Cross-Resolution Distillation for Efficient 3D Medical Image Registration

Bo Hu, Shenglong Zhou, Zhiwei Xiong, *Member, IEEE*, and Feng Wu, *Fellow, IEEE*

***Abstract*— Images captured in clinic such as MRI scans are usually in 3D formats with high spatial resolutions. Existing learning-based models for medical image registration consume large GPU memories and long inference time, which is difficult to be deployed in resource-limited diagnosis scenarios. To address this problem, instead of shrinking the model size as in previous works, we turn to reducing the input resolution of existing registration models and boosting their performance through knowledge distillation. Specifically, we propose a cross-resolution distillation (CRD) scheme, which is designed to train low-resolution models under the guidance of corresponding high-resolution models. Nevertheless, due to the resolution gap between features in high/low-resolution models, straightforward distillation is difficult to apply. To overcome this challenge, we first introduce a feature-shifted teacher (FST) to shift and fuse features of high/low-resolution models. Then, we exploit this teacher model to guide the learning of the low-resolution student model with distillation losses on both features and deformation fields. Finally, we only need to use the distilled student model during inference. Experimental results on four 3D medical image datasets demonstrate that the low-resolution models trained through our CRD scheme use fewer than 20% GPU memories and less than 20% inference time while achieving competitive performance compared with corresponding high-resolution models.**

***Index Terms*— Medical image registration, knowledge distillation, deep learning.**

I. Introduction

IMAGE registration aims to build correspondence (termed deformation field) between an image pair, so that one image (termed moving image) can be aligned with the other image (termed fixed image) [1], [2]. Image registration is a fundamental task in medical image analysis such as multi-modal fusion [3], image-guided surgery [4] and image segmentation [5], [6]. Owing to medical image registration, images captured from different devices, subjects and times can be compared and analyzed [7], which is crucial in theoretical research and clinical trials.

Traditional methods [8], [9] regard the registration procedure as an optimization problem for each image pair. The optimization is iterative and therefore extremely slow in practice. Recently, deep learning-based methods [10]–[13] regard the registration procedure as a mapping from an image pair to a deformation field and provide advanced registration accuracy. Since manual annotation for supervised methods are difficult to obtain, unsupervised methods attract extensive research attention. VoxelMorph [14] and its diffeomorphic version [11] are representative unsupervised models and have been widely used in clinical applications.

Although learning-based methods provide advanced registration performance, the ensuing heavy computational burdens become a critical problem. Computational burdens are mainly determined by the input resolution and the model size. In terms of the input resolution, images captured in clinic such as MRI scans are usually in 3D formats, which means the computational complexity increases from the square to the cubic magnitude compared with 2D images. Meanwhile, 3D MRI images used for precise diagnosis usually have a high spatial resolution. A higher input resolution brings in heavier computational burdens. In terms of the model size, in order to obtain promising performance, existing state-of-the-art registration methods [15], [16] have a large model size. A larger model size means heavier computational burdens.

Current learning-based methods deploy their models on graphic processing units (GPUs) to handle computation in parallel. However, GPU devices with large memories are expensive and high energy-consuming. Furthermore, heavy computational burdens result in long inference time of models, which is unfriendly for fast diagnosis. Therefore, it is vital to lessen the computational burdens to achieve efficient 3D medical image registration, especially in resource-limited scenarios. Existing efficient registration methods focus on shrinking the model size to reduce computational burdens. For example, VoxelMorph gives a light version with fewer convolutional layers and channels compared with the raw version. Tran *et al.* [17] propose a light-weight student model and distill knowledge from cumbersome teachers to it with adversarial learning. Nevertheless, these solutions still have two drawbacks: first, the decrement of computation only has a linear relationship with the decrement of the model size, which leads to limited savings of GPU memories; more importantly,

Manuscript received 15 February 2022; revised 26 April 2022; accepted 10 May 2022. Date of publication 26 May 2022; date of current version 4 October 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700800, in part by the National Natural Science Foundation of China under Grant 62021001, and in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2019-025. This article was recommended by Associate Editor D. Gragnaniello. *(Bo Hu and Shenglong Zhou are co-first authors.) (Corresponding author: Zhiwei Xiong.)*

The authors are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230026, China, and also with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China (e-mail: hubosist@mail.ustc.edu.cn; slzhou96@mail.ustc.edu.cn; zwxiong@ustc.edu.cn; fengwu@ustc.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3178178.

Digital Object Identifier 10.1109/TCSVT.2022.3178178

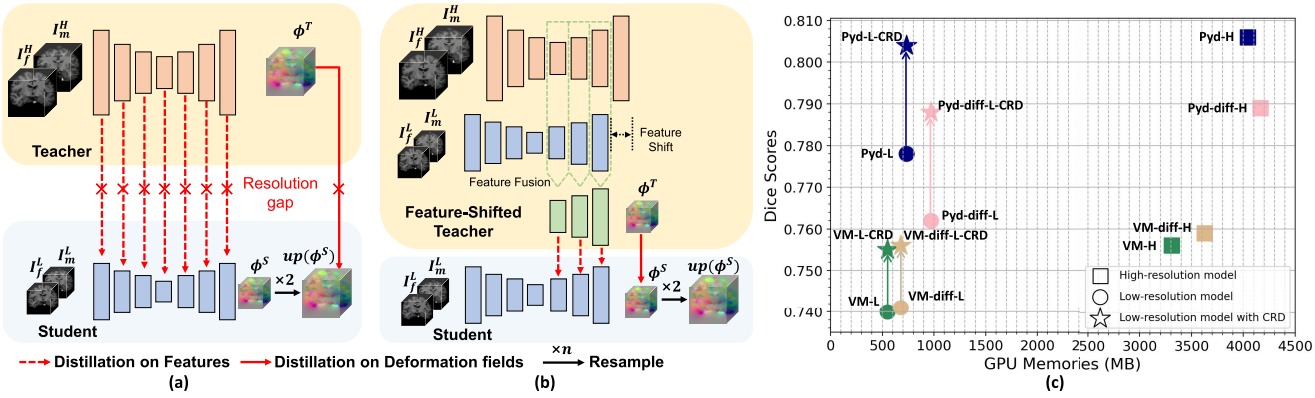

Fig. 1. (a) Straightforward distillation that is difficult to apply due to the resolution gap; (b) our proposed cross-resolution distillation scheme with a feature-shifted teacher. (c) Comparison of high/low-resolution models of different methods on OASIS. The low-resolution model trained through our scheme achieves competitive performance compared with corresponding high-resolution models while using fewer than 20% GPU memories.

shrinking the model size lowers learning abilities of the model, which leads to severe performance degradation even after distillation.

