizers. The first was the diffusion regularizer (Balakrishnan et al., 2019):

$$\mathcal{R}_{diffusion}(\phi) = \sum_{\mathbf{p} \in \Omega} \|\nabla \mathbf{u}(\mathbf{p})\|^2, \quad (14)$$

where $\mathbf{u}(\mathbf{p})$ is the spatial gradients of the displacement field \mathbf{u} . The spatial gradients were approximated using forward differences, that is, $\frac{\partial \mathbf{u}(\mathbf{p})}{\partial [x,y,z]} \approx \mathbf{u}(p_{\{x,y,z\}} + 1) - \mathbf{u}(p_{\{x,y,z\}})$.

The second regularizer was bending energy (Rueckert et al., 1999), which penalizes sharply curved deformations, thus, it may be helpful for abdominal organ registration. Bending energy operates on the second derivative of the displacement field \mathbf{u} , and it is defined as:

$$\begin{aligned}\mathcal{R}_{bending}(\phi) &= \sum_{\mathbf{p} \in \Omega} \|\nabla^2 \mathbf{u}(\mathbf{p})\|^2 = \sum_{\mathbf{p} \in \Omega} \left[\left(\frac{\partial^2 \mathbf{u}(\mathbf{p})}{\partial x^2} \right)^2 + \left(\frac{\partial^2 \mathbf{u}(\mathbf{p})}{\partial y^2} \right)^2 + \right. \\ &\quad \left. \left(\frac{\partial^2 \mathbf{u}(\mathbf{p})}{\partial z^2} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{u}(\mathbf{p})}{\partial x \partial z} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{u}(\mathbf{p})}{\partial y \partial z} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{u}(\mathbf{p})}{\partial x \partial y} \right)^2 \right],\end{aligned}\quad (15)$$

where the derivatives were estimated using the same forward differences that were used previously.

Auxiliary segmentation information. When the organ segmentations of I_f and I_m are available, TransMorph may leverage this auxiliary information during training to improve the anatomical mapping between $I_m \circ \phi$ and I_f . A loss function \mathcal{L}_{seg} that quantifies the segmentation overlap is added to the overall loss function (Eq. (11)):

$$\mathcal{L}(I_f, I_m, \phi) = \mathcal{L}_{sim}(I_f, I_m, \phi) + \lambda \mathcal{R}(\phi) + \gamma \mathcal{L}_{seg}(s_f, s_m, \phi), \quad (16)$$

where s_f and s_m represent, respectively, the organ segmentation of I_f and I_m , and γ is a weighting parameter that controls the strength of \mathcal{L}_{seg} . In the field of image registration, it is common to use Dice score (Dice, 1945) as a figure of merit to quantify registration performance. Therefore, we directly minimized the Dice loss (Milletari et al., 2016) between s_f^k and s_m^k , where k represents the k th structure/organ:

$$Dice(s_f, s_m, \phi) = 1 -$$

$$\frac{1}{K} \sum_k \frac{2 \sum_{\mathbf{p} \in \Omega} s_f^k(\mathbf{p}) [s_m^k \circ \phi](\mathbf{p})}{\sum_{\mathbf{p} \in \Omega} (s_f^k(\mathbf{p}))^2 + \sum_{\mathbf{p} \in \Omega} ((s_m^k \circ \phi)(\mathbf{p}))^2}. \quad (17)$$

To allow backpropagation of the Dice loss, we used a method similar to that described in Balakrishnan et al. (2019), in which we designed s_f and s_m as image volumes with K channels, each channel containing a binary mask defining the segmentation of a specific structure/organ. Then, $s_m \circ \phi$ is computed by warping the K -channel s_m with ϕ using linear interpolation so that the gradients of \mathcal{L}_{seg} can be backpropagated into the network.

3.3. Probabilistic and B-spline variants

In this section, we demonstrate that by simply altering the decoder, TransMorph can be used in conjunction with the concepts from prior research to ensure a diffeomorphic deformation such that the resulting deformable mapping is continuous, differentiable, and topology-preserving. The diffeomorphic registration was achieved using the scaling-and-squaring approach (described in Section 2.1.2) with a stationary velocity field representation (Arsigny et al., 2006). Two existing diffeomorphic models, VoxelMorph-diff (Dalca et al., 2019) and MIDIR (Qiu et al., 2021), have been adopted as bases for the proposed TransMorph diffeomorphic variants, designated by TransMorph-diff (Appendix H) and TransMorph-bspl (Appendix I), respectively. The architectures of the two variants are shown in Fig. 6. The detailed derivation of these two variants are listed in Appendix A.

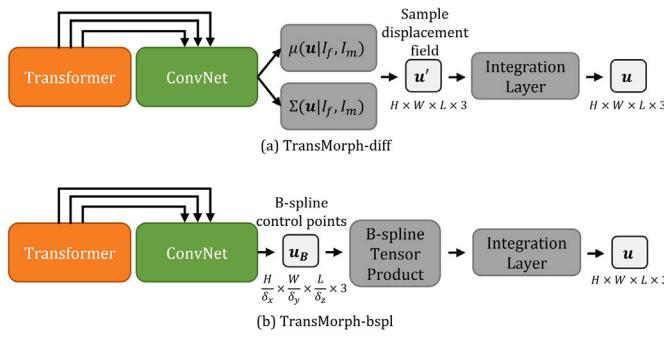


Fig. 6. The probabilistic and B-spline variants of TransMorph. (a): The architecture of the probabilistic diffeomorphic TransMorph. (b): The architecture of the B-spline diffeomorphic TransMorph.

TransMorph-diff was trained using the same loss functions as VoxelMorph-diff (Dalca et al., 2019):

$$\begin{aligned} \mathcal{L}_{prob.}(I_f, I_m, \phi_u; \psi) &= -\mathbb{E}_{\mathbf{u} \sim q_{\psi}} [\log p(I_f | \mathbf{u}, I_m)] + \text{KL}[q_{\psi}(\mathbf{u} | I_f, I_m) || p(\mathbf{u})] \\ &= \frac{1}{2\sigma^2} \|I_f - I_m \circ \phi_u\|^2 + \frac{1}{2} \left[\text{tr}(\lambda \mathbf{D} \Sigma_{\psi} - \log \Sigma_{\psi}) + \boldsymbol{\mu}_{\psi}^T \boldsymbol{\Lambda}_u \boldsymbol{\mu}_{\psi} \right], \end{aligned} \quad (18)$$

and when anatomical label maps are available:

$$\begin{aligned} \mathcal{L}_{prob. w/ aux.}(I_f, s_f, I_m, s_m, \phi_u; \psi) &= \frac{1}{2\sigma^2} \|I_f - I_m \circ \phi_u\|^2 + \frac{1}{2\sigma_s^2} \|s_f - s_m \circ \phi_u\|^2 \\ &\quad + \frac{1}{2} \left[\text{tr}(\lambda \mathbf{D} \Sigma_{\psi} - \log \Sigma_{\psi}) + \boldsymbol{\mu}_{\psi}^T \boldsymbol{\Lambda}_u \boldsymbol{\mu}_{\psi} \right]. \end{aligned} \quad (19)$$

However, it is important to note that in Dalca et al. (2019), s_f and s_m represent *anatomical surfaces* obtained from label maps. In contrast, we directly used the *label maps* as s_f and s_m in this work. They were image volumes with multiple channels, each channel contained a binary mask defining the segmentation of a certain structure/organ.

3.4. Bayesian uncertainty variant

In this section, we extend the proposed TransMorph to a Bayesian neural network (BNN) using the variational inference framework with Monte Carlo dropout (Gal and Ghahramani, 2016), for which we refer readers to (Gal and Ghahramani, 2016; Yang et al., 2017a, 2016) for both theoretical and technical details. We denoted the resulting model as TransMorph-Bayes. In this model, Dropout layers were inserted into the Transformer encoder of the TransMorph architecture but not into the ConvNet decoder, in order to avoid imposing excessive regularity for the network parameters and thus decreasing performance. We added a dropout layer after each fully connected layer in the MLPs (Eq. (9)) and after each self-attention computation (Eq. (10)). Note that these are the locations where dropout layers are commonly used for Transformer training. We set the dropout probability p to 0.15 to further avoid the network imposing an excessive degree of regularity on the network weights.

Both the transformation and appearance uncertainty can be estimated as the variability from the predictive mean (i.e., the variance), where the predictive mean of the deformation fields and the deformed images can be estimated by Monte Carlo integration (Gal and Ghahramani, 2016):

$$\hat{\phi} = \frac{1}{T} \sum_{t=1}^T \phi_t, \quad (20)$$

and

$$\hat{I}_f = \frac{1}{T} \sum_{t=1}^T I_m \circ \phi_t. \quad (21)$$

This is equivalent to averaging the output of T forward passes through the network during inference, where ϕ_t represents the deformation field produced by t th forward pass. The transformation and appearance uncertainty can be estimated using the predictive variances of the deformation fields and the deformed images, respectively, as:

$$\hat{\Sigma}_{\phi}^2 = \frac{1}{T} \sum_{t=1}^T (\phi_t - \hat{\phi})^2, \quad (22)$$

and

$$\hat{\Sigma}_f^2 = \frac{1}{T} \sum_{t=1}^T (I_m \circ \phi_t - \hat{I}_f)^2. \quad (23)$$

3.4.1. Appearance uncertainty calibration

An ideal uncertainty estimate should be properly correlated to the inaccuracy of the registration results; that is, a high uncertainty value should indicate a large registration error, and vice versa. Otherwise, doctors/surgeons may be misled by the erroneous estimate of registration uncertainty and place unwarranted confidence in the registration results, resulting in severe consequences (Luo et al., 2019; Rishholm et al., 2013, 2011). The appearance uncertainty given by Eq. (23) is expressed as the variability from the mean model prediction. Such an appearance uncertainty estimation does not account for the systematic errors (i.e., bias) between the mean registration prediction and the target image; therefore, a low uncertainty value given by Eq. (23) does not always guarantee an accurate registration result.

When the predicted uncertainty values closely corresponded to the expected model error, the uncertainty estimates are considered to be well-calibrated (Laves et al., 2019; Levi et al., 2019). In an ideal scenario, the estimated registration uncertainty should completely reflect the actual registration error. For instance, if the predictive variance of a batch of registered images generated by the network is found to be 0.5, the expectation of the squared error should likewise be 0.5. Accordingly, if the expected model error is quantified by MSE, then the perfect calibration of appearance uncertainty may be defined as the following (Guo et al., 2017; Levi et al., 2019; Laves et al., 2020c):

$$\mathbb{E}_{\Sigma^2} [\|I_m \circ \phi - I_f\|^2 | \hat{\Sigma}^2 = \Sigma^2] = \Sigma^2 \quad \forall \{\Sigma^2 \in \mathbb{R} | \Sigma^2 \geq 0\}. \quad (24)$$

In the conventional paradigm of Bayesian neural networks, the uncertainty estimate is derived from the predictive variance $\hat{\Sigma}^2$ relative to the predictive mean \hat{I}_f as in Eq. (23). However, it can be shown that this predictive variance can be miscalibrated as a result of overfitting the training dataset (as shown in Appendix B). Therefore, the uncertainty values estimated based on $\hat{\Sigma}_f^2$ in Eq. (23) may be biased. This bias must be corrected in applications such as image denoising or classification (Laves et al., 2019; Guo et al., 2017; Kuleshov et al., 2018; Phan et al., 2018; Laves et al., 2020c,a), such that the uncertainty values closely reflect the expected error. In image registration, however, the expected appearance error may be computed even during the test time since the target image is always known. Therefore, a perfectly calibrated appearance uncertainty quantification may be achieved without additional effort. Here, we propose to replace the predicted mean \hat{I}_f with the target image I_f in Eq. (23). Then, the appearance uncertainty is the equivalent to the expected error:

$$\Sigma_f^2 = err(I_m \circ \phi) = \frac{1}{T} \sum_{t=1}^T (I_m \circ \phi_t - I_f)^2. \quad (25)$$

A comparison between the two appearance uncertainty estimate methods (i.e., $\hat{\Sigma}_f^2$ and Σ_f^2) is shown later in this paper.

4. Experiments

4.1. Datasets and preprocessing

Three datasets including over 1000 image pairs were used to thoroughly validate the proposed method. The details of each dataset are described in the following sections.

4.1.1. Inter-patient brain MRI registration

For the inter-patient brain MR image registration dataset, we used a dataset of 260 T1-weighted brain MRI images acquired at Johns Hopkins University. The images were anonymized and acquired under IRB approval. The dataset was split into 182, 26, and 52 (7:1:2) volumes for training, validation, and test sets. Each image volume was used as a moving image to form two image pairs by randomly matching it to two other volumes in the set (i.e., the fixed images). Then, the moving and fixed images were inverted to form another two image pairs, resulting in four registration pairings of I_f and I_m . The final data comprises 768, 104, and 208 image pairs for training, validation, and testing, respectively. Fischl (2012) was used to perform standard pre-processing procedures for structural brain MRI, including skull stripping, resampling, and affine transformation. The pre-processed image volumes were all cropped to size of $160 \times 192 \times 224$. Label maps including 29 anatomical structures were obtained using FreeSurfer for evaluating registration performances.

4.1.2. Atlas-to-patient brain MRI registration

We used a publicly available dataset to evaluate the proposed model with atlas-to-patient brain MRI registration task. A total number of 576 T1-weighted brain MRI images from the Information eXtraction from Images (IXI) database² was used as the fixed images. The moving image for this task was an atlas brain MRI obtained from Kim et al. (2021). The dataset was split into 403, 58, and 115 (7:1:2) volumes for training, validation, and test sets. FreeSurfer was used to pre-process the MRI volumes. We carried out the same pre-processing procedures we used for the previous dataset applied to the IXI dataset. All image volumes were cropped to size of $160 \times 192 \times 224$. Label maps of 29 anatomical structures were used to evaluate registration performances.

4.1.3. Learn2Reg OASIS brain MRI registration

We additionally evaluated TransMorph on a public registration dataset, OASIS (Marcus et al., 2007; Hoopes et al., 2021), obtained from the 2021 Learn2Reg challenge (Hering et al., 2021) for inter-patient registration. This dataset contains a total of 451 brain T2 MRI images, with 394, 19, and 38 images being used for training, validation, and testing, respectively. FreeSurfer (Fischl, 2012) was used to pre-process the brain MRI images, and label maps for 35 anatomical structures were provided for evaluation.

4.1.4. XCAT-to-CT registration

Computerized phantoms have been widely used in the medical imaging field for algorithm optimization and imaging system validation (Christoffersen et al., 2013; Chen et al., 2019; Zhang et al., 2017). The four-dimensional extended cardiac-torso (XCAT) phantom (Segars et al., 2010) was developed based on anatomical images from the Visible Human Project data. While the current XCAT phantom³ can model anatomical variations through organ and phantom scaling, it cannot completely replicate the anatomical variations seen in humans. As a result, XCAT-to-CT registration (which can be thought of as atlas-to-image registration) has become a key method for creating anatomically variable phantoms (Chen et al., 2020; Fu et al., 2021; Segars et al., 2013). This research used a CT dataset from Segars et al. (2013) that

includes 50 non-contrast chest-abdomen-pelvis (CAP) CT scans that are part of the Duke University imaging database. Selected organs and structures were manually segmented in each patient's CT scan. The structures segmented included the following: the body outline, the bone structures, lungs, heart, liver, spleen, kidneys, stomach, pancreas, large intestine, prostate, bladder, gall bladder, and thyroid. The manual segmentation was done by several medical students, and the results were subsequently corrected by an experienced radiologist at Duke University. The CT volumes have voxel sizes ranging from $0.625 \times 0.625 \times 5$ mm to $0.926 \times 0.926 \times 5$ mm. We used trilinear interpolation to resample all volumes to an identical voxel spacing of $2.5 \times 2.5 \times 5$ mm. The volumes were all cropped and zero-padded to have a size of $160 \times 160 \times 160$ voxels. The intensity values were first clipped in the range of $[-1000, 700]$ Hounsfield Units and then normalized to the range of $[0, 1]$. The XCAT attenuation map was generated with a resolution of $1.1 \times 1.1 \times 1.1$ mm using the material compositions and attenuation coefficients of the constituents at 120 keV. It was then resampled, cropped, and padded so that the resulting volume matched the size of the CT volumes. The XCAT attenuation map's intensity values were also normalized to be within a range of $[0, 1]$. The XCAT and CT images were rigidly registered using the proposed affine network. The dataset was split into 35, 5, and 10 (7:1:2) volumes for training, validation, and testing. We conducted five-fold cross-validation on the fifty image volumes, resulting in 50 testing volumes in total.

4.2. Baseline methods

We compared TransMorph to various registration methods that have previously demonstrated state-of-the-art registration performance. We begin by comparing TransMorph with four non-deep-learning-based methods. The hyper-parameters of these methods, unless otherwise specified, were empirically set to balance the trade-off between registration accuracy and runtime. The methods and their hyperparameter settings are described below:

- SyN⁴ (Avants et al., 2008): For both inter-patient and atlas-to-patient brain MR registration tasks, we used the mean squared difference (MSQ) as the objective function, along with a default Gaussian smoothing of 3 and three scales with 180, 80, 40 iterations, respectively. For XCAT-to-CT registration, we used cross-correlation (CC) as the objective function, a Gaussian smoothing of 5 and three scales with 160, 100, 40 iterations, respectively.
- NiftyReg⁵ (Modat et al., 2010): We used the sum of squared differences (SSD) as the objective function and bending energy as the regularizer for all registration tasks. For inter-patient brain MR registration, we empirically used a regularization weighting of 0.0002 and three scales with 300 iterations each. For atlas-to-patient brain MR registration, the regularization weighting was set to 0.0006, and we used three scales with 500 iterations each. For XCAT-to-CT registration, we used a regularization weight of 0.0005 and five scales with 500 iterations each.
- deedsBCV⁶ (Heinrich et al., 2015): The objective function was self-similarity context (SSC) (Heinrich et al., 2013b) by default. For both inter-patient and atlas-to-patient brain MR registration, we used the hyperparameter values suggested in Hoffmann et al. (2020) for neuroimaging, in which the grid spacing, search radius, and quantization step were set to $6 \times 5 \times 4 \times 3 \times 2$, $6 \times 5 \times 4 \times 3 \times 2$, and $5 \times 4 \times 3 \times 2 \times 1$, respectively. For XCAT-to-CT registration, we used the default parameters suggested for abdominal CT registration (Heinrich et al., 2015), where the grid spacing, search radius, and quantization step were $8 \times 7 \times 6 \times 5 \times 4$, $8 \times 7 \times 6 \times 5 \times 4$, and $5 \times 4 \times 3 \times 2 \times 1$, respectively.

² <https://brain-development.org/ixi-dataset/>

³ as of October, 2021.

⁴ <https://github.com/ANTsX/ANTsPy>

⁵ <https://www.ucl.ac.uk/medical-image-computing>

⁶ <https://github.com/mattiaspaul/deedsBCV>

- LDDMM⁷ Beg et al. (2005): MSE was used as the objective function by default. For both inter-patient and atlas-to-patient brain MR registration, we used the smoothing kernel size of 5, the smoothing kernel power of 2, the matching term coefficient of 4, the regularization term coefficient of 10, and the iteration number of 500. For XCAT-to-CT registration, we used the same kernel size, kernel power, the matching term coefficient, and the number of iteration. However, the regularization term coefficient was empirically set to 3.

Next, we compared the proposed method with several existing deep-learning-based methods. For a fair comparison, unless otherwise indicated, the loss function (Eq. (11)) that consists of MSE (Eq. (12)) and diffusion regularization (Eq. (14)) was used for inter-patient brain MR registration, while we instead used LNCC (Eq. (13)) for atlas-to-patient MRI registration. For XCAT-to-CT registration, we used the loss function (Eq. (16)) that consists of LNCC (Eq. (13)), bending energy (Eq. (15)), and Dice loss (Eq. (17)). Auxiliary data (organ segmentation) was used for XCAT-to-CT registration only. Recall that the hyperparameters λ and γ define, respectively, the weight for deformation field regularization and Dice loss. The detailed parameter settings used for each method were as follows:

- VoxelMorph⁸ (Balakrishnan et al., 2018, 2019): We employed two variants of VoxelMorph, the second variant doubles the number of convolution filters in the first variant; they are designated as VoxelMorph-1 and -2, respectively. For inter-patient and atlas-to-patient brain MR registration, the regularization hyperparameter λ was set, respectively, to 0.02 and 1, where these values were reported as the optimal values in Balakrishnan et al. (2019). For XCAT-to-CT registration, we set $\lambda = \gamma = 1$.
- VoxelMorph-diff⁹ (Dalca et al., 2019): For both inter-patient and atlas-to-patient brain MR registration tasks, the loss function $\mathcal{L}_{prob.}$ (Eq. (18)) was used with σ set to 0.01 and λ set to 20. For XCAT-to-CT registration, we used the loss function $\mathcal{L}_{prob.w/aux.}$ (Eq. (19)) with $\sigma = \sigma_s = 0.01$ and $\lambda = 20$.
- CycleMorph¹⁰ (Kim et al., 2021): In CycleMorph, the hyperparameters α , β , and λ , correspond to the weights for cycle loss, identity loss, and deformation field regularization. For inter-patient brain MR registration, we set $\alpha = 0.1$, $\beta = 0.5$, and $\lambda = 0.02$. Whereas for atlas-to-patient brain MR registration, we set $\alpha = 0.1$, $\beta = 0.5$, and $\lambda = 1$. These values were recommended in Kim et al. (2021) as the optimal values for neuroimaging. For XCAT-to-CT registration, we modified the CycleMorph by adding a Dice loss with a weighting of 1 to incorporate organ segmentation during training, and we set $\alpha = 0.1$ and $\beta = 1$. We observed that the λ value of 1 suggested in Kim et al. (2021) yielded over-smoothed deformation field in our application. Therefore, the value of λ was decreased to 0.1.
- MIDIR¹¹ (Qiu et al., 2021): The same loss function and λ value as VoxelMorph were used. In addition, the control point spacing δ for B-spline transformation was set to 2 for all tasks, which was shown to be an optimal value in Qiu et al. (2021).

