Automated Learning for Deformable Medical Image Registration by Jointly Optimizing Network Architectures and Objective Functions

Xin Fan *Senior Member, IEEE*, Zi Li, Ziyang Li, Xiaolin Wang, Risheng Liu, *Member, IEEE,* Zhongxuan Luo, and Hao Huang

***Abstract*—Deformable image registration plays a critical role in various tasks of medical image analysis. A successful registration algorithm, either derived from conventional energy optimization or deep networks, requires tremendous efforts from computer experts to well design registration energy or to carefully tune network architectures with respect to medical data available for a given registration task/scenario. This paper proposes an automated learning registration algorithm (AutoReg) that cooperatively optimizes both architectures and their corresponding training objectives, enabling non-computer experts to conveniently find off-the-shelf registration algorithms for various registration scenarios. Specifically, we establish a triple-level framework to embrace the searching for both network architectures and objectives with a cooperating optimization. Extensive experiments on multiple volumetric datasets and various registration scenarios demonstrate that AutoReg can automatically learn an optimal deep registration network for given volumes and achieve state-of-the-art performance. The automatically learned network also improves computational efficiency over the mainstream UNet architecture from 0.558 to 0.270 seconds for a volume pair on the same configuration.**

***Index Terms*—Medical image registration, Automatic machine learning, Neural architecture search, Hyperparameter optimization, Convolution neural network.**

I. Introduction

DEFORMABLE image registration (DIR) establishes dense spatial correspondences between different medical image acquisitions [1]. It plays a critical role in various tasks of medical image analysis including anatomical change diagnosis [2], longitudinal studies [3] and statistical atlas building [4]. Given a source image $\mathbf{s}$ and a target image $\mathbf{t}$ on a spatial domain $\Omega \in \mathbb{R}^d$, the goal of DIR is to find an optimal non-linear dense transformation or field $\boldsymbol{\varphi} : \Omega \times \mathbb{R} \rightarrow \Omega$ that minimizes the energy:

$$\min_{\boldsymbol{\varphi}} E_D(\boldsymbol{\varphi}; f \circ \mathbf{s}, f \circ \mathbf{t}) + E_R(\boldsymbol{\varphi}), \tag{1}$$

where E_D is a data matching term, evaluating the similarity between aligned features $f \circ \mathbf{s}$ and $f \circ \mathbf{t}$ of the source and target while E_R is a regularization term that reflects the nature of the transformation. Medical image analysis faces a wide range of registration scenarios involving different facets of developing a registration algorithm [5]: estimating spatial correspondences relies on distinct features robust to intensity variations from intra/inter subjects and acquisition equipment; the deforming complexity varies with the nature of anatomical organs; the similarity metric for registering images of the same modality is evidently different from that for cross-modality images. Hence, the three major components, *i.e.*, a feature extractor, a deformation model, and an objective function, have to be well designed in order to construct the energy (1) for a specific scenario, as shown in the left of Fig. 1(a). This modeling process demands deep understanding of the registration scenario as well as sophisticated mathematical skills. Meanwhile, the energy optimization invokes a computationally intensive process, typically taking minutes or even hours for registration [6]–[8].

Witnessing the great success of deep learning (DL) in many computer vision tasks, researchers replace manually-crafted feature extractors and parametric deformation models with deep networks so that DIR turns out to be one step prediction with network parameters (weights) learned from image pairs, rendering faster deformation estimation than iterative optimization [9]–[17]. Nevertheless, these deep learning based methods highly depend on training examples available for a scenario. One has to re-train the network when applying to a new scenario with significantly different data. In many occasions, tuning network architectures and/or loss functions is also necessary to gain optimal performance on the new scenario, as shown in the right of Fig. 1(a). For example, a network trained for unimodal registration tasks cannot accurately align images of different modalities. Recent studies [18], [19] present new training strategies that automatically find optimal hyper-parameters for loss functions. Unfortunately, algorithm developers still have to adjust network architectures for feature extraction and/or deformation by exploring numerous possi-

X. Fan, Z. Li, Z. Li, X. Wang, R. Liu, and Z. Luo are with the DUT-RU International School of Information Science & Engineering and the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology, Dalian 116024, China. Z. Luo is also with the Institute of Artificial Intelligence, Guilin University of Electronic Technology, Guilin 541004, China. (E-mail: xin.fan@ieee.org; alisonbrielee@gmail.com; liziyang1997@mail.dlut.edu.cn; wxl1009@mail.dlut.edu.cn; rsliu@dlut.edu.cn; zxluo@dlut.edu.cn). (Corresponding author: X. Fan).

H. Huang is with the Department of Radiology, Children's Hospital of Philadelphia, Philadelphia, PA, United States. H. Huang is also with the Department of Radiology, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, United States (E-mail: huangh6@email.chop.edu).

Manuscript received April 19, 2021; revised August 16, 2021.

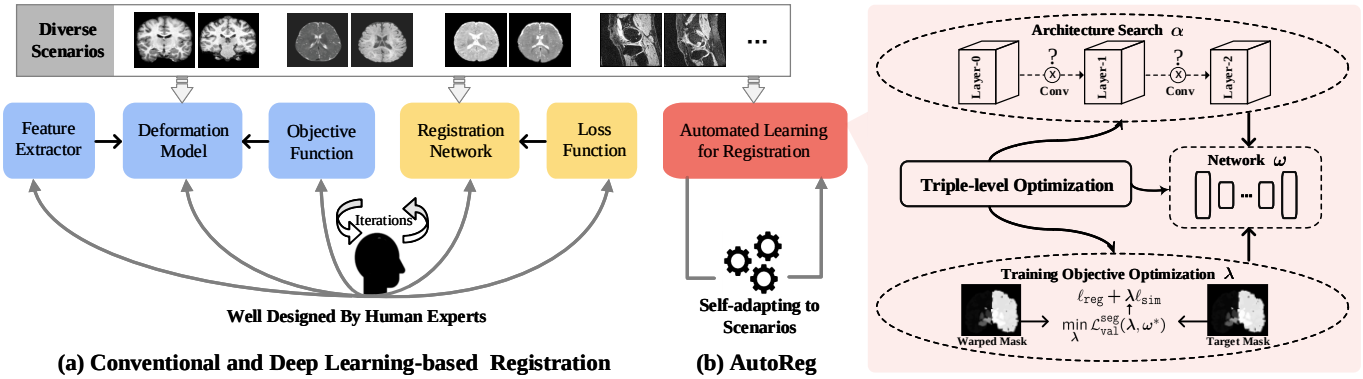

Fig. 1. (a) Both conventional and learning-based registration techniques require computer experts to well design core components for different medical image registration tasks, and (b) AutoReg, a user-friendly automatic registration framework, learns off-the-shelf deep registration algorithms for various scenarios by jointly optimizing the network architecture $\boldsymbol{\alpha}$, hyper-parameters $\boldsymbol{\lambda}$ in training objective, and network parameters $\boldsymbol{\omega}$.

bilities. This manually train-trial process is extremely time consuming since one single training stage may take hours or even days.

A successful DIR algorithm, either derived from conventional energy optimization or deep networks, requires tremendous efforts from computer experts to well design registration energy or to carefully tune network architectures. Therefore, existing paradigms prohibit medical/clinical users from exploring registration adaptive to their scenarios and available data. A new user-friendly automatic registration paradigm is desired to enable medical/clinical users even without algorithm-developing expertise to conveniently find off-the-shelf registration algorithms for various scenarios.

A. Our Contributions

We propose a triple-level optimization framework that jointly learns the weights, architecture, and loss function of a deep network for DIR, as shown in Fig. 1(b). Specifically, besides one level of optimization to learn network weights, we formulate one level to search optimal architectures for *feature* and *deformation* networks and another level to discover proper hyper-parameters for *objective functions*. Consequently, our framework enables automatically learning nearly all major components of a registration algorithm tailored to a given scenario, and thus frees both medical researchers and computer engineers from enormous efforts to sophisticated modeling and tedious parameter tuning. Moreover, the auto-learned registration also runs extremely fast inheriting from common deep-based algorithms. We summarize our contributions below:

- We devise an automated learning registration algorithm (AutoReg) to cooperatively optimize the architecture and loss function of a deep registration network. To our best knowledge, this is the first piece of work that achieves Automated Machine Learning (AutoML)[1] for medical image registration.
- To efficiently discover the architecture adaptive to the registration scenario of interest, we construct a search space with an adaptive feature cell and a task-aware deformation cell, which consist of various convolution operators practically effective to feature learning and deformation estimation, respectively.
- We introduce the similarity from two aspects of the intensity distribution and structural preservation into a unified loss function and further optimize the hyper-parameters weighing the similarity and regularization terms in order to deploy the scenario-oriented training loss.
- We convert the triple-level optimization into hierarchical bilevel sub-problems and then iteratively apply efficient gradient-based algorithms to the complicated optimization that couples variables of different natures. This auto-search mechanism with cooperating optimization drastically alleviates time expenses compared with the manual train-trial process.

We validate the AutoReg algorithm on several typical MRI registration scenarios ranging from mono-modality (registering brain T1 images to an averaged T1 template, brain T1 to T1 images, brain T2 to T2 images, knee T1 to T1 images, and lung CT inspiration to expiration images) to multi-modality (registering brain T1 to T2 images). Despite of significant data variations occurring in these scenarios, AutoReg can learn an efficient algorithm that outputs more accurate deformation fields than the state-of-the-art algorithms re-trained by the data from respective scenarios.

This paper is organized as follows: we firstly describe related works in Section 2 and introduce our automated learning for registration in Section 3. Next, we demonstrate ablation studies and experimental comparisons in Section 4. Section 5 concludes this paper.