In this paper, we propose a novel cross-resolution distillation (CRD) scheme to reduce the computational burdens of learning-based registration models for 3D medical images. Instead of shrinking the model size as in previous works, our main idea is reducing the input resolution of existing registration models and boosting their performance through knowledge distillation. Specifically, we first downsample the input images by a factor of n at each dimension, and send them to our distilled low-resolution model, which directly makes the computation n^3 times less. We then upsample the deformation field generated by this low-resolution model back to the input resolution as the final output. This idea is simple, however, *can we design a competent teacher model to guarantee the performance of its low-resolution student after distillation?*

As shown in Fig. 1(a), a straightforward strategy to obtain a teacher model is taking the high-resolution images as inputs to train a model with the same architecture as the low-resolution model. However, this high-resolution teacher can hardly transfer knowledge to the low-resolution student. Taking VoxelMorph and $n = 2$ as an example, features of the high-resolution model are two times larger at each dimension than those of the low-resolution model at the corresponding level. This resolution gap makes straightforward distillation difficult to apply. Although directly downsampling the features of the teacher could be a solution, it significantly corrupts the knowledge of the teacher and leads to poor distillation effects. Moreover, since this straightforward high-resolution teacher is trained without low-resolution images, there is also a knowledge gap between the teacher and the student.

The following observation helps us find a way out: since the resolution of features differs two times at each dimension between adjacent levels in each model, features of the high-resolution model have the same resolution as those of the low-resolution one at the adjacent level.

Based on this observation, we design a feature-shifted teacher (FST) which shifts and fuses features of both high/low-resolution models, as shown in Fig. 1(b). Through this design, we properly deal with the resolution gap and the knowledge gap between the teacher and the student. We train the low-resolution student model under the guidance of the teacher with distillation losses on both features and deformation fields. Finally, we only need to use the distilled student model during inference.

Experimental results on four representative 3D medical image datasets, OASIS, Mindboggle-101, LPBA40 and SLIVER demonstrate the superiority and high efficiency of CRD. Compared with the corresponding high-resolution models, low-resolution models guided by our CRD scheme only consume fewer than 20% GPU memories as shown in Fig. 1(c) and less than 20% inference time while achieving competitive performance.

II. Related Works

A. Medical Image Registration

1) Traditional Methods: Traditional methods often model registration as an optimization problem and minimize the dissimilarity between each image pair under the smoothness constraint in an iterative approach. These methods include the elastic body models [18], [19], free-form deformations with B-Splines [20] and Demons [21]. There are also many methods proposed to obtain a deformation field within the space of diffeomorphic maps, so that the invertibility properties can be guaranteed. These methods include diffeomorphic B-Splines [22], diffeomorphic Demons [23], large deformation diffeomorphic metric [24] and symmetric image normalization (SyN) [8].

2) Learning-Based Methods: Recent methods formulate the registration procedure as a mapping from an input image pair to a deformation field with the help of deep learning. Learning-based methods can be categorized as supervised methods and unsupervised methods. Supervised methods [10], [25]–[28] rely on manual annotations for learning models. SVF-Net [25] builds reference deformations by utilizing segmented regions of interests and estimates deformation fields with a U-Net structure. Deformable registration pyramid [28] integrates the pyramid structure, cost volume, and affine transformation into its proposed registration framework.

Free from annotations, unsupervised methods [7], [11], [12], [14], [15], [29]–[32] train models by similarity metrics and attract extensive research attention. Unsupervised methods can be further categorized as U-Net-based methods which estimate deformation fields directly, and Pyramid-based methods which estimate deformation fields in a coarse-to-fine manner. For U-Net-based methods, VoxelMorph [14] estimates deformation fields through U-Net structure. The diffeomorphic version of VoxelMorph [11] proposes to estimate deformation fields through solving a differential equation of stationary velocity field (SVF). AVSM [33] further proposes to estimate deformation fields through solving the equation of momentum-based SVF to better preserve diffeomorphic properties. Dual-PRNet [12] first proposes the coarse-to-fine estimation with a feature pyramid, and LapIRN [31] proposes to employ the Laplacian pyramid to estimate and fuse deformation fields in different resolution levels. In addition to U-Net-based and pyramid-based methods, VTN [16] and RCN [15] regard a U-Net as a subnetwork, and cascade several subnetworks together to boost the performance of a single one. CRN [34] and HyperMorph [35] propose to adaptively tune the hyperparameter of the loss function for unsupervised registration.

B. Knowledge Distillation

Knowledge distillation is presented to transfer knowledge from a cumbersome teacher model to a light student model, so that the performance of the light student model can be improved without extra computation or storage. Hinton *et al.* [36] propose direct distillation on logits by mimicking the output vector of the teacher model. Romero *et al.* [37] propose to transfer knowledge on intermediate features from the teacher to the student. Zagoruyko and Komodakis [38] build attention maps from features and apply distillation on them. Qi *et al.* [39] propose to boost low-resolution detection model with a feature-aligned teacher. Zhang *et al.* [40] propose an evolutionary teacher by minimizing the capability gap between the teacher and the student. Recently, many works introduce the idea of distillation to medical image analysis. Xing *et al.* [41] solve the overfitting problem in medical image detection by distillation. Dian *et al.* [42] propose a comprehensive distillation architecture for medical image segmentation. Yu *et al.* [43] apply distillation to cardiac motion estimation, which is a closely related area to medical image registration. They cascade two identical networks together as the teacher, and then employ its generated deformation filed to guide the learning of a single student network. Tran *et al.* [17] propose a distillation method ALDK for deformable image registration. There are mainly two differences between the ALDK method and our CRD scheme. Regarding the distillation ideas, the main idea of the ALDK method is to employ a cumbersome teacher model to guide the learning of the light-weight student model LDR, while the main idea of our CRD scheme is to employ the teacher model that takes high-resolution images as inputs to guide the learning of the student model that takes corresponding low-resolution images as inputs. In other words, the efficient registration of the ALDK method comes from the shrunk model size, while the efficient registration of our CRD scheme comes from a new perspective in which the input resolution is reduced. Regarding the distillation implementation, the ALDK method uses the adversarial learning to let the output of the light-weight student LDR mimic the output of the cumbersome teacher, while our CRD scheme uses the feature distillation and the deformation distillation to make the intermediate features and the output deformation fields between the student model and the teacher model similar.