To evaluate the proposed Swin-Transformer-based network architecture, we compared its performance to existing Transformer-based networks that achieved state-of-the-art performance in other applications (e.g., image segmentation, object detection, etc.). We customized these models to make them suitable for image registration. They were modified to produce 3-dimensional deformation fields that warp the

Table 1

The ablation study of TransMorph models with skip connections and positional embedding. “Conv. skip.” denotes the skip-connections from convolutional layers (indicated by green arrows in Fig. 1); “Trans. skip,” denotes the skip-connections from the Transformer blocks (indicated by orange arrows in Fig. 1); “lm. positional embedding” denotes the learnable positional embedding; “sin. positional embedding” denotes the sinusoidal positional embedding.

| Model | Conv. skip. | Trans. skip. | Parameters (M) |
|-----------------------------|-------------|--------------|----------------|
| w/o conv. skip. | ✓ | – | 46.70 |
| w/o Trans. skip. | – | ✓ | 41.55 |
| w/o positional embedding | ✓ | ✓ | 46.77 |
| w/shuffling | ✓ | ✓ | 46.77 |
| w/rel. positional bias | ✓ | ✓ | 46.77 |
| w/lm. positional embedding | ✓ | ✓ | 63.63 |
| w/sin. positional embedding | ✓ | ✓ | 46.77 |

given moving image. Note that the only change between the methods below and VoxelMorph is the network architecture, with the spatial transformation function, loss function, and network training procedures remaining the same. The first three models used the hybrid Transformer-ConvNet architecture (i.e., ViT-V-Net, PVT, and CoTr), while the last model used a pure Transformer-based architecture (i.e., nnFormer). Their network hyperparameter settings were as follows:

- ViT-V-Net¹² (Chen et al., 2021a): This registration network was developed based on ViT (Dosovitskiy et al., 2020). We applied the default network hyperparameter settings suggested in Chen et al. (2021a).
- PVT¹³ (Wang et al., 2021c): The default settings were applied, except that the embedding dimensions were to be {20, 40, 200, 320}, the number of heads was set to {2, 4, 8, 16}, and the depth was increased to {3, 10, 60, 3} to achieve a comparable number of parameters to that of TransMorph.
- CoTr¹⁴ (Xie et al., 2021): We used the default network settings for all registration tasks.
- nnFormer¹⁵ (Zhou et al., 2021): Because nnFormer was also developed on the basis of Swin Transformer, we applied the same Transformer hyperparameter values as in TransMorph to make a fair comparison.

4.3. Implementation details

The proposed TransMorph was implemented using PyTorch (Paszke et al., 2019) on a PC with an NVIDIA TITAN RTX GPU and an NVIDIA RTX3090 GPU. All models were trained for 500 epochs using the Adam optimization algorithm, with a learning rate of 1×10^{-4} and a batch size of 1. The brain MR dataset was augmented with flipping in random directions during training, while no data augmentation was applied to the CT dataset. Restricted by the sizes of the image volumes, the window sizes (i.e., $\{M_x, M_y, M_z\}$) used in Swin Transformer were set to {5, 6, 7} for MR brain registration, {5, 5, 5} for XCAT-to-CT registration, respectively. The Transformer hyperparameter settings for TransMorph are listed in the first row of Table 2. Note that the variants of TransMorph (i.e., TransMorph-Bayes, TransMorph-bspl, and TransMorph-diff) share the same Transformer settings as TransMorph. The hyperparameter settings for each proposed variant are described as follows:

- TransMorph: The identical loss function parameters as VoxelMorph were used for all tasks.

⁷ <https://github.com/brianlee324/torch-lddmm>

⁸ <http://voxelmorph.csail.mit.edu>

⁹ <http://voxelmorph.csail.mit.edu>

¹⁰ https://github.com/boahK/MEDIA_CycleMorph

¹¹ <https://github.com/qiuahuqi/midir>

¹² <https://bit.ly/3bWDynR>

¹³ <https://github.com/whai362/PVT>

¹⁴ <https://github.com/YtongXie/CoTr>

¹⁵ <https://github.com/282857341/nnFormer>

Table 2

The architecture hyperparameters of the TransMorph models used in the ablation study. “Embed. Dimension” denotes the embedding dimension, C , in the very first stage (described in Section 3.2), “Swin-T”. denotes Swin Transformer.

| Model | Embed. Dimension | Swin-T. block numbers | Head numbers | Parameters (M) |
|------------------|------------------|-----------------------|---------------|----------------|
| TransMorph | 96 | {2, 2, 4, 2} | {4, 4, 8, 8} | 46.77 |
| TransMorph-tiny | 6 | {2, 2, 4, 2} | {4, 4, 8, 8} | 0.24 |
| TransMorph-small | 48 | {2, 2, 4, 2} | {4, 4, 4, 4} | 11.76 |
| TransMorph-large | 128 | {2, 2, 12, 2} | {4, 4, 8, 16} | 108.34 |
| VoxelMorph-huge | – | – | – | 63.25 |

- **TransMorph-Bayes:** The identical loss function parameters as VoxelMorph were applied here for all tasks. The dropout probability was set to 0.15.
- **TransMorph-bspl:** The loss function settings for all tasks were the same ones as those used in VoxelMorph. The control point spacing, δ , for B-spline transformation was also set to 2, the same value used in MIDIR.
- **TransMorph-diff:** We applied the same loss function parameters as those used in VoxelMorph-diff.

The affine model presented in this work comprises of a compact Swin Transformer. The Transformer parameter settings were identical to TransMorph except that the embedding dimension was set to be 12, the numbers of Swin Transformer block were set to be {1, 1, 2, 2}, and the head numbers were set to be {1, 1, 2, 2}. The resulting affine model has a total number of 19.55 millions of parameters and a computational complexity of 0.4 GMacs. Because the MRI datasets were affinely aligned as part of the preprocessing, the affine model was only used in the XCAT-to-CT registration.

4.4. Additional studies

In this section, we present experiments designed to verify the effect of the various Transformer modules in TransMorph architecture. Specifically, we carried out two additional studies of network components and model complexity. They are performed using the validation datasets from the three registration tasks, and the system-level comparisons are reported on test datasets. The following subsections provide detailed descriptions of these studies.

4.4.1. Ablation study on network components

We begin by examining the effects of several network components on registration performance. Table 1 lists three variants of TransMorph that either keep or remove the network’s long skip connections or the positional embeddings in the Transformer encoder. In “w/o conv. skip”, the long skip connections from the two convolutional layers were removed (including two convolutional layers), which are the green arrows in Fig. 1. In “w/o trans. skip.”, the long skip connections coming from the Swin Transformer blocks were removed, which are the orange arrows in Fig. 1. We claimed in Section 3.2 that the positional embedding (i.e., E_{pos} in Eq. (8)) was not a necessary element of TransMorph, because the positional information of tokens can be learned implicitly in the network via the consecutive up-sampling in the decoder and backpropagating the loss between output and target. Here, we conducted experiments to study the effectiveness of positional embeddings. Table 1 also lists five variants of TransMorph that either keep or remove the positional embeddings in the Transformer encoder. In the third variation, “w/o positional embedding”, we did not employ any type of positional embedding. In the fourth variant, “w/ shuffling”, we did not employ any positional embedding but instead randomly shuffled the positions of the tokens (i.e., the dimension N of \mathbf{z} in Eqs. (8) and (9)) just before the self-attention calculation. Following the self-attention calculation, the positions are permuted back into their original order. This way, the self-attention modules in the Transformer

Table 3

Quantitative evaluation results of the inter-patient (i.e., the JHU dataset) and the atlas-to-patient (i.e., the IXI dataset) brain MRI registration (left table), and XCAT-to-CAT registration (right table). Dice score and percentage of voxels with a non-positive Jacobian determinant (i.e., folded voxels) are evaluated for different methods. The **bolded** numbers denote the highest scores, while the *italicized* ones indicate the second highest.

| Model | Inter-patient MRI | | Atlas-to-patient MRI | |
|------------------|----------------------|------------------------|----------------------|------------------------|
| | DSC | % of $ J_\phi \leq 0$ | DSC | % of $ J_\phi \leq 0$ |
| Affine | 0.572 ± 0.166 | – | 0.386 ± 0.195 | – |
| SyN | 0.729 ± 0.127 | <0.0001 | 0.645 ± 0.152 | <0.0001 |
| NiftyReg | 0.723 ± 0.131 | 0.061 ± 0.093 | 0.645 ± 0.167 | 0.020 ± 0.046 |
| LDDMM | 0.716 ± 0.131 | <0.0001 | 0.680 ± 0.135 | <0.0001 |
| deedsBCV | 0.719 ± 0.130 | 0.253 ± 0.110 | 0.733 ± 0.126 | 0.147 ± 0.050 |
| VoxelMorph-1 | 0.718 ± 0.134 | 0.426 ± 0.231 | 0.729 ± 0.129 | 1.590 ± 0.339 |
| VoxelMorph-2 | 0.723 ± 0.132 | 0.389 ± 0.222 | 0.732 ± 0.123 | 1.522 ± 0.336 |
| VoxelMorph-diff | 0.715 ± 0.137 | <0.0001 | 0.584 ± 0.165 | <0.0001 |
| CycleMorph | 0.719 ± 0.134 | 0.231 ± 0.168 | 0.737 ± 0.123 | 1.719 ± 0.382 |
| MIDIR | 0.710 ± 0.132 | <0.0001 | 0.742 ± 0.128 | <0.0001 |
| ViT-V-Net | 0.729 ± 0.128 | 0.402 ± 0.249 | 0.734 ± 0.124 | 1.609 ± 0.319 |
| PVT | 0.729 ± 0.130 | 0.427 ± 0.254 | 0.727 ± 0.128 | 1.858 ± 0.314 |
| CoTr | 0.725 ± 0.131 | 0.415 ± 0.258 | 0.735 ± 0.135 | 1.292 ± 0.342 |
| nnFormer | 0.729 ± 0.128 | 0.399 ± 0.234 | 0.747 ± 0.135 | 1.595 ± 0.358 |
| TransMorph-Bayes | 0.744 ± 0.125 | 0.389 ± 0.241 | 0.753 ± 0.123 | 1.560 ± 0.333 |
| TransMorph-diff | 0.730 ± 0.129 | <0.0001 | 0.594 ± 0.163 | <0.0001 |
| TransMorph-bspl | 0.740 ± 0.123 | <0.0001 | 0.761 ± 0.122 | <0.0001 |
| TransMorph | 0.745 ± 0.125 | 0.396 ± 0.240 | 0.754 ± 0.124 | 1.579 ± 0.328 |

Table 4

Quantitative evaluation results of XCAT-to-CT registration. Dice score of 16 organs, percentage of voxels with a non-positive Jacobian determinant (i.e., folded voxels), and SSIM are evaluated for different methods. The **bolded** numbers denote the highest scores, while the *italicized* ones indicate the second highest.

| Model | DSC | % of $ J_\phi \leq 0$ | SSIM |
|--------------------|----------------------|------------------------|----------------------|
| w/o registration | 0.220 ± 0.242 | – | 0.576 ± 0.071 |
| Affine Transformer | 0.330 ± 0.291 | – | 0.751 ± 0.018 |
| SyN | 0.498 ± 0.342 | 0.001 ± 0.002 | 0.894 ± 0.021 |
| NiftyReg | 0.488 ± 0.333 | 0.025 ± 0.046 | 0.886 ± 0.027 |
| LDDMM | 0.519 ± 0.265 | 0.006 ± 0.007 | 0.874 ± 0.031 |
| deedsBCV | 0.568 ± 0.306 | 0.126 ± 0.123 | 0.863 ± 0.029 |
| VoxelMorph-1 | 0.532 ± 0.313 | 2.275 ± 1.283 | 0.899 ± 0.027 |
| VoxelMorph-2 | 0.548 ± 0.317 | 1.696 ± 0.909 | 0.910 ± 0.027 |
| VoxelMorph-diff | 0.526 ± 0.330 | <0.0001 | 0.911 ± 0.020 |
| CycleMorph | 0.528 ± 0.321 | 3.263 ± 1.188 | 0.909 ± 0.024 |
| MIDIR | 0.551 ± 0.303 | <0.0001 | 0.896 ± 0.022 |
| ViT-V-Net | 0.582 ± 0.311 | 2.109 ± 1.032 | 0.915 ± 0.020 |
| PVT | 0.516 ± 0.321 | 2.939 ± 1.162 | 0.900 ± 0.027 |
| CoTr | 0.550 ± 0.313 | 1.530 ± 1.052 | 0.905 ± 0.029 |
| nnFormer | 0.536 ± 0.315 | 1.371 ± 0.620 | 0.902 ± 0.024 |
| TransMorph-Bayes | 0.594 ± 0.313 | 1.475 ± 0.857 | 0.919 ± 0.024 |
| TransMorph-diff | 0.541 ± 0.324 | <0.0001 | 0.910 ± 0.025 |
| TransMorph-bspl | 0.575 ± 0.311 | <0.0001 | 0.908 ± 0.025 |
| TransMorph | 0.604 ± 0.314 | 1.679 ± 0.772 | 0.918 ± 0.023 |

encoder are truly invariant to the order of the tokens. In the fifth variant, “w/rel. positional bias”, we used the relative positional bias in the self-attention computation (i.e. B in Eq. (10)) as used in the Swin Transformer (Liu et al., 2021a). In the second to last variant, “w/lrn. positional embedding”, we added the same learnable positional embedding to the patch embeddings at the start of the Transformer encoder as used in the ViT (Dosovitskiy et al., 2020) while keeping the relative positional bias. In the last variant, “w/sin. positional embedding”, we substituted the learnable positional embedding with a sinusoidal positional embedding, the same embedding used in the original Transformer (Vaswani et al., 2017), which hardcodes the positional information in the tokens.

4.4.2. Model complexity study

The impact of model complexity on registration performance was also investigated in this paper. Table 2 listed the parameter settings and the number of trainable parameters of four variants of the proposed TransMorph model. In the base model, TransMorph, the embedding dimension C was set to 96, and the number of Swin Transformer blocks in the four stages of the encoder was set to 2, 2, 4, and 2, respectively. Additionally, we introduced TransMorph-tiny, TransMorph-small, and TransMorph-large, which are about 1/200x, 1/4x, and 2x the

Table 5

Quantitative evaluation results for brain MRI registration of the OASIS dataset from the 2021 Learn2Reg challenge task 3. Dice score of 35 cortical and subcortical brain structures, the 95th percentile percentage of the Hausdorff distance, and the standard deviation of the logarithm of the Jacobian determinant (SDlogJ) of the displacement field are evaluated for different methods. The validation results came from the challenge's leaderboard, whereas the test results came directly from the challenge's organizers. The **bolded** numbers denote the highest scores, while the *italicized* ones indicate the second highest.

| Validation | | | |
|-----------------------|----------------------|----------------------|----------------------|
| Model | DSC | HdDist95 | SDlogJ |
| Lv et al. (2022) | 0.827 ± 0.013 | 1.722 ± 0.318 | 0.121 ± 0.015 |
| Siebert et al. (2021) | 0.846 ± 0.016 | 1.500 ± 0.304 | 0.067 ± 0.005 |
| Mok and Chung (2021) | 0.861 ± 0.015 | 1.514 ± 0.337 | 0.072 ± 0.007 |
| VoxelMorph-huge | 0.847 ± 0.014 | 1.546 ± 0.306 | 0.133 ± 0.021 |
| TransMorph | 0.858 ± 0.014 | <i>1.494 ± 0.288</i> | 0.118 ± 0.019 |
| TransMorph-Large | 0.862 ± 0.014 | 1.431 ± 0.282 | 0.128 ± 0.021 |
| Test | | | |
| Model | DSC | HdDist95 | SDlogJ |
| Initial | 0.56 | 3.86 | – |
| Lv et al. (2022) | 0.80 | 1.77 | <i>0.08</i> |
| Siebert et al. (2021) | 0.81 | 1.63 | 0.07 |
| Mok and Chung (2021) | 0.82 | 1.67 | 0.07 |
| TransMorph | 0.816 | 1.692 | 0.124 |
| TransMorph-Large | 0.820 | 1.656 | 0.124 |

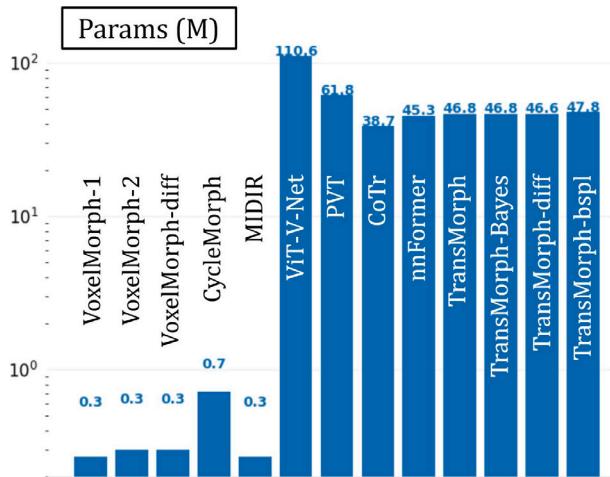


Fig. 7. The number of parameters in each deep-learning-based model. The values are in units of millions of parameters.

model size of TransMorph. Finally, we compared our model to a customized VoxelMorph (denoted VoxelMorph-huge), which has a comparable parameter size to that of TransMorph w/ln. positional embedding. Specifically, we maintained the same number of layers in VoxelMorph-huge as in VoxelMorph, but increased the number of convolution kernels in each layer. As a result, VoxelMorph-huge has 63.25 million trainable parameters.

4.5. Evaluation metrics

The registration performance of each model was evaluated based on the volume overlap between anatomical/organ segmentation, which was quantified using the Dice score (Dice, 1945). We averaged the Dice scores of all anatomical/organ structures for all patients. The mean and standard deviation of the averaged scores were compared across various registration methods.

To quantify the regularity of the deformation fields, we also reported the percentages of non-positive values in the determinant of the Jacobian matrix on the deformation fields (i.e., $|J_\phi| \leq 0$).

Additionally, for XCAT-to-CT registration, we used the structural similarity index (SSIM) (Wang et al., 2004) to quantify the structural difference between the deformed XCAT and the target CT images. The mean and standard deviation of the SSIM values of all patients were reported and compared.

5. Results

5.1. Inter-patient brain MRI registration

The top-left panel of Fig. 8 shows the qualitative results of a sample slice for inter-patient brain MRI registration. The scores in blue, orange, green, and pink correspond to ventricles, third ventricle, thalamus, and hippocampi, respectively. Additional qualitative comparisons across all methods are shown in Fig. C.20 in Appendix C. Among the proposed models, diffeomorphic variants (i.e., TransMorph-diff and TransMorph-bspl) generated smoother displacement fields, with TransMorph-bspl producing the smoothest deformations inside the brain area. On the other hand, TransMorph and TransMorph-Bayes showed better qualitative results (highlighted by the yellow arrows) with higher Dice scores for the delineated structures.

The quantitative evaluations are shown in Table 3. The results presented in the table show that the proposed method, TransMorph, achieved the highest mean Dice score of 0.745. Although the diffeomorphic variants produced slightly lower Dice scores than TransMorph, they still outperformed the existing registration methods and generated almost no foldings (i.e., $\sim 0\% \text{ of } |J_\phi| \leq 0$) in the deformation fields. By comparison, TransMorph improved Dice score by >0.2 when compared to VoxelMorph and CycleMorph. We found that the Transformer-based models (i.e., TransMorph, ViT-V-Net, PVT, CoTr, and nnFormer) generally produced better Dice scores than the ConvNet-based models. Note that even though ViT-V-Net had almost twice the number of the trainable parameters (as shown in Fig. 7), TransMorph still outperformed all the Transformer-based models (including ViT-V-Net) by at least 0.1 in the Dice score, demonstrating Swin-Transformer's superiority over other Transformer architectures. When we conducted hypothesis testing on the results using the paired *t*-test with Bonferroni correction (Armstrong, 2014) (i.e., dividing the *p*-values by 13, the total number of the paired *t*-tests performed), the *p*-values between the best performing TransMorph variant (i.e., TransMorph) and all other methods were $p \ll 0.0005$.

Figs. C.21 and C.22 show additional Dice results for a variety of anatomical structures, with Fig. C.21 comparing TransMorph to current registration techniques (both optimization- and learning-based methods), and Fig. C.22 comparing the Dice scores between the Transformer-based models.

5.2. Atlas-to-patient brain MRI registration

The top-right panel of Fig. 8 shows the qualitative results of the TransMorph variants on a sample MRI slice for atlas-to-patient brain MRI registration. As highlighted by the yellow arrows, the diffeomorphic variants resulted in the deformed images that were less comparable to the fixed image in terms of visual appearance. In contrast, the variants without diffeomorphic deformations (i.e., TransMorph and TransMorph-Bayes) produced better qualitative results, with the sulci in the deformed atlas images more closely matching those in the fixed image. Additional qualitative comparisons are shown in Fig. D.23 in Appendix D, where we observed that all the learning-based methods yielded more detailed and precise deformation fields than the conventional methods. This might be owing to the high parameterization of the DNNs, which enables the modeling of more complicated deformations.

Table 3 shows the quantitative evaluation results of the atlas-to-patient registration. The highest mean Dice score of 0.761 was achieved by the proposed TransMorph-bspl with nearly no folded voxels.

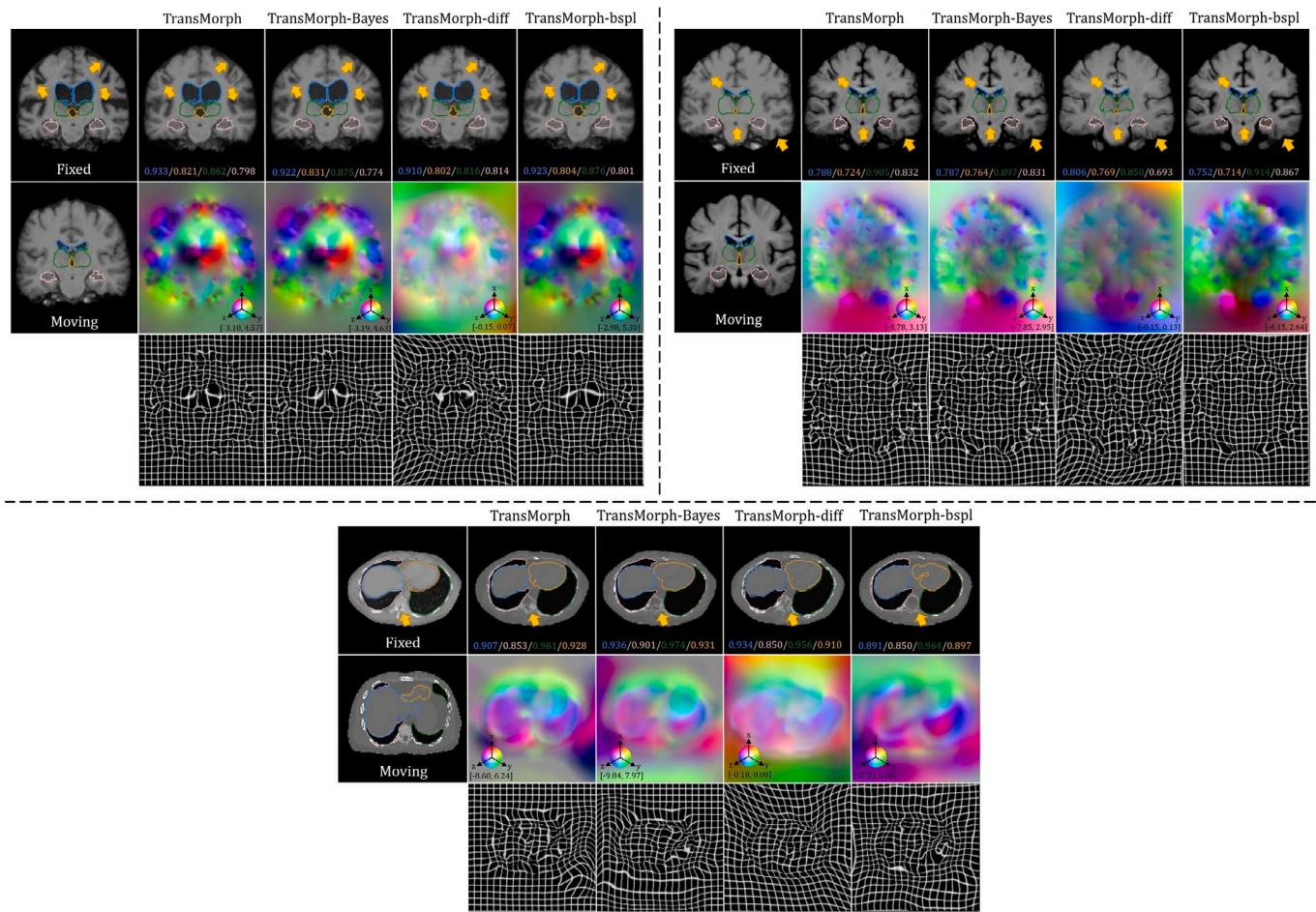


Fig. 8. Qualitative results of TransMorph (2nd column) and its Bayesian- (3rd column), probabilistic- (4th column), and B-spline (5th column) variants. Top-left & Top-right panels: Results of inter-patient and atlas-to-patient brain MRI registration. The blue, orange, green, and pink contours define, respectively, the ventricles, third ventricle, thalamus, and hippocampi. Bottom panel: Results of XCAT-to-CT registration. The blue, orange, green, and pink contours define, respectively, the liver, heart, left lung, and right lung. The second row in both panels exhibits the displacement fields \mathbf{u} , where spatial dimension x , y , and z is mapped to each of the RGB color channels, respectively. The $[p, q]$ in color bars denotes the magnitude range of the fields.