II. Related Works

This section briefly reviews recent deep algorithms for DIR and AutoML techniques for computer vision tasks other than image registration.

A. Deep Learning-based Deformable Image Registration

Deep learning based algorithms have gained impressive progress in DIR by taking advantage of powerful representation ability of convolutional neural networks. Inspired by the

[1]AutoML automates the time-consuming, iterative tasks of ML development, making ML available for non-ML experts.

spatial-transformer work [20], DL-based approaches employ unsupervised feature learning in the training process and apply one-step inference rather than costly iterative optimization in conventional registration to generate deformation fields [9], [10]. Further studies estimate the velocity or momentum fields to impose diffeomorphism to the final deformations [11], [12]. Alternatively, Niethammer *et al.* embed a spatially-varying regularizer into a feature network [21]. In our previous work, we constitute an unsupervised deep network to simultaneously learn deformation maps and features [13].

Recent attempts improve the generalization ability of deep registration adapting to a wider range of scenarios. Hoopes *et al.* introduce a hyper-parameter and modulate a registration hypernetwork in order to produce an optimal deformation field given the hyper-parameter value [18]. The work of [19] proposes a bilevel training strategy that learns optimal hyper-parameters of training loss functions for deep networks of DIR. Nevertheless, existing methods can learn a subset of major components of a deep registration network from given data, but manually designing and/or tuning other components still requires tremendous human experience and efforts.

B. AutoML for Computer Vision Tasks

We focus on Neural Architecture Search (NAS) that automates designing architectures of deep neural networks for computer vision tasks. NAS has drawn increasing attention owing to its capability of relieving intensive human labors meanwhile outputting considerable performance. In pioneering works, researchers pose the problem of searching for an optimal network architecture as discrete optimization, and thus develop either reinforcement learning [22] or evolutionary algorithms [23] to find the solution to optimal network architectures for classification tasks. Unfortunately, these discrete optimization strategies are computationally demanding. Liu *et al.* propose the continuous relaxation of the discrete search space and take differentiable optimization techniques in order to efficiently learn deep network architectures for image classification and language modeling tasks [24]. These differentiable NAS strategies can significantly reduce the search time from months/days to even hours, making NAS viable for common users [25], [26].

These NAS techniques have been applied to various types of deep networks for many CV tasks including low-level vision, image segmentation, and object detection [27]–[32]. Recently, researchers employ NAS to search UNet-like architectures for medical image segmentation [33]–[35]. Automated designing networks for DIR desires searching heterogeneous networks for both feature and deformation learning. This study develops such an algorithm along with hyper-parameter optimization, which fully automates learning optimal DIR algorithms adaptive to various scenarios.

III. Automated Learning for Deformable Image Registration

We construct a triple-level algorithmic framework to jointly optimize the network architecture and training objective so that we can automate the process of designing three major components for DIR, *i.e.*, feature extraction, deformation model, and objective function, given a set of data for a scenario. This section begins with the optimization formulation, then gives the two levels of optimization for architecture and objective, and ends with the optimization/learning process.

A. Problem Formulation

In a general deep learning-based framework for DIR, the spatial correspondence map $\boldsymbol{\varphi}$ between the source $\mathbf{s}$ and targe $\mathbf{t}$ is represented by a deep network $\Psi(\boldsymbol{\omega}; \mathbf{s}, \mathbf{t})$, where $\boldsymbol{\omega}$ denotes learnable network parameters of Ψ. Taking the inputs $\mathbf{s}$ and $\mathbf{t}$, the prediction or inference of Ψ upon the learned $\boldsymbol{\omega}$ yields the deformable map [9].

Herein, we introduce an architecture parameter $\boldsymbol{\alpha}$ that represents a series of covolutional operators constituting the network. Meanwhile, the training objective for Ψ contains hyper-parameters $\boldsymbol{\lambda}$ that weigh the objective terms for similarity and regularization. We formulate jointly learning $\boldsymbol{\alpha}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ as triple-level optimization:

$$s.t. \begin{cases} \min\limits_{\boldsymbol{\lambda}} \ \mathcal{L}_{\texttt{val}}^{\texttt{seg}}(\boldsymbol{\lambda}, \boldsymbol{\alpha}^*, \boldsymbol{\omega}^*; \mathbf{s}, \mathbf{t}), \\ \boldsymbol{\alpha}^*(\boldsymbol{\lambda}) = \arg\min\limits_{\boldsymbol{\alpha}} \ \mathcal{L}_{\texttt{val}}^{\texttt{reg}}(\boldsymbol{\alpha}, \boldsymbol{\omega}^*(\boldsymbol{\alpha}); \boldsymbol{\lambda}, \mathbf{s}, \mathbf{t}), \\ s.t. \ \boldsymbol{\omega}^*(\boldsymbol{\alpha}) = \arg\min\limits_{\boldsymbol{\omega}} \ \mathcal{L}_{\texttt{tr}}^{\texttt{reg}}(\boldsymbol{\omega}; \boldsymbol{\alpha}, \boldsymbol{\lambda}, \mathbf{s}, \mathbf{t}). \end{cases} \tag{2}$$

The first level of optimization trains the network parameters $\boldsymbol{\omega}$ using a training data set for a scenario given a network architecture and a loss function $\mathcal{L}_{\texttt{tr}}^{\texttt{reg}}$ evaluating similarity between the target and deformed source. The second and third levels learn $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ from a validation set for the same scenario, respectively. The training losses of $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ share the same definition $\mathcal{L}^{\texttt{reg}}$ as the learned network $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\omega})$ given by $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ determines the deformation map for registration. We define the loss function $\mathcal{L}_{\texttt{val}}^{\texttt{seg}}$ for learning $\boldsymbol{\lambda}$ by evaluating the overlapped anatomical regions between the target and deformed source. This definition is inspired by the use of Dice scores when manually tuning hyper-parameters $\boldsymbol{\lambda}$. Our formulation embraces learning the three major components for DIR as $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\omega})$ implies feature extraction and deformation models while $\boldsymbol{\lambda}$ adjusts objective functions, shown in Fig. 1(b).

B. Architecture Search

Both feature extraction and deformation generation take a three-layer convolutional network, running at two scales, shown as the dark and purple blocks in the top of Fig. 2(a). We take this multi-scale architecture as the backbone because extensive studies on brain MRI registration validate its superior performance [13], [19]. However, image intensities and/or organ geometries significantly vary with different registration scenarios. Computer engineers have to devote great efforts to trial various types and kernel sizes of convolutional operators applied between layers, that are central building blocks for neural networks, in order to accommodating significant data variations. This study strives to automatically discover appropriate operations, and hence free users from tedious hand-crafted trial-tests.

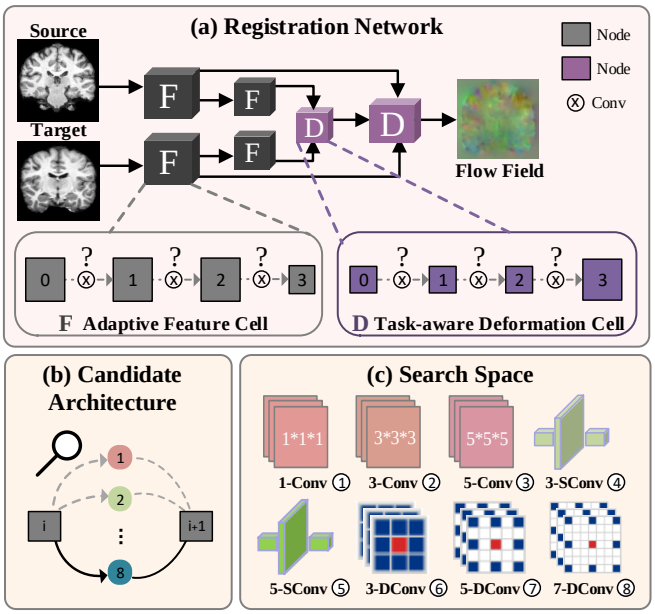

Fig. 2. (a) The network backbone consisting of cells for feature extraction F and deformation generation D at multiple scales, (b) the structure for searching cells, and (c) eight candidate operations.

Specifically, we construct an Adaptive Feature Cell and a Task-aware Deformation Cell at each scale to represent candidate architectures for feature learning and deformation generation, respectively, shown in the bottom of Fig. 2(a). These two types of cells share a common structure, *i.e.*, a directed acyclic graph (DAG) with four nodes, whose edges are candidate convolutional operations. The first node is the network input while the other three represent the outputs by applying the edge operator to the previous node. Figure 2(b) illustrates the structure of two consecutive nodes. The cells differ in the first and/or last nodes since their input and output are different as discussed below.

Adaptive Feature Cell: We take the input source or target image as the first node while down-sample the output of the last convolution by taking the stride of 2 as the last node. We learn one common cell for both source and target so that they share the same network architecture but have different network parameters given the architecture.

Task-aware Deformation Cell: We concatenate the outputs of the feature learning network for the source and target images as the first node. Meanwhile, we up-sample the output at the last convolution by a resizing operation as the last node. To obtain a diffeomorphism deformation, we parameterize the deformation field with a stationary velocity field, and thus generate the final field under the government of ordinary differential equations [36].

As shown in Fig. 2(b), we set eight convoutional candidates as the edge connecting two nodes including three types of convolutions, *i.e.*, regular, separable, and dilated ones, with two or three kernel sizes. In practice, a computer engineer typically attempts to numerate the combinations of these operations as much as possible in order to gain improvements for a new registration scenario. Figure 2(c) visualizes the eight operations listed below.