III. Method

Our method is presented in three subsections. The first subsection briefly introduces the basic model and denotation. We take the representative VoxelMorph as the basic model for example to demonstrate our method, which can be easily extended to other learning-based methods. The second subsection describes the feature-shifted teacher. The third subsection describes a novel cross-resolution distillation scheme. The overall workflow is illustrated in Fig. 2.

A. Basic Model and Denotation

VoxelMorph takes a pair of images I_f and I_m as inputs, where I_f denotes the fixed image and I_m denotes the moving image. We use (X, Y, Z) to represent the (width, height, depth) of an input image respectively. VoxelMorph [14] employs a U-Net with the encoder-decoder architecture [44] as the model to estimate a deformation field ϕ, which is used to warp the moving image I_m to realize registration [45].

The loss function $\mathcal{L}$ of VoxelMorph consists of two terms, the similarity term $\mathcal{L}_{sim}$ and the regularization term $\mathcal{L}_{reg}$. For the similarity term $\mathcal{L}_{sim}$, we adopt the negative cross-correlation to penalize the intensity differences between I_f and $I_m(\phi) = I_m \circ \phi$, where $\circ$ denotes the warping operation and $I_m(\phi)$ denotes the moving image warped by ϕ. We denote the fixed and moving images which have subtracted the mean of a $9 \times 9 \times 9$ area $A(p)$ out as $\hat{I_f}(p)$ and $\hat{I_m}(\phi(p))$, then $\mathcal{L}_{sim}$ can be formulated in (1), as shown at the bottom of the next page, where p denotes each voxel in images, and p_i denotes each voxel around p over $A(p)$. For the regularization term $\mathcal{L}_{reg}$, we penalize the discontinuity of ϕ by optimizing its gradients, which can be formulated as

$$\mathcal{L}_{reg}(\phi) = \sum_{p} \|\nabla \phi(p)\|^2. \tag{2}$$

The final loss function $\mathcal{L}$ for VoxelMorph is defined as

$$\mathcal{L} = \mathcal{L}_{sim}(I_m(\phi), I_f) + \lambda \mathcal{L}_{reg}(\phi), \tag{3}$$

where λ is a hyperparameter.

We name the features of the decoder part as F_l according to their level l where $l \in \{0, 1, 2, 3, 4\}$. The spatial size of features F_l at the level l is (X_l, Y_l, Z_l), which is $1/2^l$ of that of the input images.

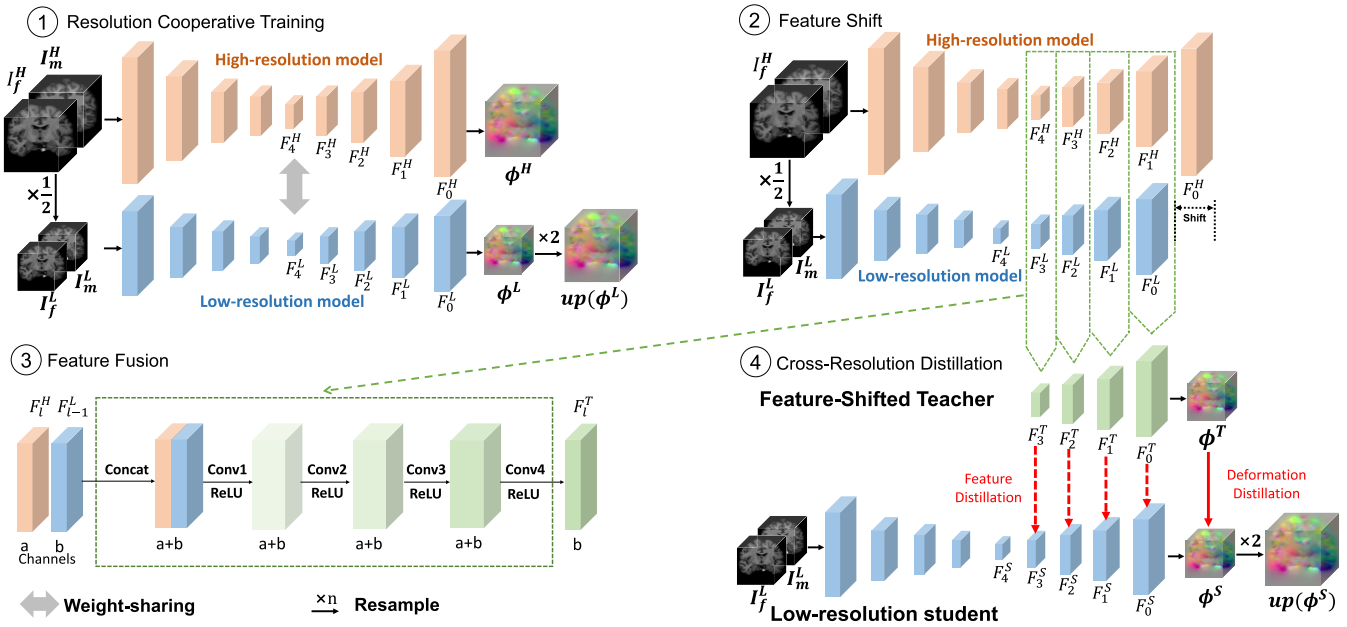

Fig. 2. Workflow of our method. First, we take both high-resolution and low-resolution images as inputs, and obtain both high/low-resolution models through resolution cooperative training. Then, we shift the low-resolution model by a level so that features at corresponding levels of high/low-resolution models can be matched. After that, we fuse features from both high/low-resolution models together and generate an advanced deformation field. Finally, we employ these fused features with the generated deformation field to guide the learning of the low-resolution student.