The second best Dice score of 0.754 was achieved by both TransMorph and TransMorph-Bayes, while TransMorph-Bayes yielded a smaller standard deviation. In comparison to these TransMorph variants, TransMorph-diff produced a lower Dice score of 0.594. However, note that this score is still higher (~ 0.02) than the one produced by VoxelMorph-diff, which is the base model of TransMorph-diff. Additionally, we observed that the registration methods that used MSE for training or optimization resulted in lower Dice scores (i.e., SyN, NiftyReg, LDDMM, VoxelMorph-diff, and TransMorph-diff). This was most likely due to the significant disparity in the intensity values of brain sulci between the atlas and the patient MRI images. As seen in the top-right panel of Fig. 8, the sulci in the atlas image (i.e., the moving image) exhibited low-intensity values comparable to the background, but the sulci in the patient MRI image had intensity values more comparable to the neighboring gyri. Thus, the discrepancies in the sulci intensity values may account for the majority of the MSE loss during training, compelling the registration models to fill the sulci in the atlas image with other brain structures (as shown in Fig. D.23, these models produced significantly smaller sulci than models trained with LNCC), thereby limiting registration performance. The paired t -tests with Bonferroni correction (Armstrong, 2014) revealed the p -values of $p \ll 0.0005$ between the best performing model (i.e., TransMorph-bspl) and all other methods. This indicates that the proposed method outperformed the comparative registration methods and network architectures.

A detailed breakdown of Dice scores for a variety of anatomical structures is shown in Figs. D.24 and D.25 in Appendix D.

5.3. Learn2Reg OASIS brain MRI registration

Table 5 shows the quantitative results of the validation and test sets of the challenge. The validation scores of the various methods were obtained from the leaderboard of the challenge, whilst the test scores were obtained directly from the organizers. TransMorph performed similarly to the best-performing method (LapIRN (Mok and Chung, 2021)) of the challenge on the validation set, where TransMorph-large achieved the best mean Dice score of 0.862 and mean HdDist95 of 1.431. VoxelMorph-huge performed significantly poor than TransMorph, with a p -value less than 0.01 from paired t -test. This reveals the superiority of Transformer-based architecture over ConvNet despite having a comparable number of parameters. On the test set, the TransMorph and TransMorph-large achieved comparable mean Dice score to that of LapIRN. Despite the comparable performance, LapIRN produced much smoother deformation fields as measured by SDlogJ. In a separate study, we presented a simple extension of TransMorph that significantly outperformed LapIRN while maintaining smooth deformation fields. We direct interested readers to Chen et al. (2022) for further details. Moreover, LapIRN employed a multiresolution framework in which three ConvNet registration backbones were involved in generating deformation fields at three different scales. TransMorph, however, operated on a single resolution. We underline

that TransMorph is a registration backbone, and that it may be easily adapted to LapIRN or any advanced registration frameworks.

5.4. XCAT-to-CT registration

The bottom panel of Fig. 8 shows the qualitative results for a representative CT slice. The blue, orange, green, and pink lines denote the liver, heart, left lung, and right lung, respectively, while the bottom values show the corresponding Dice scores. Similar to the findings in the previous sections, TransMorph and TransMorph-Bayes gave more accurate registration results (highlighted by the yellow arrows and the delineated structures), while the diffeomorphic variants produced smoother deformations. Additional qualitative comparisons are shown in Fig. E.26 in Appendix E. It is possible to see certain artifacts in the displacement field created by nnFormer (as shown in Fig. E.26); these were most likely caused by the patch operations of the Transformers used in its architecture. nnFormer is a near-convolution-free model (convolutional layers are employed only to form displacement fields). In contrast to the relatively small displacements in brain MRI registration, displacements in XCAT-to-CT registration may exceed the patch size. Consequently, the lack of convolutional layers to refine the stitched displacement field patches may have resulted in artifacts. Four example coronal slices of the deformed XCAT phantoms generated by various registration methods are shown in Fig. E.27 in Appendix E.

The quantitative evaluation results are presented in Table 4. They include Dice scores for all organs and scans, the percentage of non-positive Jacobian determinants, and the structural similarity index (SSIM) (Wang et al., 2004) between the deformed XCAT phantom and the target CT scan. The window size used in SSIM was set to 7. Without registration or affine transformation, a Dice score of 0.22 and an SSIM of 0.576 demonstrate the large dissimilarity between the original XCAT phantom and patient CT scans. The Dice score and SSIM increased to 0.33 and 0.751, respectively, after aligning the XCAT and patient CT using the proposed affine Transformer. Among the traditional registration methods, deedsBCV, which was initially designed for abdominal CT registration-based segmentation (Heinrich et al., 2015), achieved the highest Dice score of 0.568, which is even higher than most of the learning-based methods. Among the learning-based methods, Transformer-based models outperformed ConvNet-based models on average, which is consistent with the findings from the brain MR registration tasks. The *p*-values from the paired *t*-tests with Bonferroni correction (Armstrong, 2014) between TransMorph and all non-TransMorph methods were *p* ≪ 0.05. The proposed TransMorph models yielded the highest Dice and SSIM scores of all methods in general, with the best Dice of 0.604 given by TransMorph and the best SSIM of 0.919 given by TransMorph-Bayes. The diffeomorphic variants produced lower Dice and SSIM scores as a consequence of not having any folded voxels in the deformation.

Figs. E.28 and E.29 show additional boxplots of Dice scores on the various abdominal organs, with Fig. E.28 comparing TransMorph to current registration techniques (both optimization- and learning-based methods), and Fig. E.29 comparing the Dice scores between the Transformer-based models.

5.5. Ablation studies

Inter-patient registration. The first figure in the first row of Fig. 9 shows the violin plots of Dice scores from the ablation study on the validation dataset of inter-patient brain MR registration. When evaluating the effectiveness of skip connections, we observed that the skip connections from both the convolution and Transformer layers improved registration performance. TransMorph scored a mean Dice of 0.753 after the skip connections from the convolutional layers were removed. However, the score decreased to 0.740 when the skip connections from the Transformer blocks were removed. In comparison, the skip connections from convolutional layers were less effective, with a mean

Dice improvement of 0.003. Note that the TransMorph with shuffling, and with and without positional embeddings all generated comparable mean Dice scores and violin plots, suggesting that positional embedding may not be necessary.

Atlas-to-patient registration. The violin plots from the ablation study on the atlas-to-patient registration task are shown in the second figure in the first row of Fig. 9. Comparable violin plots with similar mean Dice scores around 0.752 were observed with and without the skip connections from the convolutional layers. When the skip connections from the Transformer blocks were removed, the Dice score decreased by 0.019, reflecting the effectiveness of these skip connections. Comparable violin plots and mean Dice scores around 0.750 were observed with shuffling, and with and without various positional embeddings, confirming that TransMorph's performance is unaffected by whether or not positional embedding was used.

XCAT-to-CT registration. The second to last figure in the first row of Fig. 9 shows the violin plots from the validation dataset of XCAT-to-CT registration task. Without the skip connections from the convolution and Transformer layers, the Dice scores dropped by 0.013 and 0.016, respectively, when compared to TransMorph, further supporting the observation that skip connections can improve performance. Learnable and relative positional embeddings yielded comparable mean Dice scores for XCAT-to-CT registration in the range of 0.593. When sinusoidal positional embedding was employed, a score of 0.583 was attained, whereas a score of 0.588 was produced when the positions were shuffled. With a score of 0.600, without using positional embeddings yielded a slight improvement among other variants. The effect of each component is addressed in depth in the Discussion section (Section 6.1).

In conclusion, the results from all three tasks (i.e., inter-patient, atlas-to-patient, and XCAT-to-CT registration tasks) indicate that using skip connections improves performance. The results also reveal that with and without using positional embedding or even randomly shuffling the token positions provided similar performances. Additionally, we applied the TransMorph models to the test datasets of the three registration tasks for system-level comparisons, and the results are shown in the upper panel of Table 6. The scores on the test datasets followed the same trend as those on the validation datasets, where the positional embeddings had an insignificant influence on registration performance.

5.6. Computational complexity

The barplot in the left panel of Fig. 10 shows the computational complexity comparisons between the deep-learning-based registration models. The plot was created using an input image with a resolution of 160 × 192 × 224, the same size as the brain MRI images. The numbers were expressed in Giga multiply-accumulate operations (GMACs), with a higher value indicating a more computationally expensive model that may also be more memory intensive. The proposed model, TransMorph, and its Bayesian variant, TransMorph-Bayes, had a moderate computational complexity with 687 GMACs which is much less than CoTr and CycleMorph. In practice, the GPU memory occupied during training for TransMorph was about 15 GiB with a batch size of 1 and an input image size of 160 × 192 × 224. The diffeomorphic variants, TransMorph-diff and TransMorph-bspl, had 281 and 454 GMACs, which are comparable to that of the conventional ConvNet-based registration models, VoxelMorph-1 and -2. In practice, they occupied approximately 11 GiB of GPU memory during training, which is a size that can be readily accommodated by the majority of modern GPUs. In terms of the number of parameters, all ConvNet-based models had fewer than 1M network parameters (as shown in Fig. 7); yet their GMACs (i.e., computational complexity) were comparable to TransMorph, but their registration performances were significantly inferior. Transformer-based models were all of large scale, with more than

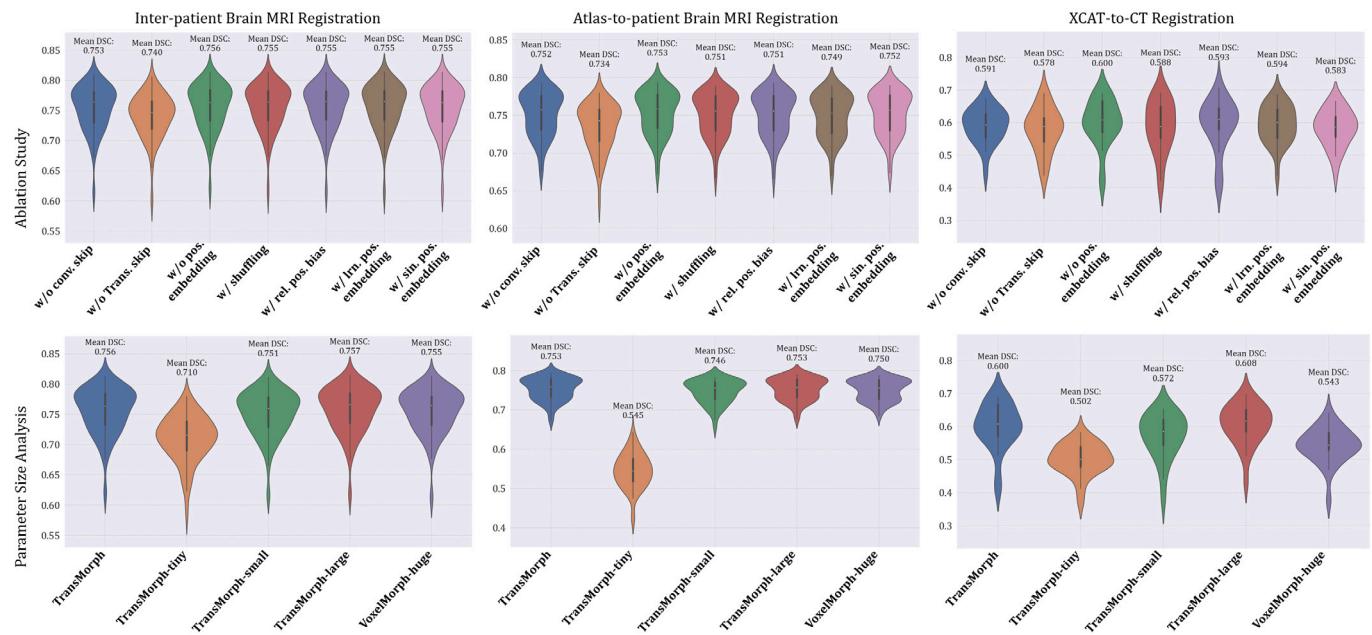


Fig. 9. Quantitative evaluation results of the additional studies performed on the validation datasets of the two brain MRI and XCAT-to-CT registration tasks.

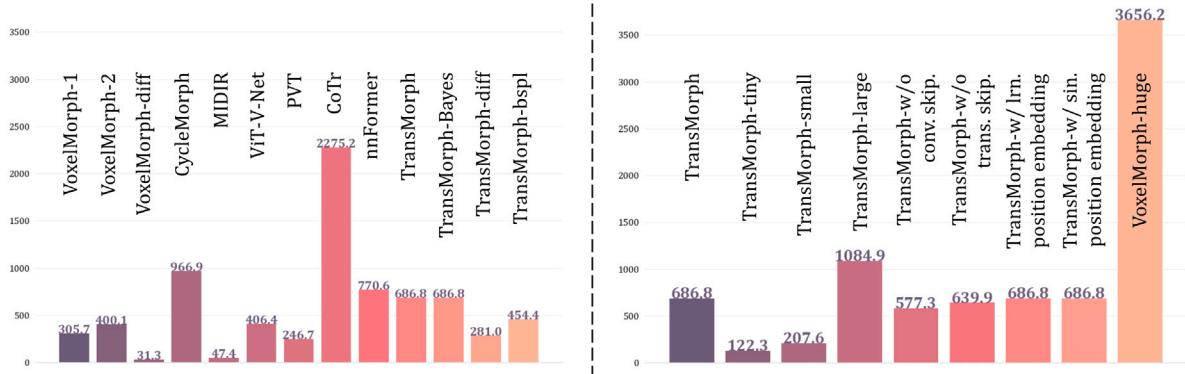


Fig. 10. Model computational complexity comparisons represented in Giga multiply-accumulate operations (GMACs). Greater values imply a greater degree of computational complexity. These values were obtained using an input image of size $160 \times 192 \times 224$.

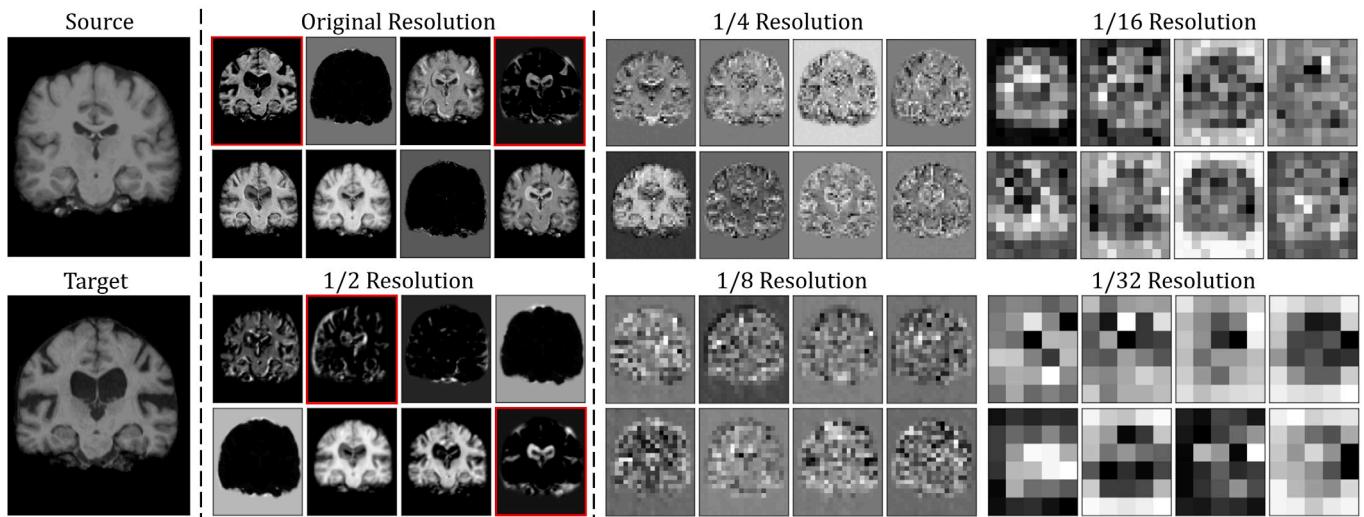


Fig. 11. Examples of feature maps in TransMorph's skip connections. Eight feature maps are randomly selected from the feature maps associated with each skip connection. Left panel: Example 2D slices of source and target images (i.e., I_m and I_f), which are used as inputs to TransMorph. Middle panel: Feature maps in the skip connections of the two convolutional layers (denoted by the green arrows in Fig. 1). Right panel: Feature maps in the skip connections of the Swin Transformer blocks (denoted by the orange arrows in Fig. 1).

Table 6

System-level comparison of various TransMorph designs and the customized VoxelMorph on the test datasets of inter-patient MRI and XCAT-to-CT registration tasks. “Val. DSC” denotes the Dice scores on the validation dataset; “Test DSC” denotes the system-level comparison of the Dice scores on the test dataset. The **bolded** numbers denote the highest scores, while the *italicized* ones indicate the second highest.

| Model | Inter-patient MRI | | Atlas-to-patient MRI | | XCAT-to-CT | |
|-----------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | Val. DSC | Test DSC | Val. DSC | Test DSC | Val. DSC | Test DSC |
| w/o conv. skip. | 0.753 ± 0.119 | 0.743 ± 0.124 | 0.752 ± 0.129 | 0.754 ± 0.125 | 0.591 ± 0.319 | 0.586 ± 0.314 |
| w/o Trans. skip. | 0.740 ± 0.124 | 0.727 ± 0.130 | 0.734 ± 0.127 | 0.736 ± 0.125 | 0.578 ± 0.315 | 0.588 ± 0.314 |
| w/shuffling | 0.755 ± 0.119 | 0.744 ± 0.125 | 0.751 ± 0.127 | $\textbf{0.754} \pm \textbf{0.123}$ | 0.588 ± 0.314 | 0.597 ± 0.310 |
| w/rel. positional bias | 0.755 ± 0.120 | 0.742 ± 0.125 | 0.751 ± 0.131 | 0.753 ± 0.127 | 0.593 ± 0.315 | 0.592 ± 0.319 |
| w/lrn. positional embedding | 0.755 ± 0.120 | 0.744 ± 0.125 | 0.749 ± 0.131 | 0.751 ± 0.129 | 0.594 ± 0.315 | 0.586 ± 0.315 |
| w/sin. positional embedding | 0.755 ± 0.120 | 0.744 ± 0.125 | 0.752 ± 0.126 | $\textbf{0.754} \pm \textbf{0.123}$ | 0.583 ± 0.320 | 0.572 ± 0.317 |
| TransMorph | 0.756 ± 0.119 | 0.745 ± 0.125 | $\textbf{0.753} \pm \textbf{0.127}$ | 0.754 ± 0.124 | 0.600 ± 0.317 | 0.604 ± 0.314 |
| TransMorph-tiny | 0.710 ± 0.132 | 0.696 ± 0.140 | 0.545 ± 0.180 | 0.543 ± 0.180 | 0.502 ± 0.311 | 0.501 ± 0.312 |
| TransMorph-small | 0.751 ± 0.121 | 0.740 ± 0.126 | 0.746 ± 0.128 | 0.747 ± 0.125 | 0.572 ± 0.320 | 0.570 ± 0.318 |
| TransMorph-large | $\textbf{0.757} \pm \textbf{0.119}$ | $\textbf{0.746} \pm \textbf{0.124}$ | 0.753 ± 0.130 | 0.754 ± 0.128 | $\textbf{0.608} \pm \textbf{0.305}$ | $\textbf{0.611} \pm \textbf{0.311}$ |
| VoxelMorph-huge | 0.755 ± 0.119 | 0.744 ± 0.124 | 0.750 ± 0.133 | 0.751 ± 0.130 | 0.543 ± 0.320 | 0.550 ± 0.319 |

30M parameters. Notably, ViT-V-Net and PVT had around 2x and 1.5x more parameters than TransMorph, nevertheless TransMorph outperformed them by a significant margin on all of the evaluated registration tasks. This demonstrates that the success of TransMorph owes not just to the large model size but also to the architecture itself.