- $1\times1\times1$ Conv (1-Conv)
- $3\times3\times3$ Conv (3-Conv)
- $5\times5\times5$ Conv (5-Conv)
- $3\times3\times3$ Separable Conv (3-SConv)
- $5\times5\times5$ Separable Conv (5-SConv)
- $3\times3\times3$ Dilation Conv (3-DConv)
- $5\times5\times5$ Dilation Conv (5-DConv)
- $7\times7\times7$ Dilation Conv (7-DConv)

Therefore, the architecture search becomes finding an optimal path composed of six convolutional operations [2] traversing the two cells at every scale. Our search space spans $8^{(3+3)}$ at one scale and about 50 thousand in total (two scales) possible candidates.

We follow the idea of continuous relaxation in [24] to convert the time consuming discrete search over a huge space into a continuous optimization problem where efficient gradient-based algorithms are applicable. Let $o(\cdot)$ denotes one operation of the candidate set $\mathcal{O}$ to be applied to the node $x^{(i)}$. We parameterize the weights of candidate operations for a pair of nodes $x^{(i)}$ and $x^{(i+1)}$ as a column vector $\alpha_o^{(i,i+1)}$ of the dimension $|\mathcal{O}|$ (eight for our setting). We relax the discrete choice of a particular operation to a softmax over all possible operations:

$$\bar{o}^{(i,i+1)}(x^{(i)}) = \sum_{o\in\mathcal{O}} \frac{\exp(\alpha_o^{(i,i+1)})}{\sum_{o'\in\mathcal{O}} \exp(\alpha_{o'}^{(i,i+1)})} o(x^{(i)}). \quad (3)$$

Thereby, searching network architecture reduces to computing a set of continuous variables $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_F, \boldsymbol{\alpha}_D\} = \{\alpha_F^{(i,i+1)}, \alpha_D^{(i,i+1)}\}_{i=0,1,2}$, and finally taking each edge with the most likely operation, $o^{(i,i+1)} = \arg\max_{o\in\mathcal{O}} \alpha_o^{(i,i+1)}$. We will detail the gradient-based optimization of $\boldsymbol{\alpha}$ in Sec. III-D.

C. Training Objective Optimization

Training objectives for deep networks determine both training convergence and prediction accuracy. Our triple-level formulation enables bringing additional cues for different levels of optimization so that it involves two types of objective losses, $\mathcal{L}^{\texttt{reg}}$ and $\mathcal{L}^{\texttt{seg}}$, for training the network architectures $\boldsymbol{\alpha}$ and parameters $\boldsymbol{\omega}$, and the hyper-parameters $\boldsymbol{\lambda}$, respectively.

We define the similarity term in the objective $\mathcal{L}^{\texttt{reg}}$ from two aspects, *i.e.*, intensity distribution and structural consistency, in order to accommodate a wide range of scenarios including mono-modality and cross-modality registrations. The local Normalized Cross-correlation Coefficient (NCC) evaluates the similarity of intensity distributions between an image pair from the same modality. Nevertheless, NCC can hardly characterize the similarity when evident intensity discrepancies exist especially for images from different modalities. Hence, we introduce the Modality Independent Neighborhood Descriptor (MIND) [37] into similarity evaluation. MIND extracts various types of geometrical structures, *e.g.*, corners and edges, in a local neighborhood of an image, and can thus evaluate the structural consistence between images across modalities. We

[2]All these convolutions are followed by a Leaky ReLU function.

combine NCC and MIND into one similarity metric by a trade-off parameter λ_1:

$$\ell_{\texttt{sim}} = \lambda_1 \ell_{\texttt{NCC}} + (1-\lambda_1)\ell_{\texttt{MIND}}. \tag{4}$$

We also incorporate a diffusion regularizer on spatial gradients of deformation fields [9] together with multi-scale similarities [38] into the registration loss $\mathcal{L}^{\texttt{reg}}$:

$$\mathcal{L}^{\texttt{reg}} = \ell^{1}_{\texttt{sim}} + \lambda_2 \ell_{\texttt{smooth}} + \lambda_3 \ell^{1/2}_{\texttt{sim}} + \lambda_4 \ell^{1/4}_{\texttt{sim}}, \tag{5}$$

where $\ell^{1}_{\texttt{sim}}$, $\ell^{1/2}_{\texttt{sim}}$, and $\ell^{1/4}_{\texttt{sim}}$ denote similarity loss functions at full, half, and quarter resolutions, respectively, and $\ell_{\texttt{smooth}}$ represents the regularization loss. We employ the diffusion regularizer on spatial gradients of the registration fields. We assemble all these scalar hyper-parameters into one vector $\boldsymbol{\lambda} := \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ and optimize $\boldsymbol{\lambda}$ with respect to a given validate set for a scenario.

A human expert may visually inspect the coincidence of aligned regions with anatomical parcellation when tuning the hyper-parameters $\boldsymbol{\lambda}$. From this respect, we leverage an auxiliary segmentation map, which assigns each voxel to an anatomical structure, to define the loss $\mathcal{L}^{\texttt{seg}}$ for the top level of optimization in (2) that automatically learns a scenario-specific $\boldsymbol{\lambda}$ as shown in Fig. 1(b). The regions in the warped source by an accurate deformation field $\mathbf{s} \circ \boldsymbol{\varphi}$ would be perfectly coincident with those in the target $\mathbf{t}$ from the same anatomical structure. We quantify the volume overlapping for all structures using the Dice score [39] and define the loss as:

$$\mathcal{L}^{\texttt{seg}} = \ell_{\texttt{dice}}(\mathbf{s_m} \circ \boldsymbol{\varphi}, \mathbf{t_m}), \tag{6}$$

where $\mathbf{s_m}$ and $\mathbf{t_m}$ are the segmentation maps for $\mathbf{s}$ and $\mathbf{t}$, respectively. Similar to [9], we convert the segmentation masks to the one-hot format and spatially transform them using linear interpolation when computing the loss.

D. Optimization Procedure

It is challenging to solve the triple-level optimization in Eq. (2) owing to coupling several large-scale optimization problems of different natures. We convert the triple-level model into hierarchical bilevel optimization by fixing the other variables when optimizing one. We first tackle the bi-level optimization of the architecture $\boldsymbol{\alpha}_F, \boldsymbol{\alpha}_D$ and network parameters $\boldsymbol{\omega}$ given the hyper-parameter $\boldsymbol{\lambda}$ and training objective:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}_F, \boldsymbol{\alpha}_D} \ \mathcal{L}^{\texttt{reg}}_{\texttt{val}}(\boldsymbol{\alpha}_F, \boldsymbol{\alpha}_D, \boldsymbol{\omega}^*), \\ &s.t.\ \boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} \ \mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}, \boldsymbol{\alpha}_F, \boldsymbol{\alpha}_D), \end{aligned} \tag{7}$$

where $\mathcal{L}^{\texttt{reg}}_{\texttt{val}}$ and $\mathcal{L}^{\texttt{reg}}_{\texttt{tr}}$ are the registration losses (5) evaluated on the validation and training sets for a scenario, respectively. Subsequently, we fix the network architecture $\boldsymbol{\alpha}$ and hence resolve the bi-level optimization of the hyper-parameter $\boldsymbol{\lambda}$ and network parameters $\boldsymbol{\omega}$:

$$\min_{\boldsymbol{\lambda}} \ \mathcal{L}^{\texttt{seg}}_{\texttt{val}}(\boldsymbol{\lambda}, \boldsymbol{\omega}^*), \ s.t.\ \boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} \mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}, \boldsymbol{\lambda}), \tag{8}$$

where we evaluate the segmentatin loss (6) on the validation set for the scenario of interest.

Algorithm 1 summarizes the detailed optimization procedure where we adopt the gradient-based strategy for both bi-level optimization problems (7) and (8). We calculate the gradients of $\boldsymbol{\alpha}_F$, $\boldsymbol{\alpha}_D$ and $\boldsymbol{\lambda}$ via one-step first-order approximation [19], [24] for the steps 3-5, 10-12 and 17-19 in Alg. 1 [3], and the parameter ϵ is set to be a small scalar value equal to the learning rate.

Algorithm 1 Procedure for solving the model in Eq. (2)

Require: The search space, $\mathcal{D}_{\texttt{tr}}$ and $\mathcal{D}_{\texttt{val}}$.
Ensure: The searched architecture and hyper-parameter.