B. Feature-Shifted Teacher

We propose a feature-shifted teacher (FST) to guide the learning of the low-resolution student model. FST overcomes the resolution gap and the knowledge gap of features between high/low-resolution models. We separate the procedure of obtaining an FST to resolution cooperative training, feature shift and feature fusion.

1) Resolution Cooperative Training: As shown in the first part of Fig. 2, we initially train a network on both high/low-resolution input images to obtain both high/low-resolution models cooperatively in a weight-sharing manner. Given a high-resolution input pair $\{I_f^H, I_m^H\}$ (H denotes high resolution) with (X^H, Y^H, Z^H) as its spatial size, we use trilinear interpolation to downsample it to $\{I_f^L, I_m^L\}$ (L denotes low resolution) with (X^L, Y^L, Z^L) as the downsampled spatial size. The relationship of their resolution is $X^H = nX^L$, $Y^H = nY^L$, $Z^H = nZ^L$, where $n = 2^k$, and k is the valid[1] number of shifted levels in the following feature shift. In this paper, we set $n = 2$ as default. The network for a high-resolution pair generates a high-resolution deformation field ϕ^H, and for a low-resolution pair, the network generates a low-resolution deformation field ϕ^L. We upsample ϕ^L to $up(\phi^L)$, whose resolution is consistent with ϕ^H, so that $up(\phi^L)$ can register a high-resolution input pair as well, where $up(\cdot)$ denotes the upsampling operation realized by trilinear interpolation. We thus obtain the high/low-resolution models according to the high/low-resolution inputs to the network. The loss functions $\mathcal{L}^H$ and $\mathcal{L}^L$ for learning the high-resolution deformation field ϕ^H and low-resolution deformation field ϕ^L are defined as

$$\mathcal{L}^H = \mathcal{L}_{sim}(I_m^H(\phi^H), I_f^H) + \lambda \mathcal{L}_{reg}(\phi^H), \tag{4}$$

$$\mathcal{L}^L = \mathcal{L}_{sim}((I_m^H(up(\phi^L)), I_f^H) + \lambda \mathcal{L}_{reg}(up(\phi^L)), \tag{5}$$

where we set λ as 1 (the same below) following the setting of [14]. The entire loss function for resolution cooperative training is

$$\mathcal{L}_{cooperative} = \mathcal{L}^H + \mathcal{L}^L. \tag{6}$$

Through resolution cooperative training, we obtain both high/low-resolution models as well as their features F^H and F^L. Since high/low-resolution models are trained in a weight-sharing manner, the parameters of them are the same and the only difference between them is the input resolution. The weight-sharing manner unifies the optimization process of

[1] k should be an integer which lets the registration model work properly with a valid input size. Besides, for a registration model with m-level features, k cannot be larger than m. Otherwise, there are no features of the same resolution for high/low-resolution models to fuse after a k-level feature shift.

$$\mathcal{L}_{sim}(I_f, I_m(\phi)) = -\sum_p \frac{\left(\sum_{p_i}\left(I_f(p_i) - \hat{I}_f(p)\right)\left(I_m(\phi(p_i)) - \hat{I}_m(\phi(p))\right)\right)^2}{\left(\sum_{p_i}\left(I_f(p_i) - \hat{I}_f(p)\right)\right)\left(\sum_{p_i}\left(I_m(\phi(p_i)) - \hat{I}_m(\phi(p))\right)\right)}, \tag{1}$$

TABLE I
DETAILS OF CONVOLUTIONAL LAYERS IN FEATURE FUSION. INPUT/OUTPUT C. MEANS THE NUMBER OF INPUT/OUTPUT CHANNELS

Layer	Kernel size	Stride	Input C.	Output C.	Padding
Conv1	$3 \times 3 \times 3$	1	a+b	a+b	SAME
Conv2	$3 \times 3 \times 3$	1	a+b	a+b	SAME
Conv3	$3 \times 3 \times 3$	1	a+b	a+b	SAME
Conv4	$3 \times 3 \times 3$	1	a+b	b	SAME

high/low-resolution models, which eliminates the knowledge gap between the teacher and the student caused by different input resolutions in the following distillation.

2) Feature Shift: After resolution cooperative training, we get both high/low-resolution features, and utilize them to prepare for distillation, as shown in the second part of Fig. 2. However, there is a resolution gap between features at corresponding levels of high/low-resolution models as mentioned in Section I and Fig. 1(a). We propose feature shift to solve the problem of the resolution gap.

We observe that the decoder features F_l^H of the high-resolution model at the current level l have the same resolution with the decoder features F_{l-1}^L of the low-resolution model at the previous level $l-1$. The relationship of their spatial sizes is $(X_l^H = X_{l-1}^L, Y_l^H = Y_{l-1}^L, Z_l^H = Z_{l-1}^L)$. Besides, since the decoder features are concatenated with the encoder features as depicted in [14], [44], the decoder features can represent the entire features of a model. Based on this observation, we shift the low-resolution model by a level as illustrated in the second part in Fig. 2, so that the spatial size of features in high/low-resolution models can be matched.

3) Feature Fusion: After feature shift, we freeze the parameters of both high/low-resolution models in case that the following training disturbs these well-trained features. The experiments on the separate training strategy are provided in Section V-D-3). We fuse the concatenation of $\{F_{l-1}^L, F_l^H\}$ and get fused features F_{l-1}^T as shown in the third part of Fig. 2. Specifically, we fuse $\{F_1^H, F_2^H, F_3^H, F_4^H\}$ with $\{F_0^L, F_1^L, F_2^L, F_3^L\}$ and generate features $\{F_0^T, F_1^T, F_2^T, F_3^T\}$ separately. Each fusion operation is the same: the low-resolution features F_{l-1}^L is concatenated with the high-resolution features F_l^H at the channel dimension, then we use four convolutions with leaky ReLU to obtain the fused features. The details of convolutional layers in feature fusion are shown in Table I. The number of channels of the high-resolution features F_l^H and the low-resolution features F_{l-1}^L before fusion are a and b separately. Here we set the output number of channels of the last convolution in feature fusion as b. Therefore the fused features F_l^T can have the same number of channels as the student features F_l^S, making the feature distillation applicable.