Fig. 9 shows the quantitative results of TransMorph models with various architectural settings and the customized ConvNet-based model VoxelMorph-huge on the validation datasets of the three registration datasets. When parameter size is the only variable in TransMorph models, there is a strong correlation between model complexity (as shown in the right panel of Fig. 10) and registration performance. TransMorph-tiny produced the lowest mean Dice of 0.710, 0.545, and 0.502 on the validation set of the three registration tasks, respectively. The Dice score steadily improves as the complexity of the model increases. Note that for inter-patient and atlas-to-patient brain MRI registration (the first and second figures in the bottom row of Fig. 9), the improvement in mean Dice score from TransMorph to TransMorph-large were mostly under 0.01 but the latter was almost twice as computationally costly (as shown in the right panel of Fig. 10). The customized ConvNet-based model, VoxelMorph-huge, had the comparable number of parameters as TransMorph. However, it achieved slightly lower mean Dice scores than those of TransMorph for the JHU and IXI brain MR registration tasks, and significantly lower scores for OASIS brain MR and the XCAT-to-CT registration task. This further indicates the architectural advantages of TransMorph for image registration. A significant disadvantage of VoxelMorph-huge was its computational complexity, with 3656 GMACs (as seen in the right panel of Fig. 10), it was nearly five times as computationally expensive as TransMorph, making it memory-intensive (~ 22 GiB for a patch size of 1 during training) and slow to train in practice. However, TransMorph was able to accommodate a larger number of parameters without significantly increasing computational complexity. The promising performances brought by the larger scale of parameters demonstrate the superior scaling property of Transformer-based models as described in Zhai et al. (2022), Liu et al. (2022). The TranMorph models with different model parameter settings and VoxelMorph-huge were applied to the test datasets for system-level comparisons, and the results are shown in the bottom panel of Table 6.

6. Discussion

6.1. Network components in TransMorph

6.1.1. Skip connections

As previously shown in Section 5.5, skip connections may aid in enhancing registration accuracy. In this section, we give further insight into the skip connections’ functionality.

Fig. 11 shows some example feature maps in each skip connection (a full feature map visualization is shown in Fig. G.31 in Appendix).

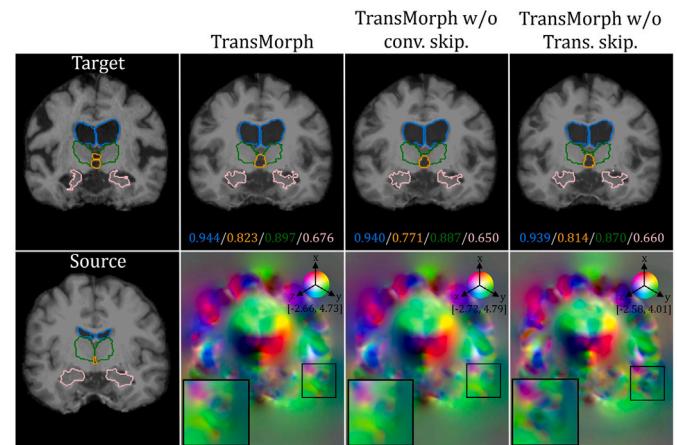


Fig. 12. Qualitative impact of skip connections on the deformation fields. The spatial dimension x , y , and z in the displacement field is mapped to each of the RGB color channels, respectively. The $[p, q]$ in color bars denotes the magnitude range of the fields.

Specifically, the left panel shows sample slices of the input volumes; the center panel illustrates selected feature maps in the skip connections of the convolutional layers, and the right panel illustrates selected feature maps in the skip connections of the Swin Transformer blocks. As seen from these feature maps that the Swin Transformer blocks provided more abstract information (right panel in Fig. 11), in comparison to the convolutional layers (middle panel in Fig. 11). Since a Transformer divides an input image volume into patches to create tokens for self-attention operations (as described in Section 3.2), it can only deliver information up to a certain resolution, which is often a factor of the patch size lower than the original resolution (i.e., $\frac{H}{P} \times \frac{W}{P} \times \frac{L}{P}$, and $P = 4$ in our case). On the other hand, the convolutional layers resulted in higher resolution feature maps with more detailed and human-readable information (e.g., edge and boundary information). Certain feature maps even revealed distinctions between the moving and fixed images (highlighted by the red boxes). Fig. 12 shows the qualitative comparisons between the proposed model with and without a specific type of skip connection. As seen by the magnified areas, TransMorph with both skip connection types provided a more detailed and accurate displacement field. Therefore, adding the skip connections from the convolutional layers is still recommended, although the actual Dice improvement were subtle on the validation datasets (0.003 for inter-patient brain MRI, 0.001 for atlas-to-patient brain MRI, and 0.009 for XCAT-to-CT registration).

6.1.2. Positional embedding

Transformers in computer vision were initially designed for image classification tasks (Dosovitskiy et al., 2020; Liu et al., 2021a; Dong

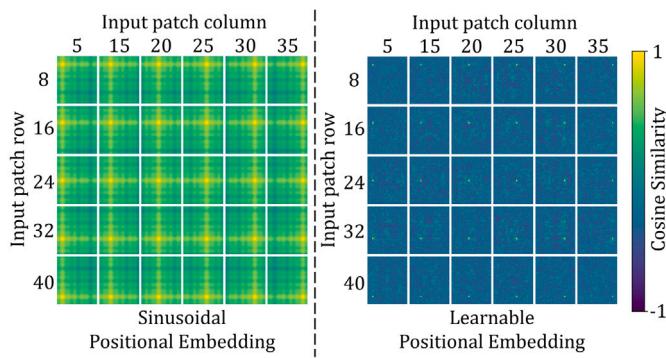


Fig. 13. Example slice of the positional embeddings used in TransMorph. Left panel: Sinusoidal positional embedding. Right panel: Learnable positional embedding. Tiles in both panels show the cosine similarities between the position embedding of the token with the indicated row and column and the position embeddings of all other tokens.

et al., 2021; Wang et al., 2021c). Such a Transformer produces a condensed probability vector that is not in the image domain but instead a description of the likelihood of being a certain class. The loss calculated based on this vector does not backpropagate any spatial information into the network. Thus, it is critical to encode positional information on the patched tokens; otherwise, as the network gets deeper, Transformer would lose track of the tokens' locations relative to the input image, resulting in unstable training and inferior predictions. However, for pixel-level tasks like image registration, the condensed features generated by Transformers are often subsequently expanded using a decoder whose output is an image with the same resolution as the input and target images. Any spatial mismatches between the output and target contributes to the loss, which is then backpropagated throughout the network. As a result, the Transformer implicitly learns the positional information of tokens, thus obviating the need for positional embedding. In this work, we compared the registration performance of TransMorph and TransMorph with positional embedding on brain MRI and XCAT-to-CT registration. The results shown in Section 5.5 indicated that positional embedding did not improve registration performance; rather, it introduced more parameters into the network. In this section, we discuss the positional embeddings in further detail.

Three positional embeddings were studied in this paper: sinusoidal (Vaswani et al., 2017), learnable (Dosovitskiy et al., 2020), and relative (Liu et al., 2021a) embeddings, which are also the major types of positional embedding. In sinusoidal positional embedding, the position of each patched token is represented by a value drawn from a predetermined sinusoidal signal according to the token's position relative to the input image. Whereas with learnable positional embedding, the network learns the representation of the token's location from the training dataset rather than giving a hardcoded value. The relative positional bias hardcodes the relative position relations between any two tokens in the dot product of the query and key representations (i.e., B in Eq. (10)). To validate that the network learned the positional information, (Dosovitskiy et al., 2020) computed the cosine similarities between a learned embedding of a token and that of all other tokens. The obtained similarity values were then used to form an image. If positional information is learned, the image should reflect increased similarities at the token's and nearby tokens' positions. Here, we computed the images of cosine similarities for both sinusoidal and learnable positional embeddings used in this work. The left and right panels in Fig. 13 show the images of cosine similarities. These images were generated based on an input image size of $160 \times 192 \times 224$ and a patch size of $4 \times 4 \times 4$ (resulting in $40 \times 48 \times 56$ patches). Each image has a size of 40×48 representing an image of cosine similarities in the plane of $z = 28$ (i.e., the middle slice). There should have been a total of 40×48 images in each panel. However, for better visualization, just

a few images were shown here. The images were chosen with step sizes of 5 and 8 in x and y direction, respectively, resulting in 6×5 images in each panel. As seen from the left panel, the images of sinusoidal embeddings exhibit a structured pattern, showing a high degree of correlation between tokens' relative locations and image intensity values. Note that the brightest pixel in each image represents the cosine similarity between a token's positional embedding and itself, which reflects the token's actual location relative to all other tokens. The similarity then gradually decreases as it gets farther away from the token. On the other hand, images generated with learnable embeddings (right panel of Fig. 13) lack such structured patterns, implying that the network did not learn the positional information associated with the tokens in the learnable embeddings. To further demonstrate that the network implicitly learned the positional information, we randomly shuffled the token positions when computing self-attention during training and testing. As a result, the self-attention modules could not explicitly perceive input tokens' positional information. However, as seen from the Dice scores in Fig. 9, regardless of shuffling and which positional embedding was employed, the mean Dice scores and violin plots were quite comparable to those produced without positional embedding. Thus, the findings confirmed that TransMorph learned the positional information of the tokens implicitly and that the learnable, sinusoidal, and relative positional embeddings were redundant in the model and had a negligible effect on registration performance.

6.2. Uncertainty quantification of TransMorph-Bayes

As previously mentioned in Section 3.4, the appearance uncertainty estimates produced by the predictive variance (Eq. (23)) were actually miscalibrated, meaning that the uncertainty values did not properly correlate to predicted model errors since variance was computed using the predictive mean instead of target image I_f . We proposed to directly use the expected model error to express appearance uncertainty since the target image is available at all times in image registration. Thus, the resulting appearance uncertainty estimate is perfectly calibrated. In this section, we examine how the proposed and existing methods differ in their estimates of appearance uncertainty.

To quantify the calibration error, we used the Uncertainty Calibration Error (UCE) introduced in Laves et al. (2020a), which is calculated on the basis of the binned difference between the expected model error (i.e., $\mathbb{E}[(I_m \circ \phi - I_f)^2]$) and the uncertainty estimation (e.g., $\hat{\Sigma}_f^2$ in Eq. (23) or Σ_f^2 in Eq. (25)). We refer the interested reader to the corresponding references for further details about UCE. The plots in the left panel of Fig. 14 exhibit the calibration plots and UCE obtained on four representative test sets. All results were based on a sample size of 25 (i.e., $T = 25$ in Eqs. (21), (23), and (25)) from 10 repeated runs. The blue lines show the results produced with the $\hat{\Sigma}_f^2$ and the shaded regions represent the standard deviation from the 10 runs, while the dashed black lines indicate the perfect calibration achieved with the proposed method. Notice that the uncertainty values obtained using $\hat{\Sigma}_f^2$ did not match well to the expected model error; in fact, they were consistently being underestimated (for reasons described in Section 3.4.1). In comparison, the proposed method enabled perfect calibration with $UCE = 0$ since its uncertainty estimate equaled the expected model error. In the right panel of Fig. 14, we show the visual comparisons of the uncertainty derived from Σ_f^2 and $\hat{\Sigma}_f^2$. When we compare either (e) to (f) or (k) to (l), we see that the former (i.e., (e) and (k)) captured more registration failures than the latter (as highlighted by the yellow arrows), indicating a stronger correlation between deformation uncertainty and registration failures. This is thus further evidence that the proposed method provides the perfect uncertainty calibration.

Despite the promising results, there are some limitations of using σ_f to estimate appearance uncertainty. In this work, we modeled σ_f as $\mathbb{E}[(I_m \circ \phi - I_f)^2]$, which is the MSE of the Monte Carlo sampled registration outputs relative to the fixed image. MSE, on the other

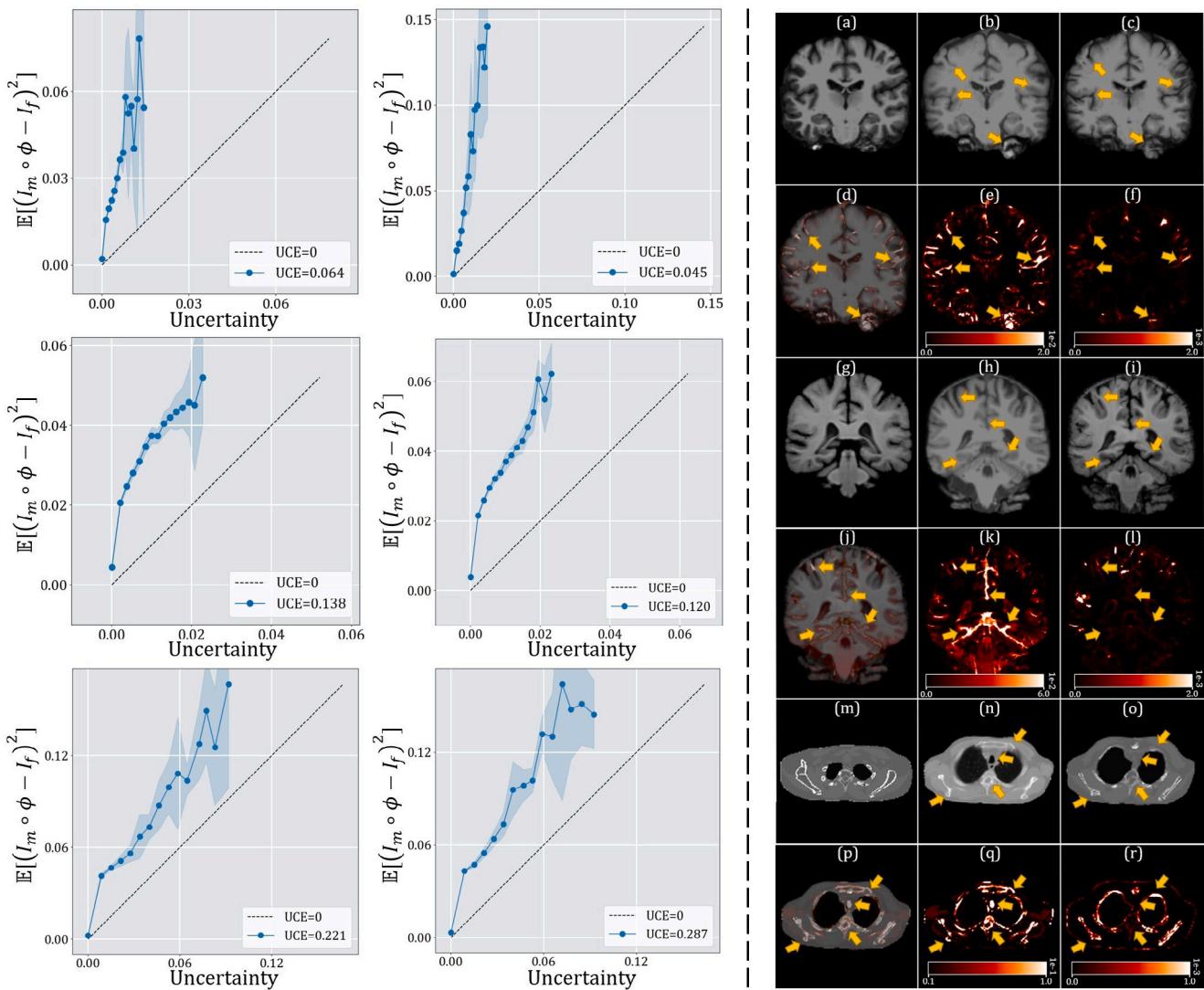


Fig. 14. Comparisons of the appearance uncertainty estimates derived from the predictive variance and the predicted model error. Left panel: Calibration plots and uncertainty calibration error (UCE) for TransMorph-Bayes on two inter-patient brain MR test sets (top), two atlas-to-patient brain MR test sets (middle), and two XCAT-to-CT test sets (bottom). The blue lines represent the results obtained using the uncertainty estimate Σ_f^2 . The dashed lines represent the perfect calibration, which are the results achieved when the uncertainty estimate is Σ_f^2 or $err(I_m \circ \phi)$ (i.e., the expected model error). The values are obtained from 10 repeated runs, and the shaded regions represent the standard deviation. Right panel: Visualization of the registration uncertainty on an inter-patient brain MRI test set (i.e., a–f), an atlas-to-patient brain MRI test set (i.e., g–l), and a CT test set (i.e., m–r). (a), (g), & (m): Moving image. (b), (h), & (n): Fixed image. (c), (i), & (o): Deformed moving image. (d), (j), & (p): Per-pixel uncertainty, represented by Σ_f^2 , overlays the deformed image. (e), (k), & (q): Per-pixel uncertainty given by Σ_f^2 (i.e., the proposed method). (f), (l), & (r): Per-pixel uncertainty given by $\hat{\Sigma}_f^2$. The yellow arrows highlight sites where Σ_f^2 identifies registration failures but $\hat{\Sigma}_f^2$ does not.

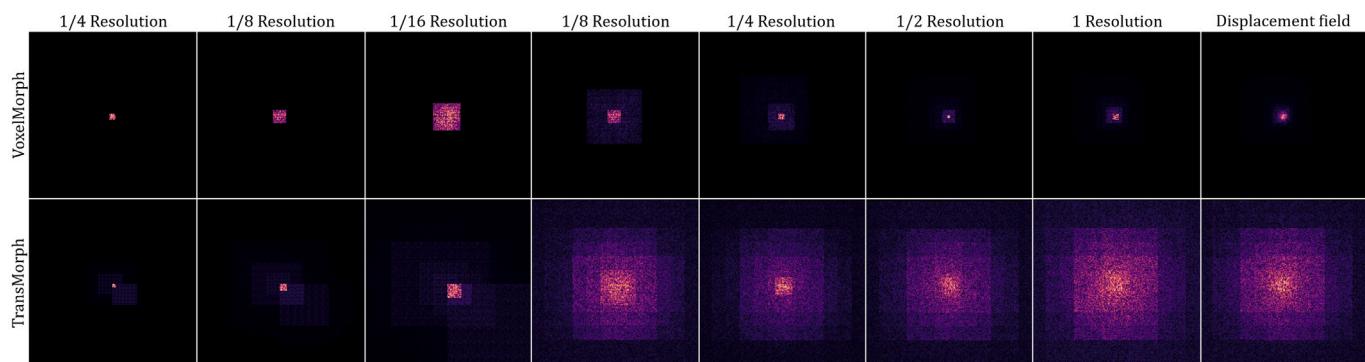


Fig. 15. Example ERFs of VoxelMorph and the proposed Transformer-based model TransMorph. The top row shows the ERF slices (i.e., $y = 80$) at each stage of the network on an input image size of $160 \times 160 \times 160$. For a consistent comparison of ERFs between VoxelMorph and TransMorph, the ERFs at 1/2 of VoxelMorph and 1/32 resolution of TransMorph were omitted.

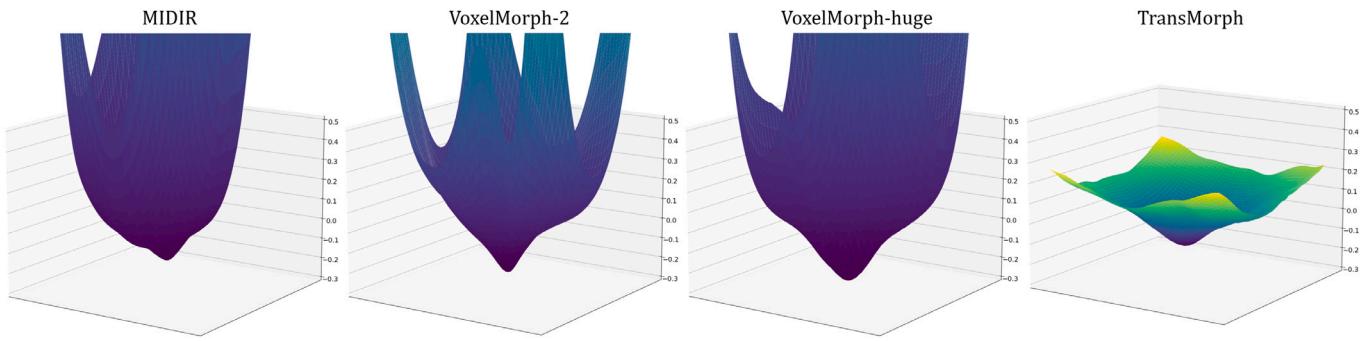


Fig. 16. The loss landscapes of MIDIR, VoxelMorph-2, VoxelMorph-huge, and TransMorph, where the loss function is composed of LNCC and diffusion regularizer. TransMorph yielded a much flatter landscape than those of ConvNet-based models.

hand, is not necessarily the optimal metric for expressing the expected error. In multi-modal registration instances like PET to CT or MRI to CT registration, MSE is anticipated to be high, given the vast difference in image appearance and voxel values across modalities. Thus, if MSE is employed to quantify the appearance uncertainty in these instances, the uncertainty values will be dominated by the squared bias (i.e., $(\hat{I}_f - I_f)^2$ in Eq. (B.3)), resulting in an ineffective uncertainty estimate. In these instances, the predicted variance may be a more appropriate choice for appearance uncertainty quantification.

Additional results for both appearance and transformation uncertainty estimations are shown in Fig. F.30 in Appendix A. Observably, the two uncertainty measures provide estimates that are substantially different, with appearance uncertainty values being high in locations with substantial appearance mismatches and transformation uncertainty values being high in regions with large deformations and generally constant intensity values.

6.3. Comparison of effective receptive fields

We demonstrate in this section that the *effective receptive fields* (ERFs) of Transformer-based models are larger than that of ConvNet-based models and span the whole spatial domain of an image. We used the definition of ERF introduced in Luo et al. (2016), which quantifies the amount of influence that each input voxel has on the output of a neural network. In the next paragraph, we briefly discuss the computation of ERF and recommend interested readers to the reference for further information.

Assume the voxels in the input image I_m and the output displacement field \mathbf{u} are indexed by (i, j, k) , with an image size of $160 \times 160 \times 160$ (i.e., the size of CT scans used in this work), the center voxel is located at $(80, 80, 80)$. ERF quantifies how much each $I_m(i, j, k)$ contributes to the center voxel of the displacement field, i.e. $\mathbf{u}(80, 80, 80)$. This is accomplished using the partial derivative $\partial\mathbf{u}(80, 80, 80)/\partial I_m(i, j, k)$, which indicates the relative relevance of $I_m(i, j, k)$ to $\mathbf{u}(80, 80, 80)$. To obtain this partial derivative, we set the error gradient to:

$$\frac{\partial \ell}{\partial \mathbf{u}(i, j, k)} = \begin{cases} 1, & \text{for } (i, j, k) = (80, 80, 80) \\ 0, & \text{otherwise} \end{cases} \quad (26)$$

where ℓ denotes an arbitrary loss function. Then this gradient is propagated downward from \mathbf{u} to the input I_m , where the resulting gradient of I_m represents the desired partial derivative $\partial\mathbf{u}(80, 80, 80)/\partial I_m(i, j, k)$. This partial derivative is independent of the input and loss function and is only a function of the network architecture and the index (i, j, k) , which adequately describes the distribution of the effective receptive field.