1: Initialize model weight $\boldsymbol{\omega}$.
   %% Stage 1: Search for $\boldsymbol{\alpha}_F$ with fixed $\boldsymbol{\lambda}$.
2: **while** not converged **do**
3: $\quad \boldsymbol{\omega}' = \boldsymbol{\omega} - \frac{\mathrm{d}\mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}, \boldsymbol{\alpha}_F)}{\mathrm{d}\boldsymbol{\omega}}$, $\boldsymbol{\omega}^{\pm} = \boldsymbol{\omega} \pm \epsilon \frac{\mathrm{d}\mathcal{L}^{\texttt{reg}}_{\texttt{val}}(\boldsymbol{\omega}', \boldsymbol{\alpha}_F)}{\mathrm{d}\boldsymbol{\omega}'}$
4: $\quad V_{\boldsymbol{\alpha}_F} = -\frac{1}{2\epsilon}\left(\frac{\partial \mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}^+, \boldsymbol{\alpha}_F)}{\partial \boldsymbol{\alpha}_F} - \frac{\partial \mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}^-, \boldsymbol{\alpha}_F)}{\partial \boldsymbol{\alpha}_F}\right)$
5: $\quad \boldsymbol{\alpha}_F \leftarrow \boldsymbol{\alpha}_F - \frac{\partial \mathcal{L}^{\texttt{reg}}_{\texttt{val}}(\boldsymbol{\omega}', \boldsymbol{\alpha}_F)}{\partial \boldsymbol{\alpha}_F} - V_{\boldsymbol{\alpha}_F}$
6: $\quad \boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \frac{\mathrm{d}\mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}, \boldsymbol{\alpha}_F)}{\mathrm{d}\boldsymbol{\omega}}$
7: **end while**
   %% Stage 2: Search for $\boldsymbol{\alpha}_D$ with fixed $\boldsymbol{\lambda}$.
8: Reinitialize model weight $\boldsymbol{\omega}$.
9: **while** not converged **do**
10: $\quad \boldsymbol{\omega}' = \boldsymbol{\omega} - \frac{\mathrm{d}\mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}, \boldsymbol{\alpha}_D)}{\mathrm{d}\boldsymbol{\omega}}$, $\boldsymbol{\omega}^{\pm} = \boldsymbol{\omega} \pm \epsilon \frac{\mathrm{d}\mathcal{L}^{\texttt{reg}}_{\texttt{val}}(\boldsymbol{\omega}', \boldsymbol{\alpha}_D)}{\mathrm{d}\boldsymbol{\omega}'}$
11: $\quad V_{\boldsymbol{\alpha}_D} = -\frac{1}{2\epsilon}\left(\frac{\partial \mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}^+, \boldsymbol{\alpha}_D)}{\partial \boldsymbol{\alpha}_D} - \frac{\partial \mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}^-, \boldsymbol{\alpha}_D)}{\partial \boldsymbol{\alpha}_D}\right)$
12: $\quad \boldsymbol{\alpha}_D \leftarrow \boldsymbol{\alpha}_D - \frac{\partial \mathcal{L}^{\texttt{reg}}_{\texttt{val}}(\boldsymbol{\omega}', \boldsymbol{\alpha}_D)}{\partial \boldsymbol{\alpha}_F} - V_{\boldsymbol{\alpha}_D}$
13: $\quad \boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \frac{\mathrm{d}\mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}, \boldsymbol{\alpha}_D)}{\mathrm{d}\boldsymbol{\omega}}$
14: **end while**
   %% Stage 3: Search for $\boldsymbol{\lambda}$.
15: Reinitialize model weight $\boldsymbol{\omega}$.
16: **while** not converged **do**
17: $\quad \boldsymbol{\omega}' = \boldsymbol{\omega} - \frac{\mathrm{d}\mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}, \boldsymbol{\lambda})}{\mathrm{d}\boldsymbol{\omega}}$, $\boldsymbol{\omega}^{\pm} = \boldsymbol{\omega} \pm \epsilon \frac{\mathrm{d}\mathcal{L}^{\texttt{seg}}_{\texttt{val}}(\boldsymbol{\omega}', \boldsymbol{\lambda})}{\mathrm{d}\boldsymbol{\omega}'}$
18: $\quad V_{\boldsymbol{\lambda}} = -\frac{1}{2\epsilon}\left(\frac{\partial \mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}^+, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} - \frac{\partial \mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}^-, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}}\right)$
19: $\quad \boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \frac{\partial \mathcal{L}^{\texttt{seg}}_{\texttt{val}}(\boldsymbol{\omega}', \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} - V_{\boldsymbol{\lambda}}$
20: $\quad \boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \frac{\mathrm{d}\mathcal{L}^{\texttt{reg}}_{\texttt{tr}}(\boldsymbol{\omega}, \boldsymbol{\lambda})}{\mathrm{d}\boldsymbol{\omega}}$
21: **end while**
22: **return** Architecture and hyper-parameter $\boldsymbol{\alpha}^*_F, \boldsymbol{\alpha}^*_D, \boldsymbol{\lambda}^*$.

IV. Experimental Results and Analysis

This section firstly provides the detailed experimental settings and then presents ablation studies that validate the effectiveness of individual modules of the proposed method. Finally, we compare with popular conventional and recent deep-based approaches on several data sets covering various scenarios

A. Training Details

Our automated registration contains two stages, *i.e.*, train-search and train-evaluation. In the first stage, we search for network operations and training objectives with respect to a validation dataset given a specific scenario. We initialize all the operation variables $\boldsymbol{\alpha}$ to zeros indicating equal possibilities over all operation candidates in the beginning. We use Adam [40] as the optimizer with the learning rates

[3] We omit the derivations of the gradients due to page limit, and our codes are available at https://github.com/Alison-brie/AutoReg.

1×10^{-4} and 4×10^{-3} for $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$, respectively. The searching procedure for $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ stops when they remain stable in 10 consecutive epochs. Subsequently, we update the network weights $\boldsymbol{\omega}$ given the searched $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ for 15 epochs, and then train all the parameters for additional 30 epochs. In the train-evaluation stage, we train the weights $\boldsymbol{\omega}$ of the registration network given in the train-search stage for 200 epochs with the batch size set to 1 on the training set of the scenario. The learning rate is set to 1×10^{-4} for the Adam optimizer. The network predicts a half-resolution deformation field and we up-sample it via linear interpolation, yielding the final full resolution deformation field for evaluation. All the experiments were performed in Pytorch on 3.20GHz Intel(R) i7-8700 CPU with 32GB RAM and a NVIDIA TITAN XP GPU.

B. Data Preparation and Evaluation Metrics

We applied AutoReg to a wide range of MRI registration scenarios from mono-modality (registering brain T1 images to an averaged T1 template, brain T1 to T1 images, brain T2 to T2 images, knee T1 to T1 images, and lung CT inspiration to expiration images) to multi-modality (registering brain T1 to T2 images).

Brain MR Image-to-Atlas registration. For image-to-atlas on T1 weighted brain MRI registration task, we use 528 scans from: ADNI [41], ABIDE [42], PPMI [43] and OASIS [3]. We adopt the atlas in [9]. We divide our data into 377, 21 and 130 volumes for training, validation and testing. Standard pre-processing operations, *e.g.,* motion correction, NU intensity correction, normalization, skull stripping, with FreeSurfer [44] and affine normalization with FSL [45] are conducted. We also segment the testing data with FreeSurfer, resulting in 29 anatomical structures in each volume. All images are cropped to size of $168 \times 192 \times 224$ with 1 mm isotropic resolution. For evaluation, all test MRI scans are anatomically segmented with Freesurfer to extract 30 anatomical structures.

Brain T1-to-T1 registration. For the image-to-image case, the target images are randomly selected from PPMI [43] dataset, resulting in 59 test pairs.

Knee T1-to-T1 registration. We employ knee MRIs from the Osteoarthritis Initiative [4] with corresponding segmentations of femur and tibia as well as femoral and tibial cartilage [46]. We divide images into 377, 21 and 130 volumes for training, validation and testing. All images are resampled to isotropic spacing of 1mm, in size of $160 \times 160 \times 160$.

Brain T2-to-T1 registration. For multi-modal registration, we utilize 135 cases from BraTS18 [5] and ISeg19 [6] datasets, and each case includes two image modalities: T1 and T2 brain images with the size of $160 \times 160 \times 160$. Among them, 10 cases have segmentation ground truth. The data set is split into 115, 10 and 10 for train, validation and test. As most T1 and T2 images are already aligned, we randomly choose one T1 scan as the atlas and try to register T2 scans to it.

Brain T2-to-T2 registration. For the T2 weighted image-to-image case, the target images are randomly selected from T2 brain images in the above multimodal datasets, resulting in 59 test pairs.

[4]https://nda.nih.gov/oai/

[5]https://www.med.upenn.edu/sbia/brats2018.html

[6]https://iseg2019.web.unc.edu/

TABLE I
QUANTITATIVE COMPARISONS IN TERMS OF RUNNING TIME ON BRAIN MR DATASETS OF SIZE $160 \times 192 \times 224$.

-	Training	Auto-search + Training
Runtime (hour)	23	71

Lung CT inspiration-expiration registration. We download CT lung images from learn2reg challenge [7], including 20 training and 10 test 3D volumes in size of $192\times192\times208$ with segmentation annotations. We take expiration data as fixed images and inspiration as moving images.

We adopt the average Dice score [39] across a representative set of structures over all testing pairs as the evaluation metric. To evaluate the smoothness of the deformation field, we compute the Jacobian matrix and count all the folds [36]. Furthermore, NCC, as an auxiliary metric, is employed to verify alignment performance. Both higher Dice scores and NCC values indicate more accurate registration, a lower variance reveals our method is more stable and robust.

C. Ablation Study

We first assess how the computational cost of our search compares to the standard approach. Then, we investigate how our auto-search works across diverse registration tasks. Furthermore, we demonstrate the effect that this strategy can have on existing registration networks.

1) Computational cost: As shown in Table. I, we first list the full training and optimization time for achieving the final models. It takes about 3 days for the proposed method to get a scenario-oriented registration model. Traditional training baseline typically involves training many separate models with various hyper-parameter configurations (architecture designs and hyper-parameters in loss function). And its computational cost mainly depends on the number of combinations of hyper-parameter configurations, while the number is usually set much larger than 10, which is computationally prohibitive. For models with many hyper-parameters, more than one hyper, the value would be even more significant. Whereas the proposed automated searching strategy could drastically alleviate computation burdens, highly efficient than manual search by orders of magnitude.

2) Automatic learning across registration tasks: To explore the influence of model weights, architectures, and training objectives in registration performance and verify the benefit from adaptive feature extraction cell, task-aware deformation estimation cell design, and scenario-oriented training objectives, we made groups of contrast experiments, covering diverse registration scenarios. Specifically, we first train a registration model on PPMI brain T1 MR image-to-atlas registration tasks, then take the model as initialization to adopt our auto-search strategy to register images of other different scenarios. The scenarios cover registration on another ADNI dataset, Brain

[7]http://doi.org/10.5281/zenodo.3835682

TABLE II
ABLATION ANALYSIS OF AUTOMATIC LEARNING ACROSS DIFFERENT KINDS OF REGISTRATION SCENARIOS IN TERMS OF DICE SCORE AND NCC. THE BEST RESULT IS IN RED WHEREAS THE SECOND BEST ONE IS IN BLUE.