Between high/low-resolution models, the features F^H from the high-resolution model provide information based on high-resolution images while the features F^L from the low-resolution model provide information close to the low-resolution student model. Inside each model, the features F_l at the deep level l provide global information while the features F_{l-1} at the shallow level $l-1$ provide local information. Since we shift the low-resolution model by a level to match the resolution of features in the high-resolution model, the features to fuse are the deep features F_l^H in the high-resolution model, and the shallow features F_{l-1}^L in the low-resolution model. With feature fusion, we combine the strengths of features from both high/low-resolution models, as well as the strengths of features at both shallow/deep levels. We regard these fused features as the teacher features F_l^T.

Based on F_l^T, we apply the decoder structure which is the same as the basic model to estimate a deformation field ϕ^T. As the resolution of ϕ^T is only half of the high-resolution input, we upsample it by a factor of 2 so that $up(\phi^T)$ can register images in high resolution. The loss function to optimize ϕ^T is

$$\mathcal{L}^T = \mathcal{L}_{sim}(I_m^H(up(\phi^T)), I_f^H) + \lambda \mathcal{L}_{reg}(up(\phi^T)). \quad (7)$$

Through this optimization, the fusion part can utilize both high/low-resolution features to generate an advanced low-resolution deformation field for the following distillation.

C. Cross-Resolution Distillation

The main idea of cross-resolution distillation (CRD) is to adopt the high-resolution feature-shifted teacher (FST) model to guide the learning of the corresponding low-resolution student model. After resolution cooperative training, feature shift and feature fusion, we obtain an FST model. Then, we adopt the FST model to guide the learning of the low-resolution student model on both features and deformation fields, which is named deformation distillation and feature distillation separately. The deformation distillation is used to guide the low-resolution student on the output fields while the feature distillation is used to guide the low-resolution student on features, as shown in the fourth part of Fig. 2.

The low-resolution student is a model trained only with low-resolution inputs and has the same network architecture as the basic model. We take the decoder features F_l^S in the student model as the student features. Our CRD scheme contains deformation distillation and feature distillation.

1) Deformation Distillation: We first force the output deformation field of the student model ϕ^S close to that of the teacher model ϕ^T like previous methods [36], [43]. Deformation distillation is illustrated in the solid red line in the fourth part of Fig. 2. Specifically, we adopt the mean square error (MSE) as the loss function for deformation distillation

$$\mathcal{L}_{deformation} = \mathcal{L}_{MSE}(\phi^T, \phi^S). \quad (8)$$

With deformation distillation, the student model can therefore get guidance from the high-resolution teacher model on output deformation fields.

2) Feature Distillation: Besides distillation on output deformation fields, we further apply distillation on features. Feature distillation is illustrated in the dashed red lines in the fourth part of Fig. 2. The feature distillation are applied on decoder features between the teacher and the student. Since the decoder

features have contained with the encoder features as depicted in existing learning-based models [12], [14], the distillation on the decoder part can represent distillation on the entire model. Both the encoder and the decoder part of the model will be updated during the distillation. We use the teacher features $\{F_0^T, F_1^T, F_2^T, F_3^T\}$ to supervise the learning of the corresponding student features $\{F_0^S, F_1^S, F_2^S, F_3^S\}$ respectively. Owing to feature shift, the resolution of teacher features is the same as that of student features. And owing to feature fusion, the number of channels of teacher features is also the same as that of student features. We still adopt the MSE loss between the features of the teacher F_l^T and student F_l^S following [37] to transfer knowledge. By forcing F_l^S close to F_l^T, the feature learning of the low-resolution student can be boosted under the guidance of the FST. The feature distillation loss is defined as

$$\mathcal{L}_{feature} = \sum_l \mathcal{L}_{MSE}(F_l^T, F_l^S). \tag{9}$$

The overall loss function for CRD is

$$\mathcal{L}_{distillation} = \mathcal{L}^L + \alpha \mathcal{L}_{deformation} + \beta \mathcal{L}_{feature}. \tag{10}$$

$\mathcal{L}^L$ is the loss function to learn a low-resolution deformation field the same as depicted in resolution cooperative training. α and β here are hyperparameters and we set them as 1 and 0.1 as default. Experiments for different hyperparameters are provided in Section V-D-4) and Table X.

With feature distillation, the student model can therefore get guidance from the high-resolution teacher model in feature learning.

3) Application to Pyramid-Based Methods: Both U-Net-based methods and pyramid-based methods are popular unsupervised registration methods. We have described our CRD scheme taking the U-Net-based method VoxelMorph as example above, and there is a slight difference when we apply CRD to pyramid-based methods. Unlike U-Net-based methods which estimate a deformation field from features directly, pyramid-based methods work in a coarse-to-fine manner and estimate the final deformation field with intermediate deformation fields ϕ_l. Therefore, when we apply CRD to pyramid-based methods, the deformation distillation loss formulated in Equation (8) should be modified to

$$\mathcal{L}_{deformation} = \sum_l \mathcal{L}_{MSE}(\phi_l^T, \phi_l^S), \tag{11}$$

because all intermediate deformation fields should also be guided. What is more, since regular pyramid-based methods adopt a dual-stream structure in their models, the feature distillation in Equation (9) should consider features of both streams. The other operations are the same as U-Net-based methods. We illustrate the difference in CRD between the U-Net-based methods and the pyramid-based methods in Fig. 3.

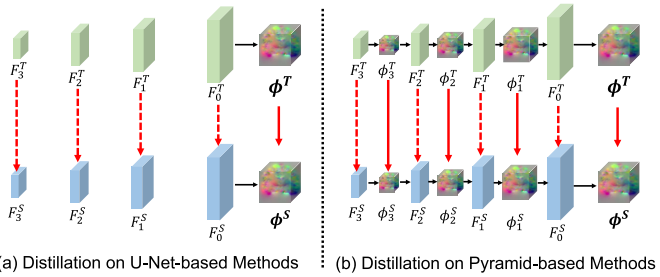

Fig. 3. Illustration of cross-resolution distillation on (a) U-Net-based methods and (b) Pyramid-based methods.