A comparison of the ERFs of VoxelMorph and TransMorph is shown in Fig. 15. Note that the other ConvNet-based models were omitted because they adopted a similar network architecture as VoxelMorph (e.g., CycleMorph and MIDIR). Due to the locality of convolution operations, VoxelMorph's ERF at each stage (top row in Fig. 15) was highly localized, particularly in the encoding stages (i.e., 1/4, 1/8, and 1/16 resolution). Even at the end of the network, the theoretical receptive field of VoxelMorph encompassed the entire image; yet, its ERF emphasized only a small portion of the image. In contrast, the ERFs of the proposed TransMorph were substantially larger than those of VoxelMorph at each stage, and the ERFs in the decoding stage covered the entire image (bottom row in Fig. 15). The ERFs reveal that ConvNet-based architectures can only perceive a portion of the input image, particularly during the encoding stages, indicating that they cannot explicitly comprehend the spatial relationships between distant voxels. For tasks that require large deformations, ConvNets may fall short of establishing accurate voxel correspondences between the moving and fixed images, which is essential for image registration. On the other hand, TransMorph adopts substantially large kernels at the encoding stages leading to substantially large ERFs throughout the network thanks to the self-attention mechanism of the Transformer.

6.4. Comparison of displacement magnitudes

As demonstrated in Section 6.3, TransMorph had substantially larger effective receptive fields than VoxelMorph, which might be beneficial for capturing semantic information that is necessary for coping with large deformations (Ha et al., 2020). In this section, we provide more evidence that Transformer-based models are more capable of producing larger deformations. We used 115 test volumes from the IXI dataset to generate histograms of displacement magnitudes in millimeters. Fig. 17 shows histograms of the displacement magnitudes for the various methods. The models that produced dense displacement fields are shown for fair comparisons. Note that VoxelMorph and CycleMorph are ConvNet-based models, whereas the other models are Transformer-based. All models were trained under the identical setting (e.g., loss functions, number of epochs, optimizers, etc.), where the only variable was the network architecture. As indicated by the histograms, all Transformer-based models had much more larger displacements than ConvNet-based models. The displacement distributions of ConvNet-based models had a mode near 0 and had more smaller displacements. We additionally showed the histograms of VoxelMorph-huge and TransMorph-small, the former of which had 63.25M parameters and the latter of which had 11.76M parameters. Despite having around 6× more parameters, VoxelMorph-huge still exhibited smaller displacements than TransMorph-small. This further indicates that the larger displacements produced by TransMorph were not a consequence of an increase in the number of parameters but rather the network architecture. Given the above-demonstrated improved registration performance of the Transformer-based models,

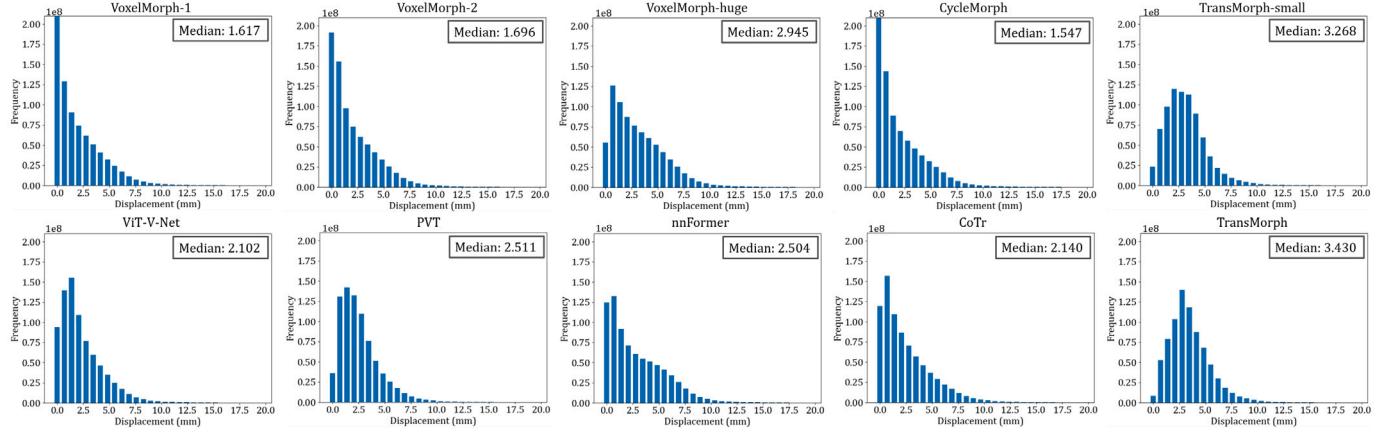


Fig. 17. Histograms of the displacement magnitudes in millimeters. These histograms were generated using 115 test volumes from the IXI dataset. The displacement magnitude is computed as $\sqrt{d_x^2 + d_y^2 + d_z^2}$, where $d_{\{x,y,z\}}$ denotes the displacement in x , y , and z directions. The median displacement magnitude is shown in the upper right corner of each plot. To provide fair comparisons, only models that produce dense displacements are shown here. VoxelMorph and CycleMorph are ConvNet-based models, whereas the other models are Transformer-based.

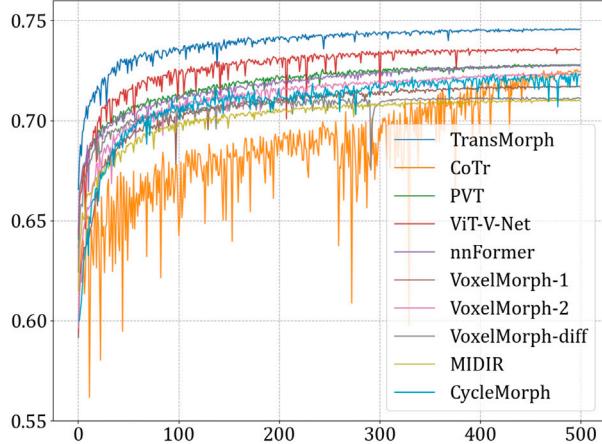


Fig. 18. Validation Dice scores for inter-patient brain MRI registration during training. The validation dataset comprises 104 image pairings that were not included in the training or testing set.

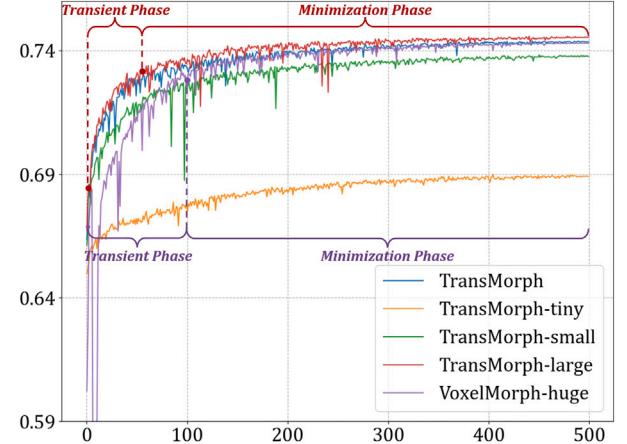
these histograms indicate that in cases where larger displacements are required, the Transformer-based models will likely provide better registration.

6.5. Comparison of loss landscapes

In this section, the loss landscapes of TransMorph and ConvNet-based models are compared. We adopted the loss landscape visualization method described in Li et al. (2018), Goodfellow et al. (2014), Im et al. (2016), in which a set of pre-trained model parameters (denoted as θ) are perturbed in two random directions (denoted as δ and η) with step sizes of α and β to acquire loss values at different locations. The loss landscape was plotted based on the function of the form:

$$f(\alpha, \beta) = \mathcal{L}(\theta + \alpha\delta + \beta\eta), \quad (27)$$

where \mathcal{L} denotes the loss function made up of LNCC and diffusion regularizer. We averaged the loss landscapes of ten samples from the validation set of the atlas-to-patient registration task to obtain the final 3D contour plot for each model. For comparison between ConvNet-based models and TransMorph, the loss landscapes of VoxelMorph, MIDIR, and TransMorph were created as shown in Fig. 16. TransMorph produced a substantially flatter loss landscape than those of the ConvNet-based models. This observation is consistent with the findings given in Park and Kim (2022), which suggest that Transformers tend



to promote flatter loss landscapes. Many studies have demonstrated that a flatter landscape results in improved performance and better generalizability (Park and Kim, 2022; Keskar et al., 2016; Santurkar et al., 2018; Foret et al., 2020; Li et al., 2018). The flatter landscape of TransMorph further demonstrates the advantages of Transformer-based models for image registration.

6.6. Convergence and speed

The left panel of Fig. 18 shows the validation dice scores of the learning-based methods during training. In comparison to other methods, the proposed TransMorph achieved > 0.7 in Dice within the first 20 epochs, showing that it learned the spatial correspondence between image pairs quicker than the competing models. Notably, TransMorph consistently outperformed the other Transformer-based models while having a comparable number of parameters and computational complexity. This implied Swin Transformer architecture was more effective than other Transformers, resulting in a performance improvement for TransMorph. On average, Transformer-based models provided better validation scores than ConvNet-based models, with the exception of CoTr, whose validation results were volatile during training (as seen from the orange curve in Fig. 18). The performance of CoTr may be limited by its architecture design, which substitutes a Transformer for the skip connections and bottleneck of a U-shaped CovNet. As a result, it lacks the direct flow of features learned during

Table 7

Average training and inference time for methods used in this work. Note that SyN, NiftyReg, and deedsBCV were applied using CPUs, while LDDMM and the learning-based methods were implemented on GPU. Inference time was averaged based on 40 repeated runs.

| Model | Training (min/epoch) | Inference (sec/image) |
|------------------|----------------------|-----------------------|
| SyN | – | 192.140 |
| NiftyReg | – | 30.723 |
| LDDMM | – | 66.829 |
| deedsBCV | – | 31.857 |
| VoxelMorph-1 | 8.75 | 0.380 |
| VoxelMorph-2 | 9.40 | 0.430 |
| VoxelMorph-diff | 4.20 | 0.049 |
| VoxelMorph-huge | 28.50 | 1.107 |
| CycleMorph | 41.90 | 0.281 |
| MIDIR | 4.05 | 1.627 |
| ViT-V-Net | 9.20 | 0.197 |
| PVT | 13.80 | 0.209 |
| CoTr | 17.10 | 0.372 |
| nnFormer | 6.35 | 0.105 |
| TransMorph-Bayes | 22.60 | 7.739 |
| TransMorph-diff | 7.35 | 0.099 |
| TransMorph-bspl | 10.50 | 1.739 |
| TransMorph | 14.40 | 0.329 |

the encoding stage to the layers creating the registration, making it difficult to converge. The right panel of Fig. 18 shows the training curves of the TransMorph variants and the customized VoxelMorph-huge. As described in Im et al. (2016), Sutskever et al. (2013), Darken and Moody (1991), the training curve of a deep learning model consists of two phases: a “transient” phase followed by a “minimization” phase, where the former identifies the neighborhood of local minima and the latter seeks the local minima inside that neighborhood. As seen in the figure, TransMorph variants had shorter “transient” phases than that of VoxelMorph-huge, indicating that they identified the local minima neighborhood more quickly. A fast convergent algorithm is often preferred since it not only saves time but also computing resources and costs. There have been many efforts to accelerate the convergence rate of deep learning models (Darken and Moody, 1991; Looney, 1996; Zeiler et al., 2013; Smith and Topin, 2019). TransMorph tends to accelerate convergence rate compared to ConvNet-based models, which promotes its potential of faster training using fewer epochs, saving time and reducing the carbon footprint.

Table 7 compares the training time in min per epoch (min/epoch) and inference time in seconds per image (sec/image) among the methods used in this paper. Note that SyN, NiftyReg, and deedsBCV packages are all CPU-based, while LDDMM and the deep-learning-based methods are all GPU-based. The speed was calculated using an input image size of $160 \times 192 \times 224$, which corresponds to the size of the brain MRI scans. The training time per epoch was computed based on 768 training image pairs. The most and second most time-consuming methods to train are two ConvNet-based methods, CycleMorph and the customized VoxelMorph-huge, which required approximately $(41.90 \text{ min} \times 500)/(60 \text{ min} \times 24 \text{ h}) \approx 15 \text{ days}$ and $(28.50 \text{ min} \times 500)/(60 \text{ min} \times 24 \text{ h}) \approx 10 \text{ days}$ for 500 epochs of training, respectively. CycleMorph was time-consuming because the cycle-consistent training virtually trains four networks simultaneously in a single epoch. Whereas the training of VoxelMorph-huge was slowed down by the extensive convolution operations. The proposed TransMorph has a moderate training speed, roughly 1.5× that of VoxelMorph-2 and 0.5× that of the customized VoxelMorph-huge. In terms of inference time, learning-based models undoubtedly operated orders of magnitudes faster than traditional registration methods. Note that TransMorph is about 3× faster than VoxelMorph-huge during inference. These findings are proportional to the calculated computational complexity as shown in the barplot on the left in Fig. 10. Among the learning-based models, TransMorph-Bayes required the highest inference time. However, the time required is due to the

sampling of $T = 25$ images for a single prediction and uncertainty estimation.

6.7. Limitations

There are some limitations to our work. First, rather than doing extensive grid searches for optimal hyperparameters for the baseline methods, the hyperparameters are either determined empirically or based on the values suggested in the original paper. Due to the time required to train some of the baseline methods and the limited memory available on the GPU, we were unable to afford the intensive grid search. Moreover, because this study introduced a generic network architecture for image registration, we concentrated on architectural comparison rather than on selecting optimal hyperparameters for loss functions or complex training methods. However, the proposed TransMorph architecture is readily adaptable using either the cycle-consistent training method used by CycleMorph (Kim et al., 2021) or the symmetric training method proposed in Mok and Chung (2020). Additionally, the proposed network may be used in conjunction with any registration loss function.

In the future, we will investigate alternative loss functions, such as mutual information, in an effort to expand the potential of the proposed method for multi-modal registration tasks.

7. Conclusion

In this paper, we introduced TransMorph, a novel model for unsupervised deformable image registration. TransMorph is built on Transformer, which is well-known for its capability to establish long-range spatial correspondence between image voxels, making TransMorph a strong candidate for image registration tasks.

Two variants of TransMorph are proposed, which provide topology-preserved deformations. Additionally, we introduced Bayesian deep learning to the Transformer encoder of TransMorph, enabling deformation uncertainty estimation without degrading registration performance.

We evaluated TransMorph on the task of inter-patient brain MR registration and a novel task of phantom-to-CT registration. The results revealed that TransMorph achieved superior registration accuracy than various traditional and learning-based methods, demonstrating its effectiveness for medical image registration.

CRediT authorship contribution statement

Junyu Chen: Conceptualization, Methodology, Software, Data curation, Investigation, Writing – original draft, Visualization. **Eric C. Frey:** Validation, Resources, Writing – review & editing, Supervision, Funding acquisition. **Yufan He:** Methodology, Validation, Investigation, Writing – review & editing. **William P. Segars:** Data curation. **Ye Li:** Validation. **Yong Du:** Validation, Resources, Data curation, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by grants from the National Institutes of Health, United States, U01-CA140204, R01EB031023, and U01EB031798. The views expressed in written conference materials or publications and by speakers and moderators do not necessarily reflect the official policies of the NIH; nor does mention by trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Appendix A. Affine network architecture

See Fig. A.19.

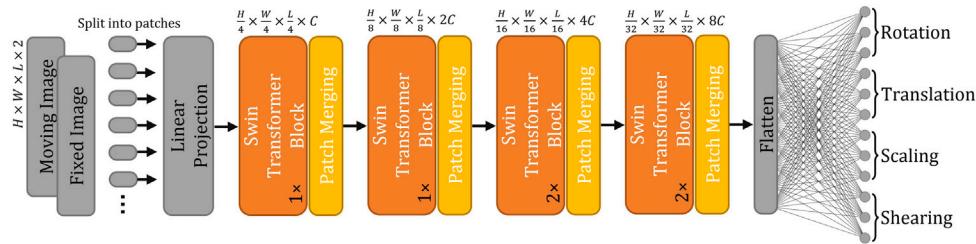


Fig. A.19. Visualization of the proposed Swin-Transformer-based affine network. This network outputs three rotation, three translation, three scaling, and three shearing parameters for rigid registration. The embedding dimension C in the network was set to 12.

Appendix B. Miscalibration in predictive variance

The expected model error (characterized by MSE) is defined as:

$$\text{err}(I_m \circ \phi) = \mathbb{E} [(I_m \circ \phi - I_f)^2] = \frac{1}{T} \sum_{t=1}^T (I_m \circ \phi_t - I_f)^2, \quad (\text{B.1})$$

where t represents the t^{th} sample from a total number of T samples. We denote $I_d = I_m \circ \phi$ for convenience, and it can be shown that:

$$\begin{aligned} \mathbb{E} [(I_d - I_f)^2] &= \mathbb{E} [(I_d - \mathbb{E}[I_d] + \mathbb{E}[I_d] - I_f)^2] \\ &= \mathbb{E} [(I_d - \mathbb{E}[I_d])^2] + (\mathbb{E}[I_d] - I_f)^2 + 2(\mathbb{E}[I_d] - I_f)\mathbb{E}[I_d \\ &\quad - \mathbb{E}[I_d]] \\ &= \mathbb{E} [(I_d - \mathbb{E}[I_d])^2] + (\mathbb{E}[I_d] - I_f)^2. \end{aligned} \quad (\text{B.2})$$

Therefore,

$$\begin{aligned} \text{err}(I_m \circ \phi) &= \frac{1}{T} \sum_{t=1}^T (I_m \circ \phi_t - I_f)^2 \\ &= \frac{1}{T} \sum_{t=1}^T \left(I_m \circ \phi_t - \frac{1}{T} \sum_{t=1}^T I_m \circ \phi_t \right)^2 + \left(\frac{1}{T} \sum_{t=1}^T I_m \circ \phi_t - I_f \right)^2 \\ &= \hat{\Sigma}_f^2 + (\hat{I}_f - I_f)^2, \end{aligned} \quad (\text{B.3})$$

where $\hat{I}_f - I_f$ is referred to as the bias between the predictive mean \hat{I}_f and the target image I_f . Due to the problem of overfitting the training set in supervised algorithms (e.g., deep learning) (Bishop, 2006), this bias may be less noticeable on training dataset but more noticeable on test images, which is a phenomenon referred to as the *bias-variance tradeoff* (Friedman, 2017). As a consequence, the predictive variance $\hat{\Sigma}_f^2$ is systematically smaller than the expected error $\text{err}(I_m \circ \phi)$, resulting in miscalibrated uncertainty estimations.

Appendix C. Additional results for inter-patient brain MRI registration

See Figs. C.20–C.22.

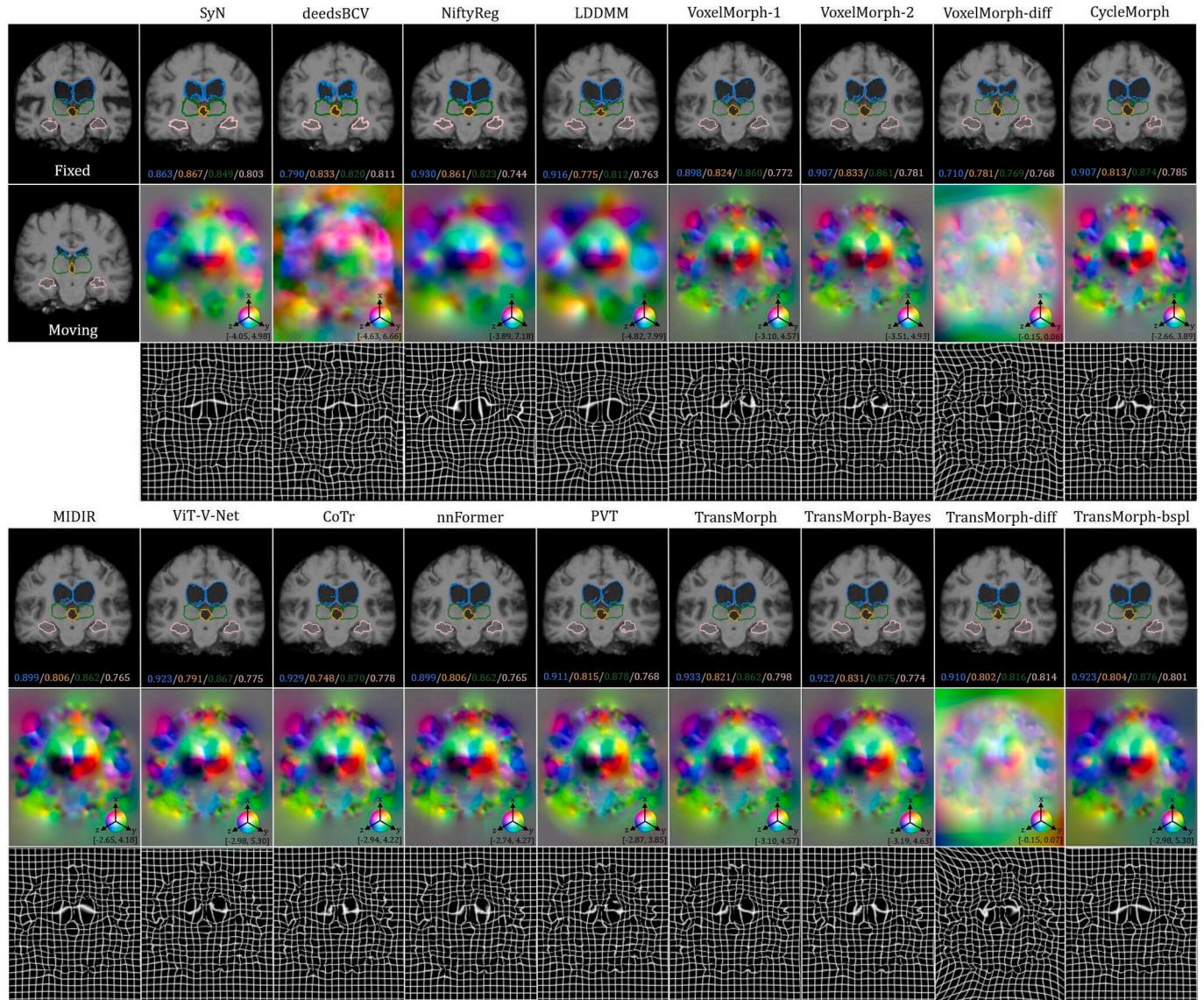


Fig. C.20. Additional qualitative comparison of various registration methods on the inter-patient brain MR registration task. The first row shows the deformed moving images, the second row shows the deformation fields, and the last row shows the deformed grids. The spatial dimension x , y , and z in the displacement field is mapped to each of the RGB color channels, respectively. The $[p, q]$ in color bars denotes the magnitude range of the fields.