$\boldsymbol{\omega}$ $\boldsymbol{\alpha}_F$ $\boldsymbol{\alpha}_D$ $\boldsymbol{\lambda}$		× × × ×	✓ × × ×	✓ ✓ × ×	✓ × ✓ ×	✓ × × ✓	✓ ✓ ✓ ✓
ADNI	Dice	0.745 ± 0.027	0.754 ± 0.024	0.763 ± 0.020	0.758 ± 0.024	0.757 ± 0.021	0.774 ± 0.022
	NCC	0.217 ± 0.007	0.221 ± 0.007	0.234 ± 0.006	0.225 ± 0.006	0.229 ± 0.006	0.258 ± 0.005
Brain T1-to-T1	Dice	0.706 ± 0.038	0.750 ± 0.022	0.764 ± 0.024	0.761 ± 0.022	0.750 ± 0.027	0.778 ± 0.023
	NCC	0.186 ± 0.011	0.233 ± 0.009	0.243 ± 0.011	0.233 ± 0.011	0.251 ± 0.011	0.258 ± 0.010
Brain T2-to-T2	Dice	0.438 ± 0.110	0.636 ± 0.008	0.646 ± 0.010	0.640 ± 0.010	0.643 ± 0.010	0.646 ± 0.010
	NCC	0.033 ± 0.002	0.132 ± 0.002	0.138 ± 0.002	0.132 ± 0.002	0.138 ± 0.002	0.143 ± 0.002
Knee T1-to-T1	Dice	0.393± 0.107	0.588 ± 0.157	0.616 ± 0.150	0.614 ± 0.119	0.606 ± 0.154	0.616 ± 0.150
	NCC	0.140 ± 0.012	0.234 ± 0.032	0.202 ± 0.023	0.242 ± 0.030	0.239 ± 0.030	0.267 ± 0.023
Brain T2-to-T1	Dice	0.407 ± 0.005	0.599 ± 0.007	0.599 ± 0.006	0.598 ± 0.008	0.621 ± 0.004	0.622 ± 0.007
	NCC	0.033 ± 0.001	0.107 ± 0.002	0.108 ± 0.002	0.107 ± 0.002	0.127 ± 0.002	0.108 ± 0.001

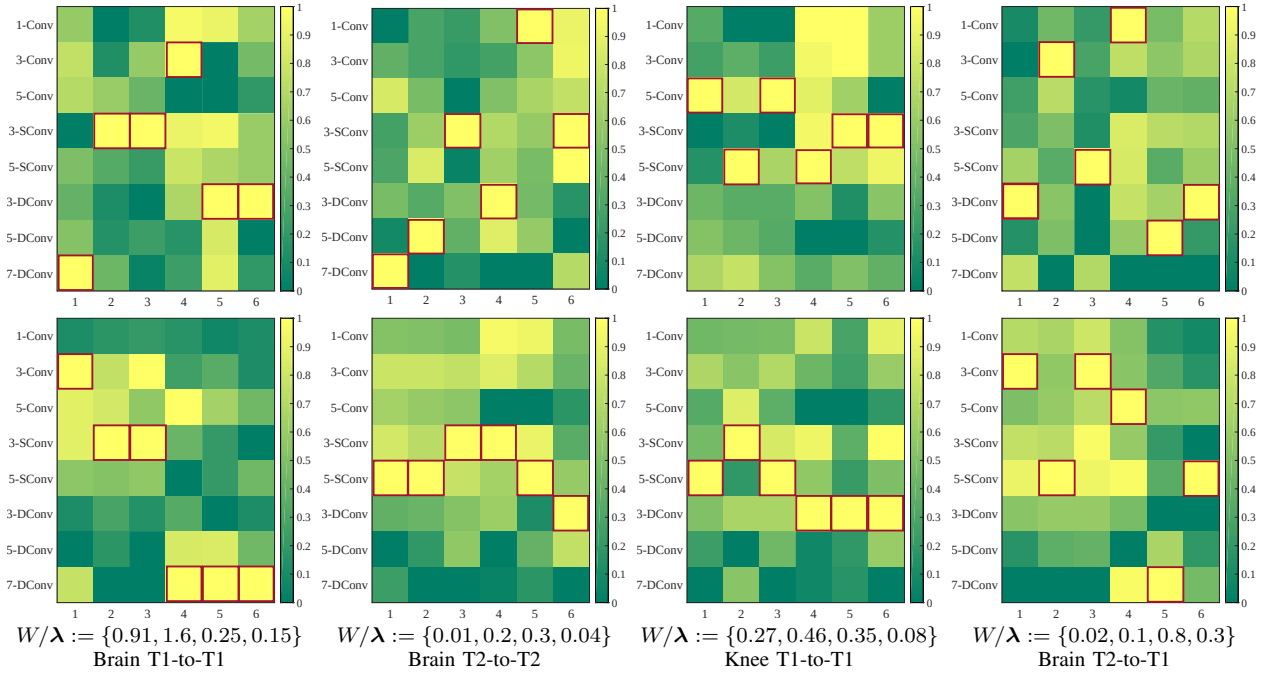

Fig. 3. Heatmaps of candidate architectures $\boldsymbol{\alpha} := \{\boldsymbol{\alpha}_F, \boldsymbol{\alpha}_D\}$ in the last searching epoch. In each sub-figure, horizontal axis represents six convolution operations of two cells to be searched, and vertical axis represents eight operation types, red boxes indicate searched architectures with the highest score. Top: the searched $\boldsymbol{\alpha}_F$. Bottom: corresponding searched $\boldsymbol{\alpha}_D$.

T1-to-T1, Brain T2-to-T2, Knee T1-to-T1 and Brain T2-to-T1 setups.

In Table. II, we directly apply the model on the PPMI dataset to other scenarios, corresponding to the 1st column result, showing poor performance. Whereas performances of re-trained models (with tailored model weight $\boldsymbol{\omega}$) on these tasks correspond to the 2nd column. We also demonstrate the performance of searched networks with auto-learned feature cells $\boldsymbol{\alpha}_F$, deformation cells $\boldsymbol{\omega}_D$, and training objectives $\boldsymbol{\lambda}$, corresponding to the 3rd to 5th columns. To fully capture the benefit of the proposed technique, we further report the increase in registration accuracy for cases where all hyper-parameters are searched in the last column. We can observe that, *firstly*, retraining the model for different alignment tasks will result in better performance. *Secondly*, searched tailored architecture and training objectives largely improve numerical results, which means automatic learning combining training objectives, architectures and hyperparameters can achieves excellent alignment performance in different alignment scenarios.

Also, the models on the diagonal with a blue background perform second best which can be seen from Table. II, providing meaningful indications and conclusions. *Firstly*, when transferring to another dataset or image contrast, feature extraction plays a dominant role in model performance. *Secondly*, whereas transferring to another anatomical structure such as the knee data, regulating the deformation estimation section has a more significant impact on the performance of the model. *Lastly*, adjusting the training objective plays a more important role in the performance of a registration network when transferring to multi-modal datasets.

3) Searched architectures and hyperparameters: From the searched architecture in Fig. 3, *firstly*, we observe that in the case of the same convolution kernel size, our searched cells

TABLE III
COMPARISON RESULTS IN TERMS OF DICE SCORES OF DIFFERENT NETWORK ARCHITECTURES ON MULTIPLE REGISTRATION TASKS.

Method	Brain T1-to-T1	Brain T2-to-T2	Knee T1-to-T1	Brain T2-to-T1
Hands-S	0.700 ± 0.036	0.610 ± 0.009	0.395 ± 0.110	0.579 ± 0.005
Hands-M	0.769 ± 0.025	0.636 ± 0.010	0.605 ± 0.131	0.617 ± 0.006
Hands-L	0.761 ± 0.025	0.610 ± 0.009	0.614 ± 0.091	0.613 ± 0.007
AutoReg	0.778 ± 0.023	0.646 ± 0.010	0.616 ± 0.150	0.622 ± 0.007

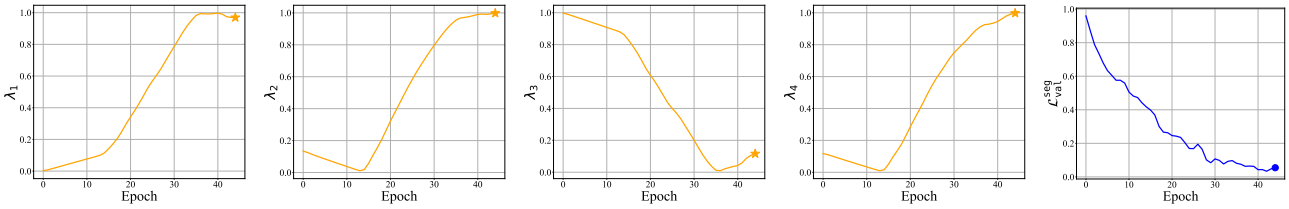

Fig. 4. Hyperparameters values and loss values of hyperparameters optimization across training epoch on the Image-to-Atlas registration of brain MRI. We rescale these values to [0,1]. As the epoch number increases, the change of values becomes smoother.

TABLE IV
ABLATION EXPERIMENTS ON OUR SCENARIO-ORIENTED OBJECTIVE DESIGN.