IV. Experiments

A. Datasets

We evaluate our proposed method with extensive experiments on four representative 3D medical image datasets, OASIS, Mindboggle-101, LPBA40 and SLIVER.

1) OASIS: OASIS [46] is a widely-recognized registration dataset, which is captured from the young, middle aged, non-demented and nemented older adults. We follow the version of Neurite OASIS [47] and make the preprocessing consistent with HyperMorph [35]. OASIS contains 414 images and all of them have been aligned to a template space previously. We use the 380 images of them to construct 380 × 379 pairs to train models, and use the rest to construct 14 × 13 and 20 × 19 pairs for validation and test. Each image is accompanied with a segmentation annotation on 35 issues. The resolution of 3D images in this dataset is 160 × 192 × 224.

2) Mindboggle-101: Mindboggle-101 dataset [48] is the largest human brain dataset with manually labeled annotation. We select 106 subareas to compute in the test phase. Mindboggle-101 contains 5 subsets, and we use NKI-RS-22 and NKI-TRT-20 as our training sets (42 × 41 pairs) and use MMRR-21 as our validation (8 × 7 pairs) set and test set (12 × 11 pairs). All of images have been pre-aligned to a template space. We carry standard preprocessing steps following [14] and crop 3D images to the resolution of 192 × 192 × 192.

3) LPBA40: LPBA40 [49] has 40 brain MRI scans with corresponding segmentation ground truth of 56 anatomical structures. We adopt ADNI [50], ABIDE [51] and ADHD [52] for training and adopt LPBA40 for validation and test following [17]. ADNI, ABIDE and ADHD have 66, 1287, 949 scans respectively. The setting of dataset splitting is the same as [17]. Every scan we use is resampled to 128 × 128 × 128 voxels after center-cropping. The preprocessing steps are consistent with VoxelMorph [14].

4) SLIVER: SLIVER [53] consists of 20 CT liver scans with corresponding segmentation ground truth. We adopt MSD [54] and BFH [16] for training and adopt SLIVER for validation and test following [17]. MSD comes from a medical segmentation challenge and contains massive 3D medical CT scans for 10 types of organs. We select 993 scans of which most likely include livers in them. BFH comes from clinic and contains 92 scans. Every scan we use is resampled to 128 × 128 × 128 voxels after center-cropping. The preprocessing steps are consistent with VoxelMorph [14].

B. Implementation

1) Registration Strategy: We take the subject-to-subject registration strategy for our experiments, which means the

fixed image is not set beforehand. In this strategy, a learning-based model takes two arbitrary images as inputs, and registers one of them to the other. Compared to atlas-based registration in which every image needs to be registered to the same image, subject-to-subject registration can generalize to more complex scenarios.

2) Training Strategies: For training the teacher model, we first train 5×10^5 steps in the resolution cooperative training phase. Then, we train another 5×10^5 steps in the feature shift and fusion phase. For distillation, we train 5×10^5 steps for the student model. The initial learning rate for each phase is 10^{-4}, and halves at 3×10^5 steps and 4×10^5 steps. We utilize the Adam [55] as our optimizer with $\beta_1 = 0.9, \beta_1 = 0.99$. LPBA40 and SLIVER are datasets without previous alignment, and OASIS and Mindboggle-101 are datasets with each scan being transformed affinely to the same template. For a fair comparison, we follow [16] to embed the same affine network on top of each learning-based model when experimenting on LPBA40 and SLIVER, so that it can predict an affine transformation before the deformable transformation.

3) Environment: The environment of all experiments of learning-based models is based on PyTorch 1.1 and we set the batch size as 4 on 4 NVIDIA TITAN XP GPUs. We implement traditional methods on CPUs of Xeon E5-2650 V4.

C. Evaluation Metrics

1) Similarity Metric: We use the average Dice score of all anatomical segments in image pairs as the similarity metric, which can reflect the overlap degree for an image pair. The Dice score 1 depicts an ideal registration, while the Dice score 0 depicts completely misalignment. Higher Dice scores mean better similarity properties.

2) Diffeomorphic Metric: We adopt the number of negative Jacobian determinants of the deformation field (denoted as Folds) as our quantitative metric for diffeomorphic properties. Jacobian matrices encode the local stretching, shearing and rotating of the deformation field, and their determinants indicate relative volumes before and after spatially transforming. A region of negative determinants indicates that the one-to-one mapping has been lost [9]. So lower Folds mean better diffeomorphic properties.

3) Efficiency Metrics: We adopt the usage of GPU memories[2] and inference time of models on a single GPU card NVIDIA TITAN XP in each method as our efficiency metrics. For a fair comparison, we subtract the system inherent usage during inference as our memory metric.

D. Comparison Methods

1) Baseline Methods: We verify the effectiveness of CRD on two types of learning-based methods: the U-Net-based method and the pyramid-based method. In terms of U-Net-based methods, we choose the most representative registration methods VoxelMorph [14] (abbreviated as VM) and its diffeomorphic version (abbreviated as VM-diff) [11]. In terms of pyramid-based methods, Dual-PRNet [12] and LapIRN [31] are the state-of-the-art pyramid-based methods with high registration performance. Since Dual-PRNet lacks official implementation and the diffeomorphic version, and LapIRN contains progressive training phases which is hard to share parameters between high/low-resolution models, we reimplement a pyramid network (abbreviated as Pyd) with its diffeomorphic version as mentioned in [11] (abbreviated as Pyd-diff). Pyd and Pyd-diff are based on the structure mentioned in the paper of Dual-PRNet and are trained in an end-to-end manner. We adjust the number of convolution layers and channels to make sure Pyd performs no worse than Dual-PRNet and LapIRN. We regard VM and Pyd which take high/low-resolution input images (denoted as -H/-L) as baseline methods. In addition, we also provide three state-of-the-art registration methods as a comparison. They are SyN [8], BSplines [20] and VTN [16]. SyN and BSplines are top-performing traditional methods, which are implemented by software packages ANTs [56] and Elastix [57] respectively, and VTN is another popular U-Net-based learning-based method.