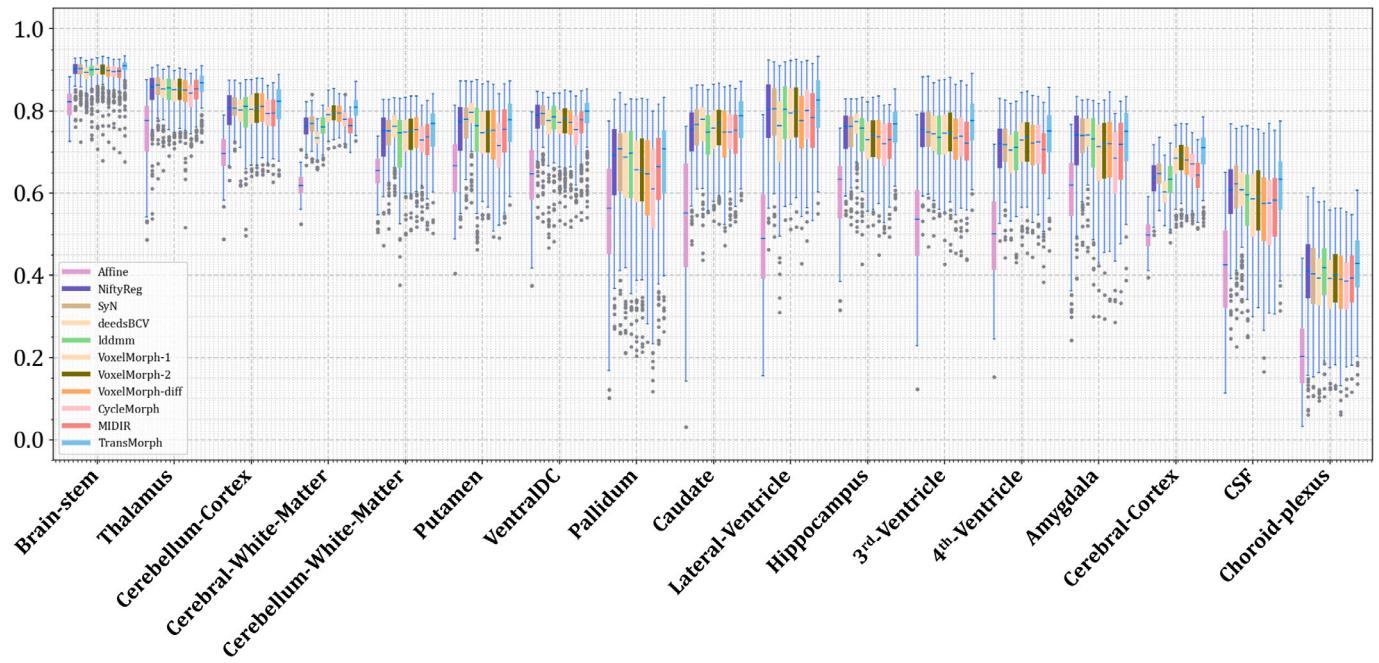


Fig. C.21. Quantitative comparison of the various registration methods on the inter-patient brain MR registration task. Boxplots showing Dice scores for different brain MR substructures using the proposed TransMorph and existing image registration methods.

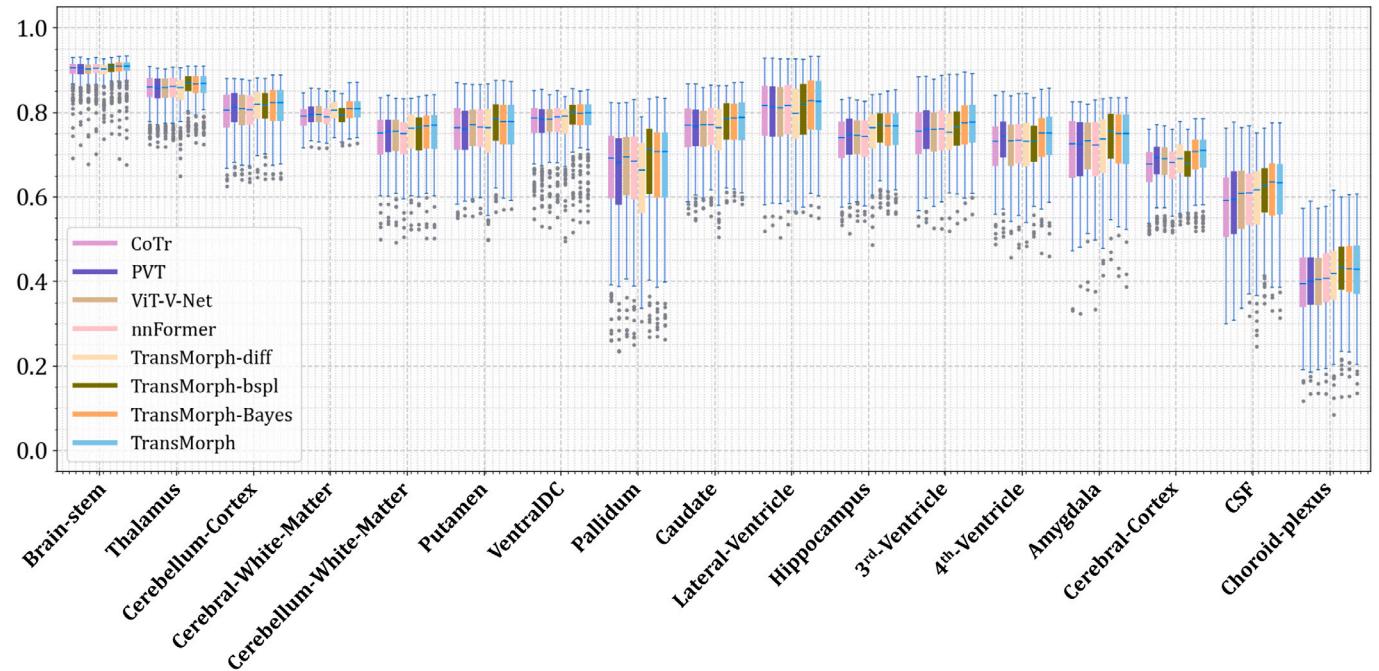


Fig. C.22. Quantitative comparison of the Transformer-based models on the inter-patient brain MR registration task. Boxplots showing Dice scores for different brain MR substructures using the proposed TransMorph, the variants of TransMorph, and other Transformer architectures.

Appendix D. Additional results for atlas-to-patient brain MRI registration

See Figs. D.23–D.25.

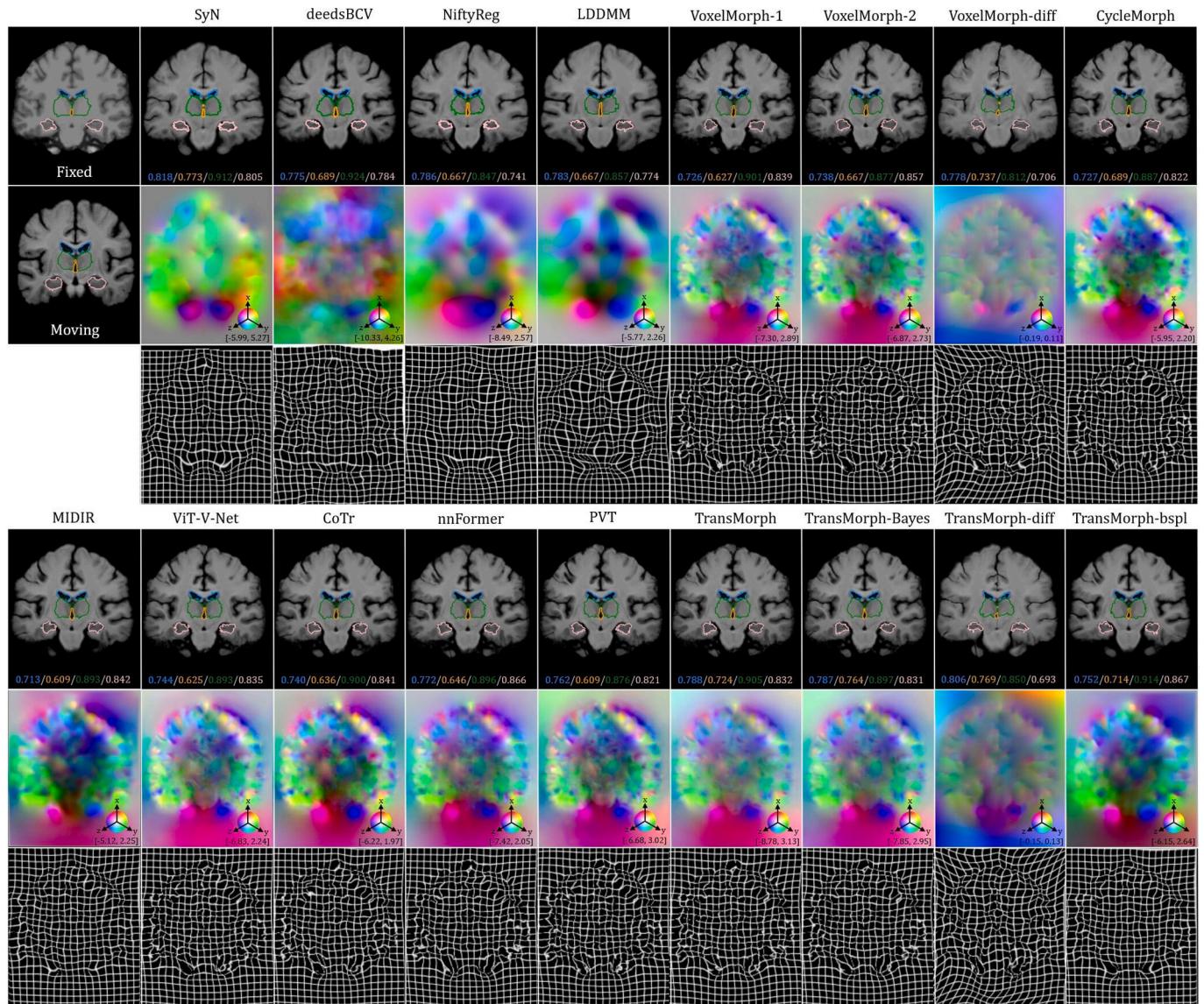


Fig. D.23. Additional qualitative comparison of various registration methods on the atlas-to-patient brain MR registration task. The first row shows the deformed moving images, the second row shows the deformation fields, and the last row shows the deformed grids. The spatial dimension x , y , and z in the displacement field is mapped to each of the RGB color channels, respectively. The $[p, q]$ in color bars denotes the magnitude range of the fields.

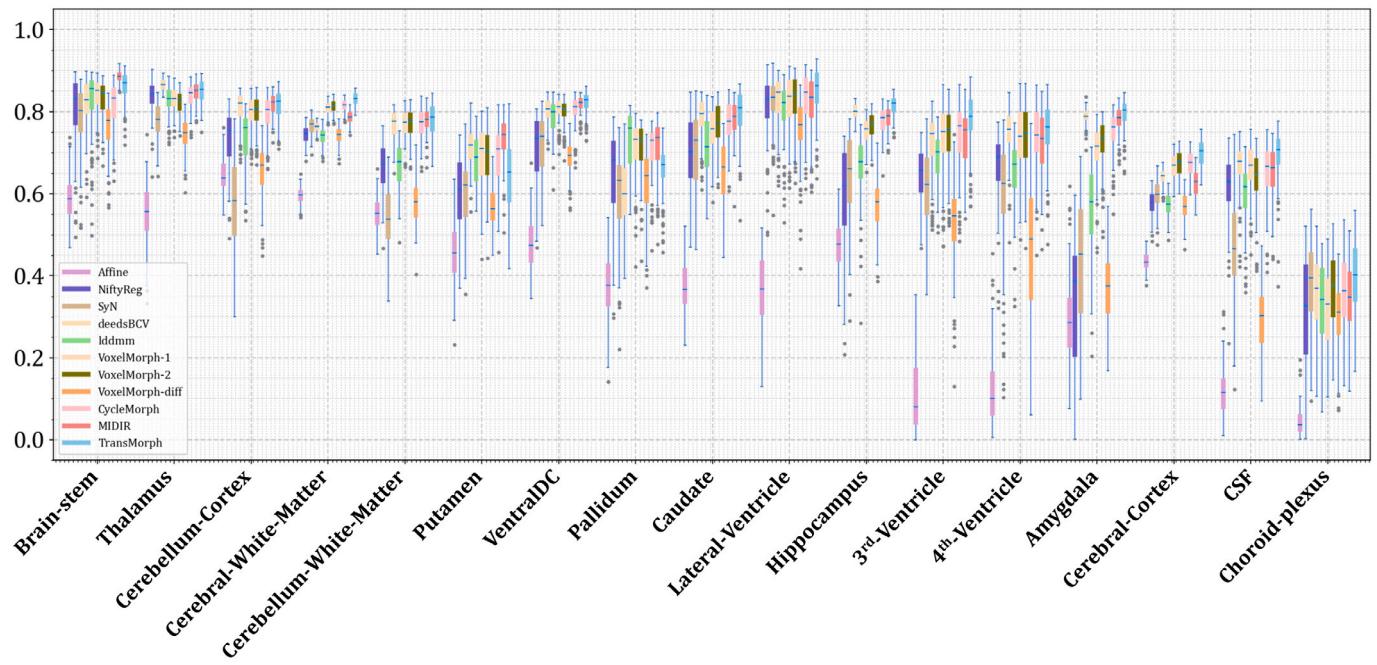


Fig. D.24. Quantitative comparison of the various registration methods on the atlas-to-patient brain MR registration task. Boxplots showing Dice scores for different brain MR substructures using the proposed TransMorph and existing image registration methods.

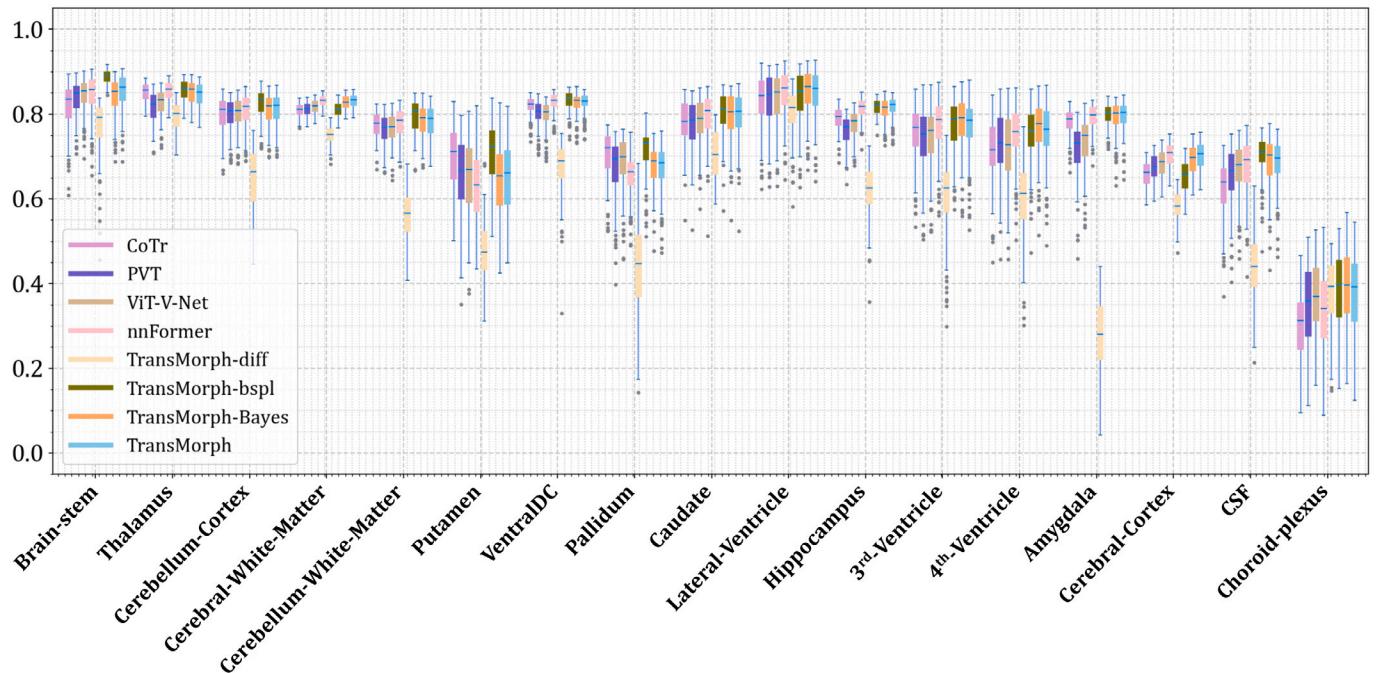


Fig. D.25. Quantitative comparison of the Transformer-based models on the atlas-to-patient brain MR registration task. Boxplots showing Dice scores for different brain MR substructures using the proposed TransMorph, the variants of TransMorph, and other Transformer architectures.

Appendix E. Additional results for XCAT-to-CT registration

See Figs. E.26–E.29.

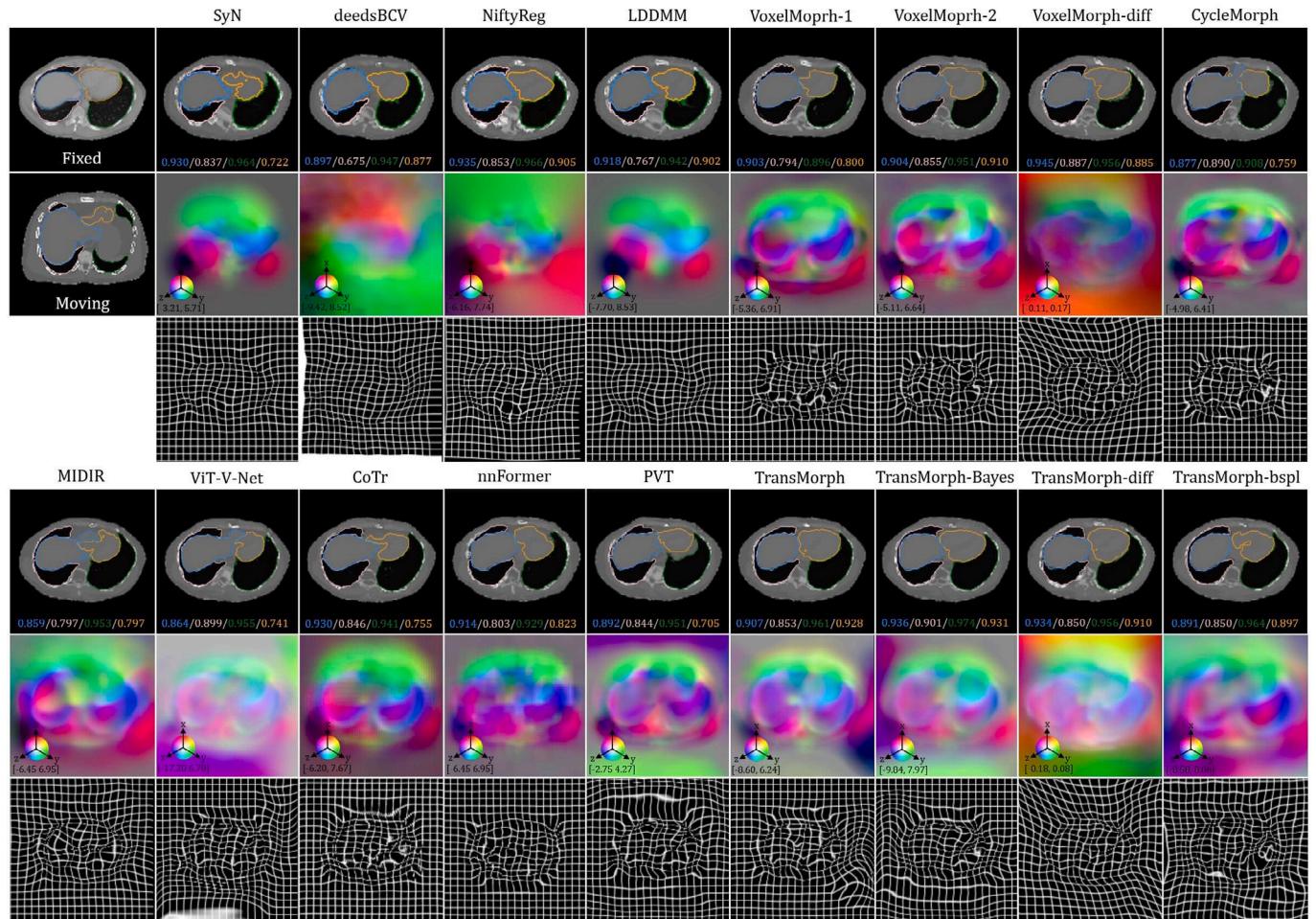


Fig. E.26. Additional qualitative comparison of various registration methods on the XCAT-to-CT registration task. The first row shows the deformed moving images, the second row shows the deformation fields, and the last row shows the deformed grids. The spatial dimension x , y , and z in the displacement field is mapped to each of the RGB color channels, respectively. The $[p, q]$ in color bars denotes the magnitude range of the fields.

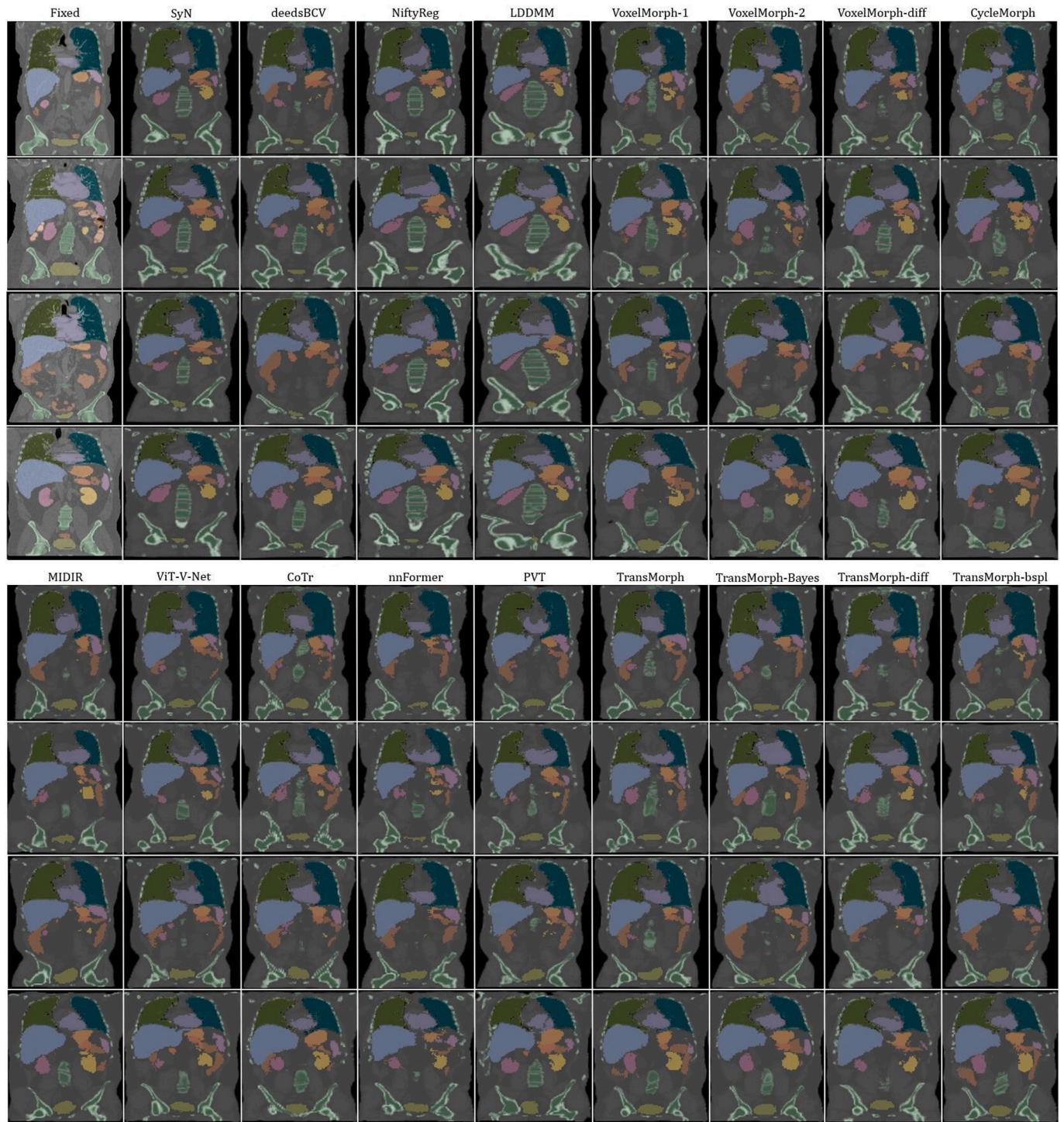


Fig. E.27. Additional coronal slices of the deformed XCAT phantom generated by various registration methods.