Application	W/ NCC	W/ MIND	W/ UNIFIED	W/ UNIFIED*
Image-to-Atlas	0.757 ± 0.014	0.737 ± 0.019	0.758 ± 0.012	0.764 ± 0.011
Brain T2-to-T1	0.591 ± 0.004	0.601 ± 0.008	0.587 ± 0.006	0.604 ± 0.005

contain more SConv and DConv than ordinary convolution. Overall, 3-SConv has been selected the most times while 1-Conv and 3-Conv are selected the least. *Secondly*, we find that the first convolution of each feature-cell and the last convolution of each deformation-cell favor the convolution type with large receptive fields, while these correspond to the high-resolution positions in the registration network. We may also draw interesting conclusions: compared to the position at smaller resolutions, the ones on larger resolution prefer larger receptive fields such as high dilation rates and large kernels. *Thirdly*, we can observe that 1-Conv only appears in the searched feature cells. It is reasonable because the number of feature maps often increases with the depth of the network whereas the 1-Conv can be used to offer a channel-wise pooling, often called feature map pooling or a projection layer. This simple technique can be used for dimensionality reduction, decreasing the number of feature maps whilst retaining their salient features, being a good choice for feature extraction. These observations provide us with meaningful and insightful cues for manually designing registration networks. Fig. 3 also gives the searched hyper-parameters under these registration scenarios. Obviously, we can observe that the optimal hyper-parameter varies substantially across registration tasks. For example, uni-modal tasks such as brain T1 MR image-to-image require a significantly different hyper-parameters value than multi-modal data.

4) Optimality verification: To verify the searched architectural design is optimal for different deformable registration tasks, we report the registration performance of the searched architecture compared to diverse architectural designs of existing deformable registration methods in Table. III. To be specific, we explore how much of a difference the choice of particular architecture actually makes by comparing experimental results among the networks with all-1-Conv, all-3-Conv, all-7-Conv, and searched architecture. Generally speaking, convolution with a bigger kernel has a larger receptive field and feature expression ability. The artificially designed all-7-Conv can achieve better performance than all-1-Conv and all-3-Conv in most scenarios. However, the all-3-Conv design is more suitable for the knee T1-to-T1 scenario, rather than all-7-Conv. In contrast, our searched ones beat all these hand-designed architectures and obtain suitable convolution types with appropriate feature expression capabilities for different scenarios.

Fig. 4 gives the variability of hyperparameters values and loss values of hyperparameters optimization across training epoch. We normalize the hyperparameters by adding regularization to the magnitude of the hyperparameters. As the figure shows, with the convergence of loss values, the change of hyperparameters values gradually stabilized. The yellow stars and blue dot indicate the optimal value as identified by automatic hyperparameter optimization. As shown in Tab. IV, we compare the results of experiments using NCC loss, MIND loss, the unified loss with manually selected hyperparameters, and the unified loss using searched hyperparameters. Results indicate that the networks trained with the searched loss functions deliver accuracy on par or even superior to those with the handcrafted losses. When switching to multi-modal data, generally, manually loss function tuning will be executed, which requires many training runs. The proposed self-tuned training could auto-adapt to this new scene and achieve satisfying performance.

5) Generalizability analysis: To further demonstrate the importance of applying auto-tuning methods across model types and not relying on previously-published hyperparameters for different applications and network designs, we involve the application of our AutoReg technique to the voxelmorph, a UNet-like registration model. As quantitative and qualitative

TABLE V
ABLATION ANALYSIS OF AUTOREG STRATEGY FOR VM MODELS ON MULTIPLE REGISTRATION TASKS.

Method	Brain T1-to-T1	Brain T2-to-T2	Knee T1-to-T1	Brain T2-to-T1
VM	0.757 ± 0.035	0.638 ± 0.012	0.440 ± 0.132	0.579 ± 0.013
VM + AutoReg	0.761 ± 0.010	0.640 ± 0.013	0.482 ± 0.151	0.596 ± 0.006

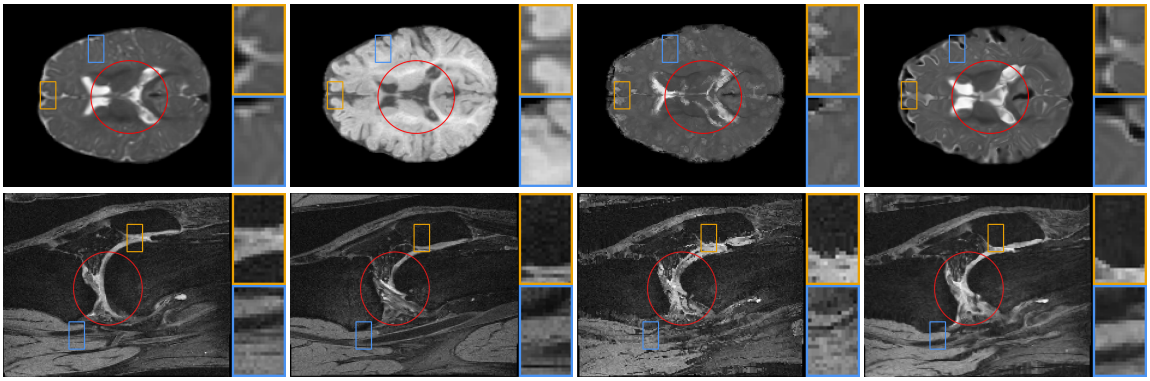

Fig. 5. Registration accuracy compared to different registration methods on diverse registration tasks in terms of Dice score. Each box shows mean accuracy over all anatomical structures for all test-image pairs.

TABLE VI
COMPARISON RESULTS OF IMAGE-TO-ATLAS REGISTRATION IN TERMS OF DICE SCORES OF DIFFERENT METHODS ON MULTIPLE DATASETS.

Method	ABIDE	ADNI	HCP	OASIS	PPMI
Initial	0.624 ± 0.024	0.571 ± 0.049	0.666 ± 0.027	0.580 ± 0.028	0.610 ± 0.033
ANTs	0.728 ± 0.029	0.761 ± 0.021	0.767 ± 0.016	0.765 ± 0.010	0.778 ± 0.013
NiftyReg	0.747 ± 0.026	0.737 ± 0.035	0.768 ± 0.013	0.748 ± 0.017	0.765 ± 0.015
VM	0.754 ± 0.016	0.761 ± 0.024	0.768 ± 0.013	0.765 ± 0.010	0.775 ± 0.013
VM-diff	0.773 ± 0.009	0.768 ± 0.020	0.413 ± 0.111	0.757 ± 0.011	0.781 ± 0.011
MultiPropReg	0.764 ± 0.016	0.773 ± 0.017	0.776 ± 0.010	0.777 ± 0.006	0.787 ± 0.010
Our AutoReg	0.784 ± 0.008	0.774 ± 0.022	0.777 ± 0.013	0.788 ± 0.010	0.788 ± 0.012

TABLE VII
COMPARISON RESULTS IN TERMS OF REGULARITY OF THE TRANSFORMATION OF DIFFERENT METHODS ON MULTI BRAIN MR DATASETS.

Method	ABIDE	ADNI	HCP	OASIS	PPMI
ANTs	27288 ± 3411	30737 ± 9537	28379 ± 9537	29094 ± 8772	25452 ± 6490
NiftyReg	11.4 ± 13	572.2 ± 878	9576 ± 2287	416 ± 416	314.3 ± 353
VM	28861 ± 1616	33047 ± 4667	30716 ± 2086	32029 ± 3498	30192 ± 3375
VM-diff	25 ± 13.1	43 ± 33.1	3945 ± 3854	35 ± 13	29 ± 24
MultiPropReg	1 ± 0.9	5.3 ± 6	0	6.2 ± 4.6	0.1 ± 0.7
AutoReg	0	5 ± 6	0	0	0.1 ± 0.7

ablation comparisons in Tab. V and Fig. 5, our AutoReg startegy could effectively facilitate the tuning of existing highly-parameterized registration models.

D. Comparison Experiments

1) Baseline methods: We compare our approach with five state-of-the-art registration methods, including two optimization-based tools: Symmetric Normalization (SyN) [6] and NiftyReg [8], three learning-based methods: VoxelMorph [9], diffeomorphic variant [11] (referred as VM and VM-diff, respectively) and MultiPropReg [19]. We train deep methods with recommended hyper-parameters on the same datasets from scratch. The parameter settings of the conventional methods are as follows. For SyN, we use the version implemented in ANTs [47] and obtain hyper-parameters from [9], which uses a wide parameter sweep across datasets same to ours. We take cross-correlation as the measurement metric and use a step size of 0.25, Gaussian parameters (9, 0.2), at three scales with 201 iterations each. As for NiftyReg, we use the Cross Correlation as the similarity measure. We run it with 12 threads through 1500 iterations.

TABLE VIII
QUANTITATIVE COMPARISONS IN TERMS OF RUNNING TIME AND MODEL SIZE ON BRAIN MR DATASETS OF SIZE $160 \times 192 \times 224$.

Methods	Runtime (s)	# (MB)
Elastix	83 ± 10	N/A
ANTs	4614 ± 1030	N/A
NiftyReg	435 ± 39	N/A
VM	0.558 ± 0.017	1.146
VM-diff	0.423 ± 0.011	1.016
MultiPropReg	0.360 ± 0.010	0.526
AutoReg	0.270 ± 0.010	0.852

2) Comparison results: First, on the registration task of brain MR image-to-atlas, we quantitatively evaluate the accuracy of all these techniques in terms of running time and

TABLE IX
COMPARISON RESULTS IN TERMS OF DICE SCORES OF DIFFERENT METHODS ON MULTIPLE REGISTRATION TASKS.