2) Efficient Methods: Existing efficient registration methods focus on shrinking the model size without changing input resolution. We first compare our CRD scheme with the existing registration distillation method ALDK [17]. The ALDK method proposes a light-weight student model LDR first, then uses the adversarial training to let the output of LDR mimic the output of the cumbersome teacher to obtain performance improvement. Besides, we provide four efficient methods for our baseline methods: VM-light-H, VM-lighter-H for VM-H and Pyd-light-H, Pyd-lighter-H for Pyd-H. VM-light-H is the official light version of VM-H [14], which has fewer convolutional layers and channels. To further explore the impact of shrinking model sizes on registration performance, we halve the number of convolutional channels of VM-light-H to obtain VM-lighter-H. For Pyd, similarly, we halve the convolution channels once and twice to obtain Pyd-light-H and Pyd-lighter-H separately. Since shrinking the model size lowers learning abilities and leads to performance degradation, it is necessary to enhance these shrunk models to guarantee their performance and knowledge distillation is a popular strategy to realize it. We choose the most representative method [36] in distillation and a distillation method in cardiac motion estimation [43] which is close to medical image registration to enhance the shrunk registration models mentioned above. Besides, we adopt the distillation method ALDK proposed in [17] to enhance the shrunk models mentioned above. These methods can also be regarded as a comparison to our CRD scheme.

V. Results

A. Comparison With Baseline Methods

Table II shows the quantitative results for baseline methods on OASIS and Mindboggle-101. As a comparison, we first give the results of several state-of-the-art registration methods on the top six rows. For VM on OASIS, compared with VM-H, VM-L has an obvious performance drop

[2]The command we use to compute the consumption of GPU memories is: torch.cuda.max_memory_allocated()

TABLE II

COMPARISON WITH BASELINE METHODS. THE TOP SIX ROWS ARE BASELINE METHODS ON OASIS AND MINDBOGGLE-101. H DENOTES THE HIGH-RESOLUTION MODEL WHILE L DENOTES THE LOW-RESOLUTION MODEL. NOTE THAT, THE TEACHER OF OUR CRD SCHEME IS OUR PROPOSED FST INSTEAD OF A RAW HIGH-RESOLUTION MODEL, AND WE COMPARE THE RESULTS OF DIFFERENT TEACHERS IN SECTION V-D-1). HIGHER DICE MEANS BETTER SIMILARITY PERFORMANCE WHILE LOWER FOLDS MEAN BETTER DIFFEOMORPHIC PROPERTY. **BOLD** FONTS MEAN THE BEST IN THE SAME CLASS METHODS, AND UNDERLINE FONTS MEAN THE SECOND BEST. STANDARD DEVIATIONS ARE IN PARENTHESES

Methods	OASIS				Mindboggle-101			
	Dice	Folds	GPU (MB)	GPU (s)	Dice	Folds	GPU (MB)	GPU (s)
SyN-H	0.748 (0.027)	0 (0)	-	2683	0.549 (0.021)	0 (0)	-	2452
B-Splines-H	0.746 (0.032)	2681 (2251)	-	378	0.544 (0.021)	1532 (1388)	-	353
VTN-H	0.725 (0.031)	3106 (1344)	1633	1.081	0.509 (0.014)	528 (1496)	1827	1.124
Dual-PRNet-H	0.734 (0.031)	3931 (1796)	1343	0.173	0.521 (0.011)	1586 (2867)	1485	0.181
LapIRN-H	0.802 (0.023)	2325 (1768)	5435	0.835	0.616 (0.013)	1999 (1673)	5571	0.875
LapIRN-diff-H	0.795 (0.021)	0 (0)	5721	2.892	0.606 (0.011)	0 (0)	5973	2.986
VM-H	**0.756 (0.028)**	3183 (1410)	3307	0.156	**0.556 (0.018)**	2809 (1429)	3395	0.159
VM-L	0.740 (0.027)	3861 (2037)	**551**	**0.023**	0.542 (0.014)	1826 (1215)	**585**	**0.024**
VM-L-CRD	0.755 (0.028)	**2762 (1268)**	**551**	**0.023**	0.554 (0.016)	**1733 (961)**	**585**	**0.024**
VM-diff-H	**0.759 (0.028)**	0 (0)	3627	2.165	**0.552 (0.015)**	0 (0)	3725	2.212
VM-diff-L	0.741 (0.025)	0 (0)	**679**	**0.315**	0.532 (0.014)	0 (0)	**691**	**0.331**
VM-diff-L-CRD	0.756 (0.027)	0 (0)	**679**	**0.315**	0.551 (0.015)	0 (0)	**691**	**0.331**
Pyd-H	**0.806 (0.023)**	3886 (1549)	4043	0.604	**0.613 (0.011)**	5261 (1316)	4165	0.628
Pyd-L	0.778 (0.022)	**1495 (678)**	**735**	**0.091**	0.584 (0.009)	**1593 (1482)**	**757**	**0.097**
Pyd-L-CRD	0.804 (0.022)	1711 (1086)	**735**	**0.091**	0.610 (0.010)	1988 (1355)	**757**	**0.097**
Pyd-diff-H	**0.789 (0.023)**	0 (0)	4115	2.538	**0.586 (0.011)**	0 (0)	4221	2.751
Pyd-diff-L	0.762 (0.022)	0 (0)	**969**	**0.232**	0.562 (0.009)	0 (0)	**997**	**0.256**
Pyd-diff-L-CRD	0.788 (0.022)	0 (0)	**969**	**0.232**	0.583 (0.011)	0 (0)	**997**	**0.256**

(Dice/Folds: 0.756→0.740/3183→3861) but consumes much fewer GPU memories (3307MB→551MB) and inference time (0.156s→0.023s). Under the guidance of our proposed CRD, the low-resolution model VM-L-CRD gets a much strong performance (Dice/Folds: 0.755/2762), which is comparable with the high-resolution model VM-H (Dice/Folds: 0.756/3183). Note that, the teacher to guide VM-L for CRD is not a naive VM-H but FST. FST takes advantage of features from both high/low-resolution model and is stronger than VM-H. Therefore, the guided low-resolution model VM-L-CRD can perform very close to VM-H. The model capacity is NOT the determining factor for the success of FST as the teacher, and we give the analysis of different teachers in Section V-D-1). For Pyd on OASIS, Pyd-L-CRD achieves performance (Dice/Folds: 0.804/1711) that is comparable to Pyd-H (Dice/Folds: 0.806/3886), but saves a mass of GPU memories (4043MB→735MB) and inference time (0.604s →0.091s). We give analysis of results on OASIS above. The advantage of our method is consistent with that on Mindboggle-101. Two visualization examples are shown in Fig. 4.