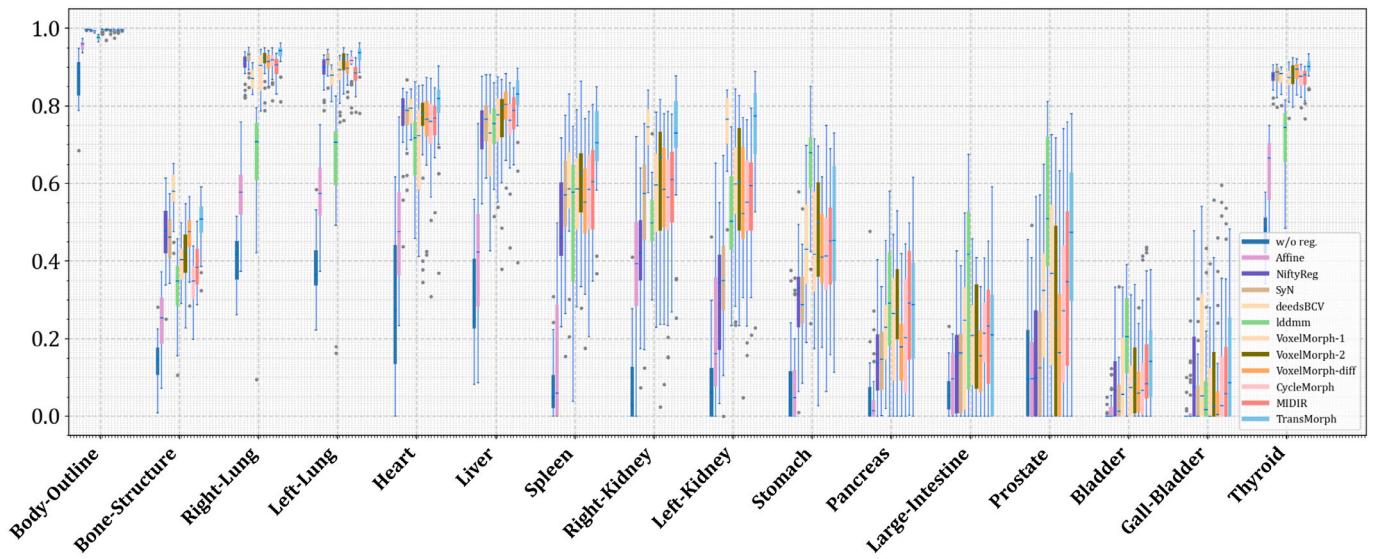


Fig. E.28. Quantitative comparison of various registration methods on the XCAT-to-CT registration task. Boxplots showing Dice scores for different organs in CT obtained using the proposed TransMorph and existing image registration methods.

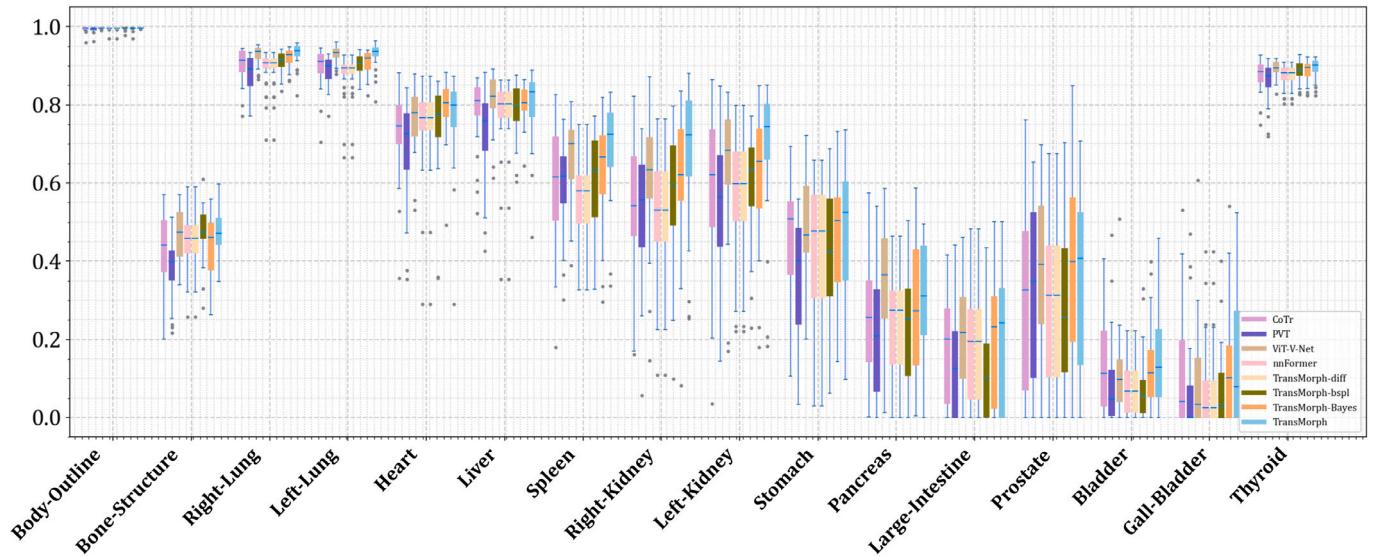


Fig. E.29. Quantitative comparison of the Transformer-based models on the XCAT-to-CT registration task. Boxplots showing Dice scores for different organs in CT obtained using the proposed TransMorph, the variants of TransMorph, and other Transformer architectures.

Appendix F. Additional qualitative results for uncertainty quantification

See Fig. F.30.

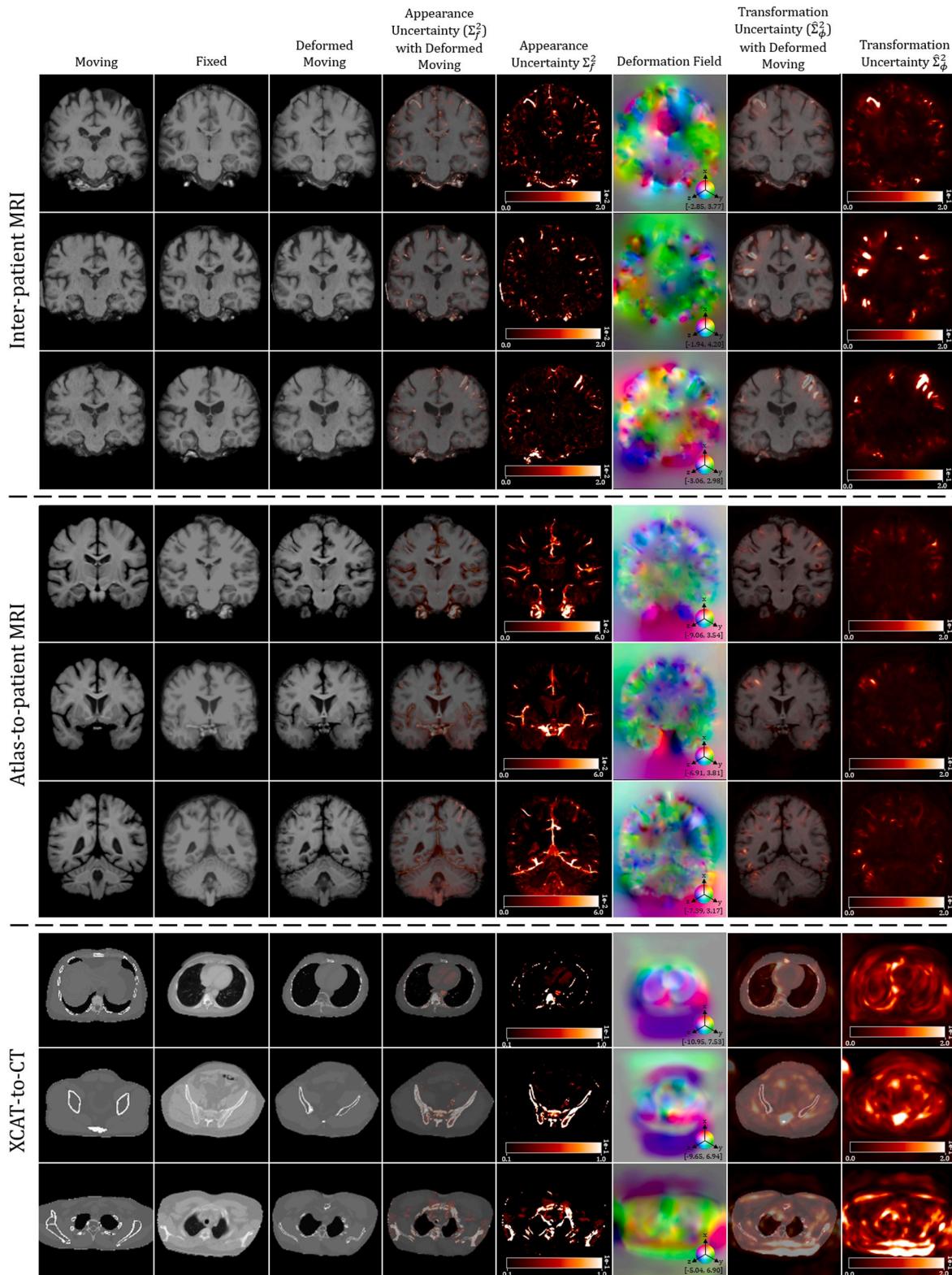


Fig. F.30. Qualitative results and registration uncertainty estimate with TransMorph-Bayes. The fourth and the fifth columns exhibit the appearance uncertainties estimated using the proposed uncertainty estimation scheme (i.e., Σ_f^2). The last and the second to last columns shows the transformation uncertainties, i.e., $\hat{\Sigma}_{\phi}^2$, where the uncertainty maps were taken as square root of the sum of the variances of the deformation in x , y , and z direction. The spatial dimension x , y , and z in the displacement field is mapped to each of the RGB color channels, respectively. The $[p, q]$ in color bars denotes the magnitude range of the fields.

Appendix G. Visualization of feature maps in skip connections

See Fig. G.31.

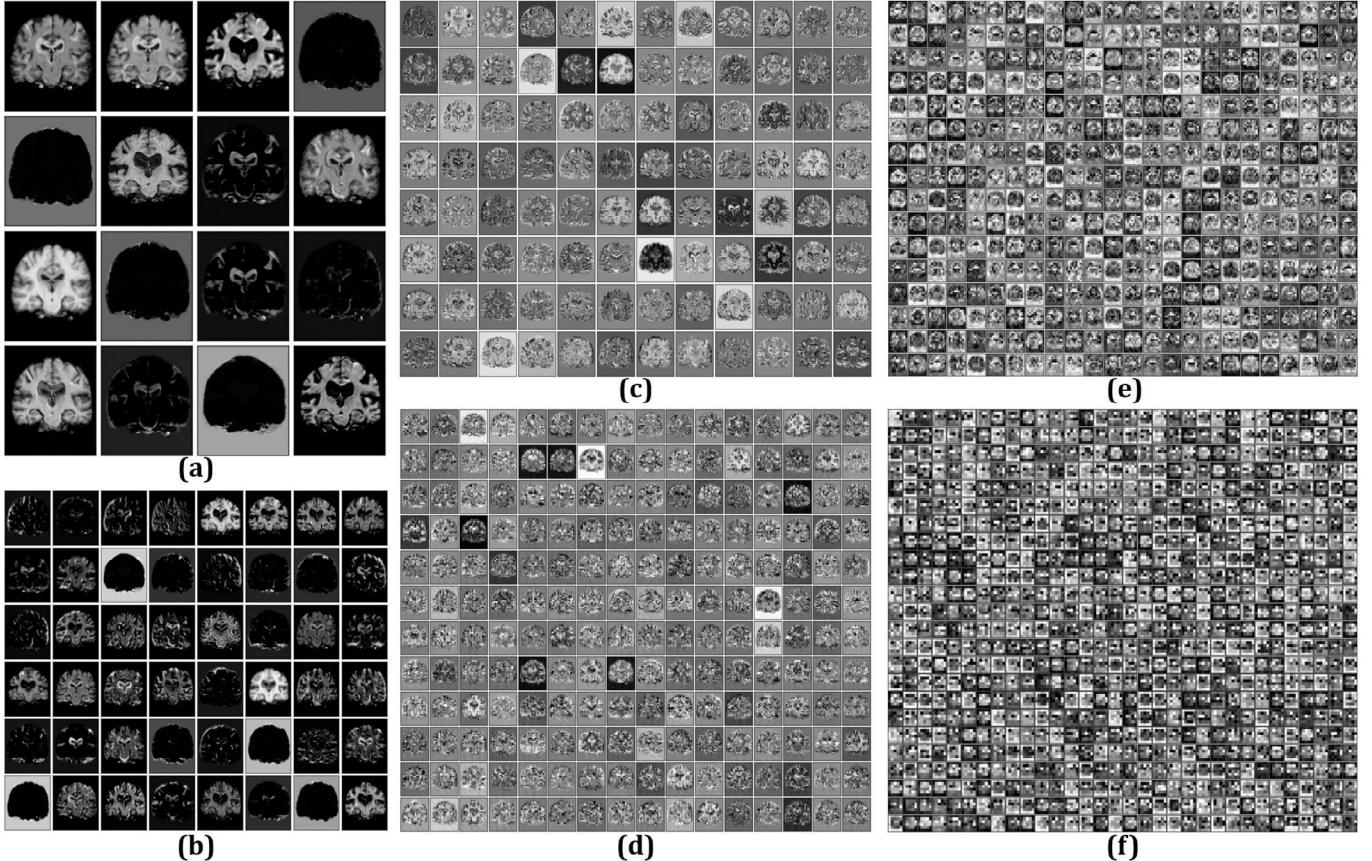


Fig. G.31. Feature maps in TransMorph’s skip connections. (a) and (b) exhibit, respectively, the feature maps in the first and second skip connections from the convolutional layers in the encoder (i.e., the green arrows in Fig. 1); (c)–(f) exhibit the feature maps in the skip connections from the Transformer blocks (i.e., the orange arrows in Fig. 1).

Appendix H. Probabilistic diffeomorphic registration

As shown in Section 3.3, we introduced a variational inference framework to the proposed TransMorph (which we denote as TransMorph-diff). A prior distribution

$$p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{0}, \Sigma_{\mathbf{u}}) \quad (\text{H.1})$$

was placed over the dense displacement field \mathbf{u} , where $\mathbf{0}$ and $\Sigma_{\mathbf{u}}$ are the mean and covariance of the multivariate Gaussian distribution. We followed Dalca et al. (2019) and defined $\Sigma_{\mathbf{u}}^{-1} = A_{\mathbf{u}} = \lambda \mathbf{L}$, where $A_{\mathbf{u}}$ denotes the precision matrix, λ controls the scale of \mathbf{u} , $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the Laplacian matrix of a neighborhood graph formed on the voxel grid, \mathbf{D} is the graph degree matrix, and \mathbf{A} is a voxel neighborhood adjacency matrix. The probability $p(I_f | I_m)$ can be computed using the law of total probability:

$$p(I_f | I_m) = \int_{\mathbf{u}} p(I_f | \mathbf{u}, I_m) p(\mathbf{u}) d\mathbf{u}. \quad (\text{H.2})$$

The likelihood $p(I_f | \mathbf{u}, I_m)$ was also assumed to be Gaussian

$$p(I_f | \mathbf{u}, I_m) = \mathcal{N}(I_f; I_m \circ \phi_{\mathbf{u}}, \sigma_1^2 \mathbb{I}), \quad (\text{H.3})$$

where σ_1^2 captures the variance of the image noise, and $\phi_{\mathbf{u}}$ is the group exponential of the time-stationary velocity field \mathbf{u} , i.e. $\phi = \exp(\mathbf{u})$, and was computed using a scaling-and-squaring approach (Section 2.1.2).

Our goal is to estimate the posterior probability $p(\mathbf{u}|I_f, I_m)$. Due to the intractable nature of the integral over \mathbf{u} in Eq. (H.2), $p(I_f | I_m)$ is

usually calculated using just the \mathbf{u} ’s that are most likely to have generated I_f (Krebs et al., 2019). Since computing the posterior $p(\mathbf{u}|I_f, I_m)$ analytically is also intractable, we instead assumed a variational posterior $q_{\psi}(\mathbf{u}|I_f, I_m)$ learned by the network with parameters ψ . The Kullback–Leibler divergence (KL) was used to relate the variational posterior to the actual posterior, which results in the evidence lower limit (ELBO) (Kingma and Welling, 2013):

$$\log p(I_f | I_m) - \text{KL} [q_{\psi}(\mathbf{u}|I_f, I_m) || p(\mathbf{u}|I_f, I_m)] = \mathbb{E}_{\mathbf{u} \sim q_{\psi}} [\log p(I_f | \mathbf{u}, I_m)] - \text{KL} [q_{\psi}(\mathbf{u}|I_f, I_m) || p(\mathbf{u})], \quad (\text{H.4})$$

where the KL-divergence on the left hand side vanishes if the variational posterior is identical to the actual posterior. Therefore, maximizing $\log p(I_f | I_m)$ is equivalent to minimizing the negative of ELBO on the right hand side of Eq. (H.4). Since the prior distribution $p(\mathbf{u})$ was assumed to be a multivariate Gaussian, the variational posterior is likewise a multivariate Gaussian, defined as:

$$q_{\psi}(\mathbf{u}|I_f, I_m) = \mathcal{N}(\mathbf{u}; \boldsymbol{\mu}_{\psi}(\mathbf{u}|I_f, I_m), \boldsymbol{\Sigma}_{\psi}(\mathbf{u}|I_f, I_m)), \quad (\text{H.5})$$

where $\boldsymbol{\mu}_{\psi}$ and $\boldsymbol{\Sigma}_{\psi}$ are the voxel-wise mean and variance generated by the network with parameters ψ . In each forward pass, the dense displacement field \mathbf{u} is sampled using reparameterization $\mathbf{u} = \boldsymbol{\mu}_{\psi} + \boldsymbol{\Sigma}_{\psi} \odot \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The variational parameters $\boldsymbol{\mu}_{\psi}$ and $\boldsymbol{\Sigma}_{\psi}$ are learned

by minimizing the loss (Dalca et al., 2019):

$$\begin{aligned} \mathcal{L}_{prob.}(I_f, I_m, \phi_u; \psi) &= -\mathbb{E}_{\mathbf{u} \sim q_\psi} [\log p(I_f | \mathbf{u}, I_m)] + \text{KL}[q_\psi(\mathbf{u} | I_f, I_m) || p(\mathbf{u})] \\ &= \frac{1}{2\sigma^2} \|I_f - I_m \circ \phi_u\|^2 + \frac{1}{2} \left[\text{tr}(\lambda \mathbf{D} \Sigma_\psi - \log \Sigma_\psi) + \mu_\psi^\top \Lambda_u \mu_\psi \right], \end{aligned} \quad (\text{H.6})$$

where $\mu_\psi^\top \Lambda_u \mu_\psi$ can be thought of as a diffusion regularization (Eq. (14)) placed over the mean displacement field μ_ψ , that is $\mu_\psi^\top \Lambda_u \mu_\psi = \frac{\lambda}{2} \sum_{\mathbf{p}} \sum_{i \in N_{(\mathbf{p})}} (\mu(\mathbf{p}) - \mu(i))^2$, where $N_{(\mathbf{p})}$ represents the neighboring voxels of the \mathbf{p}^{th} voxel.

As discussed in Section 3.2.2, when the auxiliary segmentation information is available (i.e., the label maps of I_f and I_m , denoted as s_f and s_m), Dice loss can be used for training the network to further enhance registration performance. Dice loss, however, does not preserve a Gaussian approximation of the deformation fields. Instead, we follow Dalca et al. (2019) and replace the KL divergence in Eq. (H.4) with:

$$\text{KL}[q_\psi(\mathbf{u} | I_f, I_m) || p(\mathbf{u} | I_f, s_f; I_m, s_m)], \quad (\text{H.7})$$

which yields a loss function of the form:

$$\begin{aligned} \mathcal{L}_{prob. w/ aux.}(I_f, s_f, I_m, s_m, \phi_u; \psi) &= \frac{1}{2\sigma^2} \|I_f - I_m \circ \phi_u\|^2 + \frac{1}{2\sigma_s^2} \|s_f - s_m \circ \phi_u\|^2 \\ &\quad + \frac{1}{2} \left[\text{tr}(\lambda \mathbf{D} \Sigma_\psi - \log \Sigma_\psi) + \mu_\psi^\top \Lambda_u \mu_\psi \right]. \end{aligned} \quad (\text{H.8})$$

In Dalca et al. (2019), s_f and s_m represent *anatomical surfaces* obtained from label maps. In contrast, we directly used the *label maps* as s_f and s_m in this work. They were image volumes with multiple channels, each channel contained a binary mask defining the segmentation of a certain structure/organ.

Appendix I. B-splines diffeomorphic registration

As demonstrated in Section 3.3, we incorporated a cubic B-spline model (Qiu et al., 2021) into TransMorph (which we denote as TransMorph-bspl). This network produces a lattice of low-dimensional control points instead of producing a dense displacement field at the original resolution, which might be computationally costly. As shown in Fig. 6, we denote the displacements of the B-spline control points generated by the network as \mathbf{u}_B and the spacing between the control points as δ . Then, a weighted combination of cubic B-spline basis functions (i.e., β_d) (Rueckert et al., 1999) is used to generate the dense displacement field (i.e., the B-spline tensor product in Fig. 6):

$$\hat{\mathbf{u}}(\mathbf{p}) = \sum_{\mathbf{c} \in C} \mathbf{u}_B(\mathbf{c}) \prod_{d \in \{x,y,z\}} \beta_d(\mathbf{p}_d - k(\mathbf{c}_d)), \quad (\text{I.1})$$

where \mathbf{c} is the index of the control points on the lattice C , and k denotes the coordinates of the control points $\mathbf{u}_B(\mathbf{c})$ in image space. Then the final time-stationary displacement \mathbf{u} is obtained using the same *scaling-and-squaring* approach described in Section 2.1.2.