Method	Brain T1-to-T1	Brain T2-to-T2	Knee T1-to-T1	Brain T2-to-T1	Lung CT-to-CT
Initial	0.613 ± 0.057	0.563 ± 0.018	0.340 ± 0.077	0.539 ± 0.007	0.863 ± 0.035
ANTs	0.777 ± 0.030	0.625 ± 0.013	0.498 ± 0.210	0.538 ± 0.019	0.926 ± 0.049
NiftyReg	0.773 ± 0.026	0.645 ± 0.006	0.340 ± 0.078	0.639 ± 0.011	0.940 ± 0.039
VM	0.757 ± 0.035	0.638 ± 0.012	0.440 ± 0.132	0.579 ± 0.013	0.927 ± 0.034
VM-diff	0.765 ± 0.023	0.412 ± 0.009	0.320 ± 0.007	0.334 ± 0.006	-
MultiPropReg	0.775 ± 0.027	0.610 ± 0.012	0.578 ± 0.136	0.644 ± 0.007	0.931 ± 0.014
AutoReg	0.778 ± 0.023	0.646 ± 0.010	0.616 ± 0.150	0.622 ± 0.007	0.937 ± 0.014

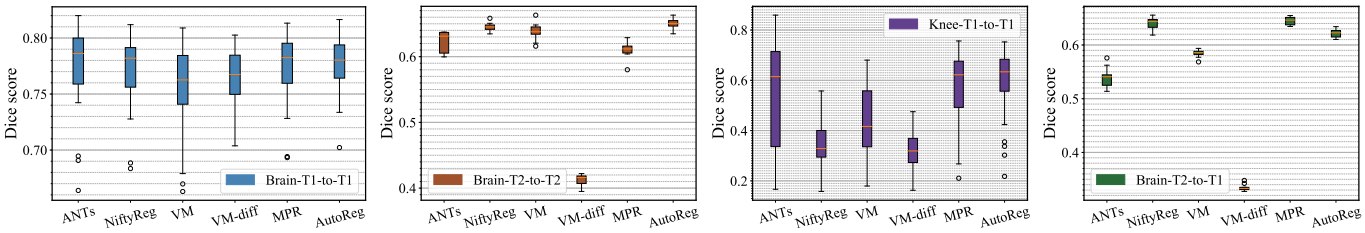

Fig. 6. Registration accuracy compared to different registration methods on diverse registration tasks in terms of Dice score. Each box shows mean accuracy over all anatomical structures for all test-image pairs.

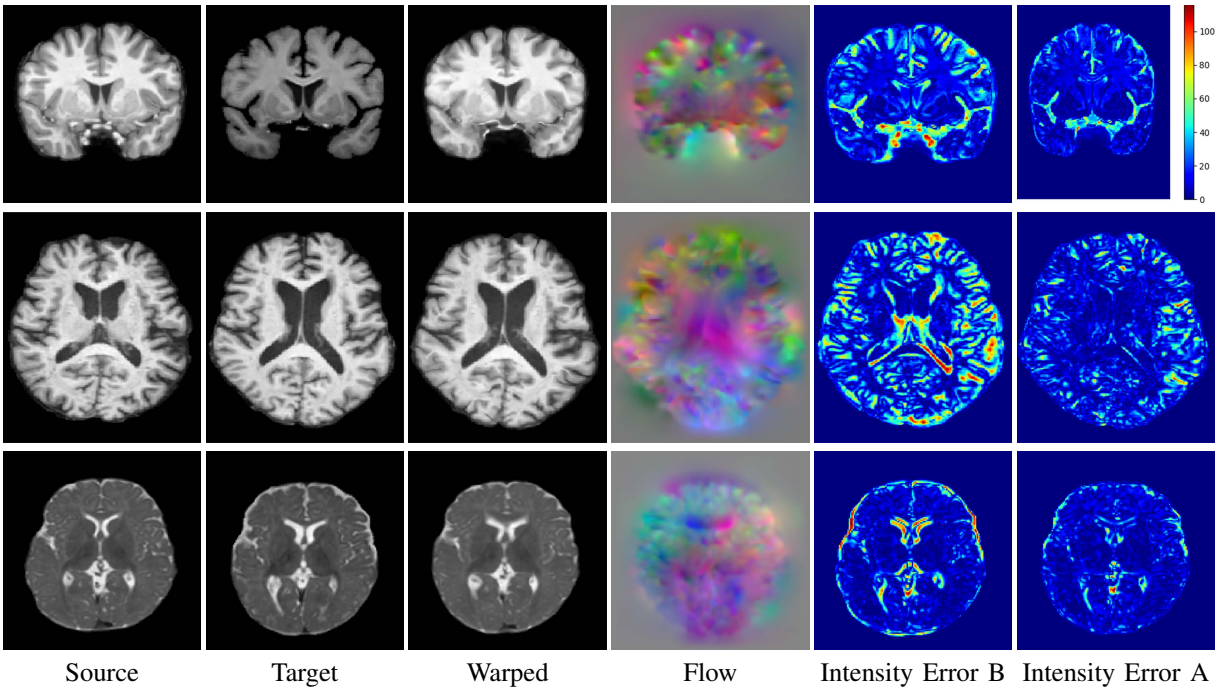

Fig. 7. Sample result of registering different images. Each row refers to an example registration case. Example 2D slices of Intensity difference Before registration and Intensity difference After registration. The registration field is visualized by RGB images with each channel representing dimension.

Dice score.

In Table. VI and Table. VII, only our method gives an obvious higher mean and lower variance of Dice score while decreasing the fold number to nearly zero, showing consistent satisfying performance on various datasets. In contrast to other registration algorithms, each registration method has a fixed appropriate network structure with training objectives for its specific alignment application, but our method can adaptively designs the optimal network structure and training objectives for different application scenarios. On the other hand, Table. VIII presents the runtime and model size among different methods whereas ours needs less inference time and fewer computational parameters.

Table. IX demonstrates performance in terms of Dice score on challenging registration tasks, including brain T1 MR image-to-image, brain T2 MR image-to-image, multi-modal, and knee T1 MR image-to-image registrations. While our method gives an obvious lower variance with a comparable Dice for all of these cases. Fig. 6 depicts the stability of the methods in view of the box-plot of Dice score, where fewer outliers and lower variance indicate a more stable registration. Note that, the optimization-based methods perform slightly better for the brain registration where the image pairs are much more similar to each other. However, they are less satisfying in

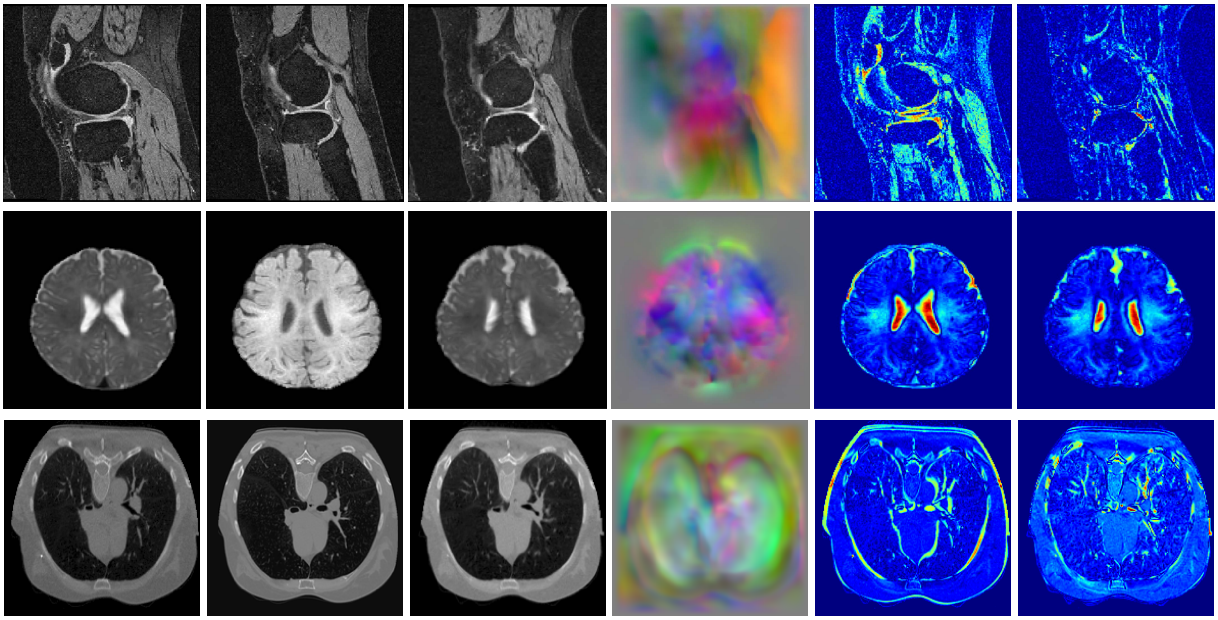

Fig. 8. Sample result of registering different images. Each row refers to an example registration case. Example 2D slices of Intensity difference Before registration and Intensity difference After registration. The registration field is visualized by RGB images with each channel representing dimension.

the knee T1-to-T1 registration. When faced with such various challenging cases, learning-based models struggle to provide comprehensive solutions. While our method gives an obvious lower variance with a comparable mean of Dice for all these cases, showing stronger stability.

Explanations of unsatisfying performance of VM-diff [11]. Diffeomorphic voxelmorph fails entirely on three of the evaluation scenarios. It is surprising given the reasonable performance of the non-diffeomorphic models in those applications. However, we think the reason is that VM-diff set the Mean Square Error as the default similarity metric which is difficult and incapable to cover challenging registration scenarios. The failure of such magnitude may be caused by weak loss function design, rather than diffeomorphic implementations.

Our representative registration results are given in Fig. 7 and Fig. 8. The first three registration cases in Fig. 7 contain image-to-atlas on T1 brain MR, image-to-image on T1 brain MR and T2 brain MR test pairs. The large deformations in scans make registration challenging and difficult. As a result, all the source images are well aligned to the target. The second three rows in Fig. 8 contain knee T1 MR data, multi-modal data, and lung CT inspiration-expiration images. Although large deformations and intensity differences exist in scans, source images are well aligned to the target, demonstrating our outstanding performance.

V. Discussions and Conclusions

We come with an automatic learning framework, dubbed as AutoReg, for medical deformable image registration. AutoReg removes the need for computer experts to well-design both architectures and training objectives for the specific type of medical data, also drastically alleviating computation burdens. We offer a compelling path toward an automated learning framework for medical image analysis. To our best knowledge, this is the first work to achieve AutoML for medical image registration. Specifically, we construct a triple-level framework to jointly optimize learns the weights, architecture, and loss function of a deep network for DIR, so that we can automate the process of designing three major components: feature extraction, deformation model and objective function, given specific medical scenario. Extensive experiments on different image contrasts, anatomical structures, and image modalities have shown that our method may automatically learn an optimal registration network for given volumes and achieve state-of-the-art performance. The auto-learned registration also runs extremely fast inheriting from common learning-based algorithms.

Potential downsides of this method. The AutoReg depends on labeled or annotated validation datasets, such as segmentation maps or landmarks, to perform optimization of hyperparameters in loss functions. On the other hand, the proposed NAS strategy involves hand-crafted designs and has limited degrees of freedom in the searching space. Some of the important elements in neural architecture design such as the presence of skip connection, the depth of the network, and element-wise multiplication and addition are not included in the searching space.

Future work. We would extend this method to cover other architectural hyperparameters, like the network topology-level search space that controls the connections among cells, number of layers, resolution levels, and even total network capacity, attempting a more generalized framework for facilitating tuning of highly-parameterized registration models.

Acknowledgment

This work was partially supported by the National Key R&D Program of China (2020YFB1313503), the National Natural

Science Foundation of China (Nos. 61922019 and 61672125), LiaoNing Revitalization Talents Program (XLYC1807088), and the Fundamental Research Funds for the Central Universities. We thank Dr. Adrian V. Dalca at the Massachusetts Institute of Technology for his contributions to the open-source community for medical image registration.

REFERENCES

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

[2] B. Jie, M. Liu, D. Zhang, and D. Shen, "Sub-network kernels for measuring similarity of brain connectivity networks in disease diagnosis," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2340–2353, 2018.

[3] D. S. Marcus, A. F. Fotenos, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies: Longitudinal MRI data in nondemented and demented older adults," *Journal of Cognitive Neuroscience*, vol. 22, no. 12, pp. 2677–2684, 2010.

[4] A. V. Dalca, M. Rakic, J. V. Guttag, and M. R. Sabuncu, "Learning conditional deformable templates with convolutional networks," in *Conference on Neural Information Processing Systems*, 2019, pp. 804–816.

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

[6] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, "Symmetric diffeomorphic image registration with cross-correlation: evaluating automated eling of elderly and neurodegenerative brain," *Medical Image Analysis*, vol. 12, no. 1, pp. 26–41, 2008.

[7] A. Chaudhury, "Multilevel optimization for registration of deformable point clouds," *IEEE Transactions on Image Processing*, vol. 29, pp. 8735–8746, 2020.

[8] W. Sun, W. J. Niessen, and S. Klein, "Free-form deformation using lower-order b-spline for nonrigid image registration," in *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2014, pp. 194–201.

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

[10] S. Zhao, Y. Dong, E. I. Chang, and Y. Xu, "Recursive cascaded networks for unsupervised medical image registration," in *IEEE International Conference on Computer Vision*, 2019, pp. 10 599–10 609.

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

[12] J. Wang and M. Zhang, "Deepflash: An efficient network for learning-based medical image registration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4443–4451.

[13] R. Liu, Z. Li, Y. Zhang, X. Fan, and Z. Luo, "Bi-level probabilistic feature learning for deformable image registration," in *International Joint Conference on Artificial Intelligence*, 2020, pp. 723–730.

[14] L. Pan, F. Shi, D. Xiang, K. Yu, L. Duan, J. Zheng, and X. Chen, "Octrexpert: a feature-based 3d registration method for retinal oct images," *IEEE Transactions on Image Processing*, vol. 29, pp. 3885–3897, 2020.

[15] J. Zhang, Y. Wang, J. Dai, M. Cavichini, D.-U. G. Bartsch, W. R. Freeman, T. Q. Nguyen, and C. An, "Two-step registration on multi-modal retinal images via deep neural networks," *IEEE Transactions on Image Processing*, vol. 31, pp. 823–838, 2021.

[16] J. Öfverstedt, J. Lindblad, and N. Sladoje, "Fast and robust symmetric image registration based on distances combining intensity and spatial information," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3584–3597, 2019.

[17] X. Cao, J. Yang, Y. Gao, Q. Wang, and D. Shen, "Region-adaptive deformable registration of ct/mri pelvic images via learning-based image synthesis," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3500–3512, 2018.

[18] A. Hoopes, M. Hoffmann, B. Fischl, J. Guttag, and A. V. Dalca, "Hypermorph: Amortized hyperparameter learning for image registration," in *International Conference on Information Processing in Medical Imaging*. Springer, 2021, pp. 3–17.

[19] R. Liu, Z. Li, X. Fan, C. Zhao, H. Huang, and Z. Luo, "Learning deformable image registration from optimization: Perspective, modules, bilevel training and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[20] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Conference on Neural Information Processing Systems*, 2015, pp. 2017–2025.

[21] M. Niethammer, R. Kwitt, and F.-X. Vialard, "Metric learning for image registration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8463–8472.

[22] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710.

[23] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *AAAI Conference on Artificial Intelligence*, 2019, pp. 4780–4789.

[24] H. Liu, K. Simonyan, and Y. Yang, "DARTS: differentiable architecture search," in *International Conference on Learning Representations*, 2019.

[25] X. Dong and Y. Yang, "Searching for a robust neural architecture in four GPU hours," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1761–1770.

[26] J. Liu, S. Zhou, Y. Wu, K. Chen, W. Ouyang, and D. Xu, "Block proposal neural architecture search," *IEEE Transactions on Image Processing*, vol. 30, pp. 15–25, 2020.

[27] G. Ghiasi, T. Lin, and Q. V. Le, "NAS-FPN: learning scalable feature pyramid architecture for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7036–7045.

[28] Y. Peng, L. Bi, M. J. Fulham, D. Feng, and J. Kim, "Multi-modality information fusion for radiomics-based neural architecture search," in *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2020, pp. 763–771.

[29] R. Liu, L. Ma, J. Zhang, X. Fan, and Z. Luo, "Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 561–10 570.

[30] C. Liu, Y. Tian, Z. Chen, J. Jiao, and Q. Ye, "Adaptive linear span network for object skeleton detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 5096–5108, 2021.

[31] X. Yang, S. Wang, J. Dong, J. Dong, M. Wang, and T.-S. Chua, "Video moment retrieval with cross-modal neural architecture search," *IEEE Transactions on Image Processing*, vol. 31, pp. 1204–1216, 2022.

[32] Y. Tang, B. Li, M. Liu, B. Chen, Y. Wang, and W. Ouyang, "Autopedestrian: an automatic data augmentation and loss function search scheme for pedestrian detection," *IEEE transactions on image processing*, vol. 30, pp. 8483–8496, 2021.

[33] Q. Yu, D. Yang, H. Roth, Y. Bai, Y. Zhang, A. L. Yuille, and D. Xu, "C2FNAS: coarse-to-fine neural architecture search for 3d medical image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4125–4134.

[34] D. Guo, D. Jin, Z. Zhu, T. Ho, A. P. Harrison, C. Chao, J. Xiao, and L. Lu, "Organ at risk segmentation for head and neck cancer using stratified learning and neural architecture search," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4222–4231.

[35] Y. He, D. Yang, H. Roth, C. Zhao, and D. Xu, "Dints: Differentiable neural network topology search for 3d medical image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5841–5850.

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

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

[38] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.

[39] L. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[40] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Conference on Neural Information Processing Systems*, 2019, pp. 8024–8035.

[41] S. G. Mueller *et al.*, "Ways toward an early diagnosis in Alzheimer's disease: The Alzheimer's disease neuroimaging initiative (ADNI)," *Alzheimer's Dementia*, vol. 1, no. 1, pp. 55–66, 2005.

[42] A. Di Martino *et al.*, "The Autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in Autism." *Molecular psychiatry*, vol. 19, no. 6, pp. 659–667, 2014.
[43] K. Marek *et al.*, "The Parkinson progression marker initiative (PPMI)," *Progress in Neurobiology*, vol. 95, no. 4, pp. 629–635, 2011.
[44] B. Fischl, "Freesurfer," *NeuroImage*, vol. 62, no. 2, pp. 774–781, 2012.
[45] M. W. Woolrich *et al.*, "Bayesian analysis of neuroimaging data in FSL," *NeuroImage*, vol. 45, no. 1, pp. S173–S186, 2009.
[46] F. Ambellan, A. Tack, M. Ehlke, and S. Zachow, "Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the osteoarthritis initiative," *Medical Image Analysis*, vol. 52, pp. 109–118, 2019.
[47] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, "A reproducible evaluation of ants similarity metric performance in brain image registration," *NeuroImage*, vol. 54, no. 3, pp. 2033–2044, 2011.