References

- Alom, M.Z., Hasan, M., Yakopcic, C., Taha, T.M., Asari, V.K., 2018. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. arXiv preprint arXiv:1802.06955.
- Armstrong, R.A., 2014. When to use the B onferroni correction. Ophthalmic Physiol. Opt. 34 (5), 502–508.
- Arsigny, V., Commowick, O., Pennec, X., Ayache, N., 2006. A log-euclidean framework for statistics on diffeomorphisms. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 924–931.
- Ashburner, J., 2007. A fast diffeomorphic image registration algorithm. Neuroimage 38 (1), 95–113.
- Atanov, A., Ashukha, A., Molchanov, D., Neklyudov, K., Vetrov, D., 2018. Uncertainty estimation via stochastic batch normalization. arXiv preprint arXiv:1802.04893.
- Avants, B.B., Epstein, C.L., Grossman, M., Gee, J.C., 2008. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. Med. Image Anal. 12 (1), 26–41.
- Balakrishnan, G., Zhao, A., Sabuncu, M.R., Guttag, J., Dalca, A.V., 2018. An unsupervised learning model for deformable medical image registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9252–9260.
- Balakrishnan, G., Zhao, A., Sabuncu, M.R., Guttag, J., Dalca, A.V., 2019. VoxelMorph: a learning framework for deformable medical image registration. IEEE Trans. Med. Imaging 38 (8), 1788–1800.
- Baumgartner, C.F., Tezcan, K.C., Chaitanya, K., Hötker, A.M., Muehlematter, U.J., Schawkat, K., Becker, A.S., Donati, O., Konukoglu, E., 2019. Phiseg: Capturing uncertainty in medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 119–127.
- Beg, M.F., Miller, M.I., Trouvé, A., Younes, L., 2005. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. Int. J. Comput. Vis. 61 (2), 139–157.
- Bishop, C.M., 2006. Pattern recognition. Mach. Learn. 128 (9).
- Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D., 2015. Weight uncertainty in neural network. In: International Conference on Machine Learning. PMLR, pp. 1613–1622.
- Cao, X., Yang, J., Wang, L., Xue, Z., Wang, Q., Shen, D., 2018. Deep learning based inter-modality image registration supervised by intra-modality similarity. In: International Workshop on Machine Learning in Medical Imaging. Springer, pp. 55–63.
- Chen, J., Frey, E., Du, Y., 2022. Unsupervised learning of diffeomorphic image registration via TransMorph. In: 10th Internatioal Workshop on Biomedical Image Registration. URL https://openreview.net/forum?id=uwl0_2xnTO.
- Chen, J., He, Y., Frey, E.C., Li, Y., Du, Y., 2021a. ViT-V-Net: Vision transformer for unsupervised volumetric medical image registration. arXiv preprint arXiv:2104.06468.
- Chen, J., Jha, A.K., Frey, E.C., 2019. Incorporating CT prior information in the robust fuzzy C-means algorithm for QSPECT image segmentation. In: Medical Imaging 2019: Image Processing, vol. 10949. International Society for Optics and Photonics, p. 109491W.
- Chen, J., Li, Y., Du, Y., Frey, E.C., 2020. Generating anthropomorphic phantoms using fully unsupervised deformable image registration with convolutional neural networks. Med. Phys.
- Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y., 2021b. Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306.
- Chetty, I.J., Rosu-Bubulac, M., 2019. Deformable registration for dose accumulation. Sem. Radiat. Oncol. 29 (3), 198–208.
- Christoffersen, C.P., Hansen, D., Poulsen, P., Sorensen, T.S., 2013. Registration-based reconstruction of four-dimensional cone beam computed tomography. IEEE Trans. Med. Imaging 32 (11), 2064–2077. <http://dx.doi.org/10.1109/TMI.2013.2272882>.
- Cui, K., Fu, P., Li, Y., Lin, Y., 2021. Bayesian fully convolutional networks for brain image registration. J. Healthcare Eng. 2021.
- Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., Zhang, L., 2021. Dynamic head: Unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7373–7382.
- Dalca, A.V., Balakrishnan, G., Guttag, J., Sabuncu, M.R., 2019. Unsupervised learning of probabilistic diffeomorphic registration for images and surfaces. Med. Image Anal. 57, 226–236.
- Darken, C., Moody, J., 1991. Towards faster stochastic gradient search. Adv. Neural Inf. Process. Syst. 4.
- de Vos, B.D., Berendsen, F.F., Viergever, M.A., Sokooti, H., Staring, M., Isgum, I., 2019. A deep learning framework for unsupervised affine and deformable image registration. Med. Image Anal. 52, 128–143.
- de Vos, B.D., Berendsen, F.F., Viergever, M.A., Staring, M., Isgum, I., 2017. End-to-end unsupervised deformable image registration with a convolutional neural network. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp. 204–212.
- Devalla, S.K., Renukanand, P.K., Sreedhar, B.K., Subramanian, G., Zhang, L., Perera, S., Mari, J.-M., Chin, K.S., Tun, T.A., Strouthidis, N.G., et al., 2018. DRUNET: a dilated-residual U-net deep learning network to segment optic nerve head tissues in optical coherence tomography images. Biomed. Opt. Express 9 (7), 3244–3265.
- DeVries, T., Taylor, G.W., 2018. Leveraging uncertainty estimates for predicting segmentation quality. arXiv preprint arXiv:1807.00502.
- Dice, L.R., 1945. Measures of the amount of ecological association between species. Ecology 26 (3), 297–302.
- Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B., 2021. Cswin transformer: A general vision transformer backbone with cross-shaped windows. arXiv preprint arXiv:2107.00652.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
- Fischl, B., 2012. Freesurfer. Neuroimage 62 (2), 774–781.
- Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B., 2020. Sharpness-aware minimization for efficiently improving generalization. arXiv preprint arXiv:2010.01412.
- Friedman, J.H., 2017. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer open.

- Fu, W., Sharma, S., Abadi, E., Iliopoulos, A.-S., Wang, Q., Sun, X., Lo, J.Y.-C., Segars, W.P., Samei, E., 2021. Iphantom: a framework for automated creation of individualized computational phantoms and its application to CT organ dosimetry. *IEEE J. Biomed. Health Inf.*
- Gal, Y., Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning. PMLR, pp. 1050–1059.
- Gear, J.I., Cox, M.G., Gustafsson, J., Gleisner, K.S., Murray, I., Glatting, G., Konijnenberg, M., Flux, G.D., 2018. EANM practical guidance on uncertainty analysis for molecular radiotherapy absorbed dose calculations. *Eur. J. Nucl. Med. Mol. Imaging* 45 (13), 2456–2474.
- Goodfellow, I.J., Vinyals, O., Saxe, A.M., 2014. Qualitatively characterizing neural network optimization problems. arXiv preprint arXiv:1412.6544.
- Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q., 2017. On calibration of modern neural networks. In: International Conference on Machine Learning. PMLR, pp. 1321–1330.
- Ha, I.Y., Wilms, M., Heinrich, M., 2020. Semantically guided large deformation estimation with deep networks. *Sensors* 20 (5), 1392.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.
- Heinrich, M.P., Jenkinson, M., Brady, M., Schnabel, J.A., 2013a. MRF-based deformable registration and ventilation estimation of lung CT. *IEEE Trans. Med. Imaging* 32 (7), 1239–1248.
- Heinrich, M.P., Jenkinson, M., Papiez, B.W., Brady, M., Schnabel, J.A., 2013b. Towards realtime multimodal fusion for image-guided interventions using self-similarities. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 187–194.
- Heinrich, M.P., Maier, O., Handels, H., 2015. Multi-modal multi-atlas segmentation using discrete optimisation and self-similarities. In: VISCERAL Challenge@ ISBI, vol. 1390, p. 27.
- Hering, A., Hansen, L., Mok, T.C., Chung, A., Siebert, H., Häger, S., Lange, A., Kuckertz, S., Heldmann, S., Shao, W., et al., 2021. Learn2reg: comprehensive multi-task medical image registration challenge, dataset and evaluation in the era of deep learning. arXiv preprint arXiv:2112.04489.
- Hernandez, M., Bossa, M.N., Olmos, S., 2009. Registration of anatomical images using paths of diffeomorphisms parameterized with stationary vector field flows. *Int. J. Comput. Vis.* 85 (3), 291–306.
- Hoffmann, M., Billot, B., Iglesias, J.E., Fischl, B., Dalca, A.V., 2020. Learning image registration without images. arXiv preprint arXiv:2004.10282.
- Hoopes, A., Hoffmann, M., Fischl, B., Guttag, J., Dalca, A.V., 2021. Hypermorph: amortized hyperparameter learning for image registration. In: International Conference on Information Processing in Medical Imaging. Springer, pp. 3–17.
- Im, D.J., Tao, M., Branson, K., 2016. An empirical analysis of deep network loss surfaces.
- Iseñee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H., 2021. nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods* 18 (2), 203–211.
- Jaderberg, M., Simonyan, K., Zisserman, A., et al., 2015. Spatial transformer networks. *Adv. Neural Inf. Process. Syst.* 28, 2017–2025.
- Jha, D., Smedsrød, P.H., Riegler, M.A., Johansen, D., De Lange, T., Halvorsen, P., Johansen, H.D., 2019. Resunet++: An advanced architecture for medical image segmentation. In: 2019 IEEE International Symposium on Multimedia. ISM, IEEE, pp. 225–2255.
- Johnson, H.J., Christensen, G.E., 2002. Consistent landmark and intensity-based image registration. *IEEE Trans. Med. Imaging* 21 (5), 450–461.
- Kendall, A., Gal, Y., 2017. What uncertainties do we need in Bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems, vol. 2017-Decem. Neural information processing systems foundation, pp. 5575–5585, arXiv:1703.04977.
- Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P., 2016. On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836.
- Kim, B., Kim, D.H., Park, S.H., Kim, J., Lee, J.-G., Ye, J.C., 2021. CycleMorph: Cycle consistent unsupervised deformable image registration. *Med. Image Anal.* 71, 102036.
- Kingma, D.P., Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.
- Krebs, J., Delingette, H., Mailhé, B., Ayache, N., Mansi, T., 2019. Learning a probabilistic model for diffeomorphic registration. *IEEE Trans. Med. Imaging* 38 (9), 2165–2176.
- Kuleshov, V., Fenner, N., Ermon, S., 2018. Accurate uncertainties for deep learning using calibrated regression. In: International Conference on Machine Learning. PMLR, pp. 2796–2804.
- Kybík, J., 2009. Bootstrap resampling for image registration uncertainty estimation without ground truth. *IEEE Trans. Image Process.* 19 (1), 64–73.
- Laves, M.-H., Ihler, S., Fast, J.F., Kahrs, L.A., Ortmaier, T., 2020a. Well-calibrated regression uncertainty in medical imaging with deep learning. In: Arbel, T., Ben Ayed, I., de Brujne, M., Descoteaux, M., Lombaert, H., Pal, C. (Eds.), Proceedings of the Third Conference on Medical Imaging with Deep Learning. In: Proceedings of Machine Learning Research, vol. 121, PMLR, pp. 393–412, URL <https://proceedings.mlr.press/v121/laves20a.html>.
- Laves, M.-H., Ihler, S., Kortmann, K.-P., Ortmaier, T., 2019. Well-calibrated model uncertainty with temperature scaling for dropout variational inference. arXiv preprint arXiv:1909.13550.
- Laves, M.-H., Tölle, M., Ortmaier, T., 2020b. Uncertainty estimation in medical image denoising with Bayesian deep image prior. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 12443 LNCS, 81–96, arXiv:2008.08837.
- Laves, M.-H., Tölle, M., Ortmaier, T., 2020c. Uncertainty estimation in medical image denoising with bayesian deep image prior. In: Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis. Springer, pp. 81–96.
- Lei, Y., Fu, Y., Wang, T., Liu, Y., Patel, P., Curran, W.J., Liu, T., Yang, X., 2020. 4D-CT deformable image registration using multiscale unsupervised deep learning. *Phys. Med. Biol.* 65 (8), 085003.
- Levi, D., Gispan, L., Giladi, N., Fetaya, E., 2019. Evaluating and calibrating uncertainty prediction in regression tasks. arXiv preprint arXiv:1905.11659.
- Li, S., Sui, X., Luo, X., Xu, X., Liu, Y., Goh, R.S.M., 2021. Medical image segmentation using squeeze-and-expansion transformers. arXiv preprint arXiv:2105.09511.
- Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T., 2018. Visualizing the loss landscape of neural nets. *Adv. Neural Inf. Process. Syst.* 31.
- Lian, C., Liu, M., Zhang, J., Shen, D., 2018. Hierarchical fully convolutional network for joint atrophy localization and alzheimer's disease diagnosis using structural MRI. *IEEE Trans. Pattern Anal. Mach. Intell.* 42 (4), 880–893.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021a. Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030.
- Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S., 2022. A ConvNet for the 2020s. arXiv preprint arXiv:2201.03545.
- Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H., 2021b. Video swin transformer. arXiv preprint arXiv:2106.13230.
- Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440.
- Looney, C.G., 1996. Stabilization and speedup of convergence in training feedforward neural networks. *Neurocomputing* 10 (1), 7–31.
- Luo, W., Li, Y., Urtasun, R., Zemel, R., 2016. Understanding the effective receptive field in deep convolutional neural networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 4905–4913.
- Luo, J., Sedghi, A., Popuri, K., Cobzas, D., Zhang, M., Preiswerk, F., Toews, M., Golby, A., Sugiyama, M., Wells, W.M., et al., 2019. On the applicability of registration uncertainty. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 410–419.
- Lv, J., Wang, Z., Shi, H., Zhang, H., Wang, S., Wang, Y., Li, Q., 2022. Joint progressive and coarse-to-fine registration of brain MRI via deformation field integration and non-rigid feature fusion. *IEEE Trans. Med. Imaging*.
- Maas, A.L., Hannun, A.Y., Ng, A.Y., et al., 2013. Rectifier nonlinearities improve neural network acoustic models. *Proc. Icmi* 30 (1), 3.
- Marcus, D.S., Wang, T.H., Parker, J., Csernansky, J.G., Morris, J.C., Buckner, R.L., 2007. Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *J. Cogn. Neurosci.* 19 (9), 1498–1507.
- Mehrash, A., Wells, W.M., Tempany, C.M., Abolmaesumi, P., Kapur, T., 2020. Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. *IEEE Trans. Med. Imaging* 39 (12), 3868–3878.
- Milletari, F., Navab, N., Ahmadi, S.-A., 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). IEEE, pp. 565–571.
- Modat, M., Ridgway, G.R., Taylor, Z.A., Lehmann, M., Barnes, J., Hawkes, D.J., Fox, N.C., Ourselin, S., 2010. Fast free-form deformation using graphics processing units. *Comput. Methods Programs Biomed.* 98 (3), 278–284.
- Mok, T.C., Chung, A., 2020. Fast symmetric diffeomorphic image registration with convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4644–4653.
- Mok, T.C., Chung, A., 2021. Conditional deformable image registration with convolutional neural network. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 35–45.
- Onofrey, J.A., Staib, L.H., Papademetris, X., 2013. Semi-supervised learning of nonrigid deformations for image registration. In: International MICCAI Workshop on Medical Computer Vision. Springer, pp. 13–23.
- Pace, D.F., Aylward, S.R., Niethammer, M., 2013. A locally adaptive regularization based on anisotropic diffusion for deformable image registration of sliding organs. *IEEE Trans. Med. Imaging* 32 (11), 2114–2126.
- Park, N., Kim, S., 2022. How do vision transformers work? arXiv preprint arXiv:2202.06709.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* 32, 8026–8037.
- Phan, B., Salay, R., Czarnecki, K., Abdelzad, V., Denouden, T., Vernekar, S., 2018. Calibrating uncertainties in object localization task. arXiv preprint arXiv:1811.11210.

- Qiu, H., Qin, C., Schuh, A., Hammernik, K., Rueckert, D., 2021. Learning diffeomorphic and modality-invariant registration using B-splines. In: Medical Imaging with Deep Learning.
- Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A., 2021. Do vision transformers see like convolutional neural networks? arXiv preprint arXiv:2108.08810.
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788.
- Risholm, P., Balter, J., Wells, W.M., 2011. Estimation of delivered dose in radiotherapy: the influence of registration uncertainty. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 548–555.
- Risholm, P., Janoos, F., Norton, I., Golby, A.J., Wells III, W.M., 2013. Bayesian characterization of uncertainty in intra-subject non-rigid registration. *Med. Image Anal.* 17 (5), 538–555.
- Rohé, M.-M., Datar, M., Heimann, T., Sermesant, M., Pennec, X., 2017. SVF-net: Learning deformable image registration using shape matching. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 266–274.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.
- Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L., Leach, M.O., Hawkes, D.J., 1999. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Trans. Med. Imaging* 18 (8), 712–721.
- Santurkar, S., Tsipras, D., Ilyas, A., Madry, A., 2018. How does batch normalization help optimization? *Adv. Neural Inf. Process. Syst.* 31.
- Segars, W., Bond, J., Frush, J., Hon, S., Eckersley, G., Williams, C.H., Feng, J., Tward, D.J., Ratnanather, J., Miller, M., et al., 2013. Population of anatomically variable 4D XCAT adult phantoms for imaging research and optimization. *Med. Phys.* 40 (4), 043701.
- Segars, W.P., Sturgeon, G., Mendonca, S., Grimes, J., Tsui, B.M., 2010. 4D XCAT phantom for multimodality imaging research. *Med. Phys.* 37 (9), 4902–4915.
- Siebert, H., Hansen, L., Heinrich, M.P., 2021. Fast 3D registration with accurate optimisation and little learning for Learn2Reg 2021. arXiv preprint arXiv:2112.03053.
- Simpson, I.J., Woolrich, M.W., Groves, A.R., Schnabel, J.A., 2011. Longitudinal brain MRI analysis with uncertain registration. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 647–654.
- Smith, L.N., Topin, N., 2019. Super-convergence: Very fast training of neural networks using large learning rates. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, vol. 11006. International Society for Optics and Photonics, 1100612.
- Sokooti, H., De Vos, B., Berendsen, F., Lelieveldt, B.P., Isgum, I., Staring, M., 2017. Nonrigid image registration using multi-scale 3D convolutional neural networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 232–239.
- Sutskever, I., Martens, J., Dahl, G., Hinton, G., 2013. On the importance of initialization and momentum in deep learning. In: International Conference on Machine Learning. PMLR, pp. 1139–1147.
- Tölle, M., Laves, M.-H., Schlaefer, A., 2021. A mean-field variational inference approach to deep image prior for inverse problems in medical imaging. *Med. Imag. Deep Learn.* 698–713, URL https://openreview.net/forum?id=DvV5C_bkLiB4.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. arXiv preprint arXiv:1706.03762.
- Vercauteren, T., Pennec, X., Perchant, A., Ayache, N., 2009. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage* 45 (1), S61–S72.
- Vickress, J., Battista, J., Barnett, R., Yartsev, S., 2017. Representing the dosimetric impact of deformable image registration errors. *Phys. Med. Biol.* 62 (17), N391.
- Viola, P., Wells III, W.M., 1997. Alignment by maximization of mutual information. *Int. J. Comput. Vis.* 24 (2), 137–154.
- Vishnevskiy, V., Gass, T., Szekely, G., Tanner, C., Goksel, O., 2016. Isotropic total variation regularization of displacements in parametric image registration. *IEEE Trans. Med. Imaging* 36 (2), 385–395.
- Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 13 (4), 600–612.
- Wang, W., Chen, C., Ding, M., Li, J., Yu, H., Zha, S., 2021a. Transbts: Multimodal brain tumor segmentation using transformer. arXiv preprint arXiv:2103.04430.
- Wang, D., Wu, Z., Yu, H., 2021b. TED-net: Convolution-free T2T vision transformer-based encoder-decoder dilation network for low-dose CT denoising. arXiv preprint arXiv:2106.04650.
- Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L., 2021c. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv preprint arXiv:2102.12122.
- Wolberg, G., Zokai, S., 2000. Robust image registration using log-polar transform. In: Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101), vol. 1. IEEE, pp. 493–496.
- Xie, Y., Zhang, J., Shen, C., Xia, Y., 2021. CoTr: Efficiently bridging CNN and transformer for 3D medical image segmentation. arXiv preprint arXiv:2103.03024.
- Xu, Z., Luo, J., Lu, D., Yan, J., Frisken, S., Jagadeesan, J., Wells III, W., Li, X., Zheng, Y., Tong, R., 2022. Double-uncertainty guided spatial and temporal consistency regularization weighting for learning-based abdominal registration. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer.
- Yang, X., Kwitt, R., Niethammer, M., 2016. Fast predictive image registration. In: Deep Learning and Data Labeling for Medical Applications. Springer, pp. 48–57.
- Yang, X., Kwitt, R., Styner, M., Niethammer, M., 2017a. Fast predictive multimodal image registration. In: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017). IEEE, pp. 858–862.
- Yang, X., Kwitt, R., Styner, M., Niethammer, M., 2017b. Quicksilver: Fast predictive image registration—a deep learning approach. *NeuroImage* 158, 378–396.
- Zeiler, M.D., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q.V., Nguyen, P., Senior, A., Vanhoucke, V., Dean, J., et al., 2013. On rectified linear units for speech processing. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, pp. 3517–3521.
- Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L., 2022. Scaling vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12104–12113.
- Zhang, J., 2018. Inverse-consistent deep networks for unsupervised deformable image registration. arXiv preprint arXiv:1809.03443.
- Zhang, Y., Ma, J., Iyengar, P., Zhong, Y., Wang, J., 2017. A new CT reconstruction technique using adaptive deformation recovery and intensity correction (ADRIC). *Med. Phys.* 44 (6), 2223–2241.
- Zhang, Z., Yu, L., Liang, X., Zhao, W., Xing, L., 2021. Transt: Dual-path transformer for low dose computed tomography. arXiv preprint arXiv:2103.00634.
- Zhou, H.-Y., Guo, J., Zhang, Y., Yu, L., Wang, L., Yu, Y., 2021. Nnformer: Interleaved transformer for volumetric segmentation. arXiv:2109.03201.
- Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J., 2019. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging* 39 (6), 1856–1867.
- Zhu, B., Liu, J.Z., Cauley, S.F., Rosen, B.R., Rosen, M.S., 2018. Image reconstruction by domain-transform manifold learning. *Nature* 555 (7697), 487–492.