We also present the quantitative results for baseline methods on LPBA40 and SLIVER in Table IV. Similar to the superiority of our CRD scheme on OASIS and Mindboggle-101, VM-L-CRD and Pyd-L-CRD are competitive or better in terms of the performance on LPBA40 and SLIVER, and consume much less GPU memories and inference time compared to all other methods. Note that, the spatial resolution of LPBA40 and SLIVER is the same, so we only validate the GPU memories and inference time taking LPBA40 as an example. Besides, since each method is embeded with an affine network when validating on LPBA40 and SLIVER as mentioned in Section IV-B-2), the consumption of model inference time and GPU memories contains the affine network part.

B. Comparison With Efficient Methods

We compare our CRD scheme with other efficient methods on OASIS and Mindboggle-101 in Table III. We provide the results in the displacement version, and the advantage of our method in the diffeomorphic version is consistent.

We first compare our method with the existing distillation method ALDK. As shown in Table III, the light-weight model LDR-H before distillation has the performance with Dice/Folds 0.731/6624 on OASIS and 0.528/2762 on Mindboggle-101. After distillation, LDR-H-ALDK obtains an obvious performance improvement to Dice/Folds 0.742/4027 on OASIS and 0.539/2206 on Mindboggle-101, but the results are still lower than the VM-L and Pyd-L guided by our CRD scheme. Moreover, our VM-L-CRD and Pyd-L-CRD consume much less GPU memories (1375MB→551MB on OASIS, 1419MB→585MB on Mindboggle-101) and time (0.106s→0.023s on OASIS, 0.108s→0.024s on Mindboggle-101) compared to LDR-H-ALDK.

We then compare the efficient methods taking VM and Pyd as examples separately. For VM, compared to VM-H, the official light version VM-light-H reduces the consumption of GPU memories (3307MB→2665MB on OASIS, 3395MB→2735MB on Mindboggle-101) and inference time (0.156s→0.112s on OASIS, 0.159s→0.120s on Mindboggle-101). Meanwhile, the performance of VM-light-H drops significantly (Dice/Folds:0.756→0.736/3183→5100

TABLE III

COMPARISON WITH OTHER EFFICIENT METHODS ON OASIS AND MINDBOGGLE-101. VM-LIGHT-H IS THE OFFICIAL LIGHT VERSION OF VOXELMORPH [14], AND WE FURTHER HALVE THE NUMBER OF CONVOLUTIONAL CHANNELS TO OBTAIN VM-LIGHTER-H AS AN EXPLORATION. SIMILARLY, WE HALVE THE NUMBER OF CONVOLUTIONAL CHANNELS OF PYD-H TO OBTAIN PYD-LIGHT-H, PYD-LIGHTER-H SEPARATELY. THESE METHODS ALL FOCUS ON SHRINKING MODEL SIZES WITH PERFORMANCE DROP, AND WE BOOST THEM THROUGH DISTILLATION METHODS [36], [43]. DIFFERENT FROM THEM, OUR CRD SCHEME FOCUSES ON REDUCING THE INPUT RESOLUTION AND WE BOOST THE PERFORMANCE OF THE LOW-RESOLUTION MODEL UNDER THE GUIDANCE OF FST

Methods	OASIS				Mindboggle-101			
	Dice	Folds	GPU (MB)	GPU (s)	Dice	Folds	GPU (MB)	GPU (s)
LDR-H	0.731 (0.033)	6624 (3891)	1375	0.106	0.528 (0.023)	2762 (1845)	1419	0.108
LDR-H-ALDK	0.742 (0.026)	4027 (2654)	1375	0.106	0.539 (0.018)	2206 (1269)	1419	0.108
VM-H	0.756 (0.028)	3183 (1410)	3307	0.156	0.556 (0.018)	2809 (1429)	3395	0.159
VM-light-H	0.736 (0.036)	5100 (2212)	2665	0.112	0.533 (0.022)	2760 (1640)	2735	0.120
VM-light-H-[36]	0.741 (0.032)	3836 (1854)	2665	0.112	0.545 (0.022)	1745 (896)	2735	0.120
VM-light-H-[43]	0.744 (0.033)	3275 (1758)	2665	0.112	0.545 (0.022)	2208 (1096)	2735	0.120
VM-light-H-ALDK	0.747 (0.038)	3989 (1985)	2665	0.112	0.547 (0.026)	2681 (1034)	2735	0.120
VM-lighter-H	0.705 (0.038)	7764 (4557)	1723	0.071	0.508 (0.025)	2966 (1958)	1758	0.078
VM-lighter-H-[36]	0.725 (0.033)	7458 (4219)	1723	0.071	0.517 (0.025)	3015 (2203)	1758	0.078
VM-lighter-H-[43]	0.734 (0.035)	4963 (3766)	1723	0.071	0.529 (0.023)	2769 (1834)	1758	0.078
VM-lighter-H-ALDK	0.741 (0.039)	5438 (4019)	1723	0.071	0.538 (0.021)	2981 (2043)	1758	0.078
VM-L-CRD	**0.755 (0.028)**	**2762 (1268)**	**551**	**0.023**	**0.554 (0.015)**	**1733 (961)**	**585**	**0.024**
Pyd-H	0.806 (0.023)	3886 (1549)	4043	0.604	0.613 (0.011)	5261 (1316)	4165	0.628
Pyd-light-H	0.793 (0.023)	2423 (920)	2657	0.512	0.598 (0.011)	2981 (816)	2819	0.523
Pyd-light-H-[36]	0.796 (0.024)	3568 (2936)	2657	0.512	0.603 (0.011)	3148 (1751)	2819	0.523
Pyd-light-H-[43]	0.801 (0.023)	2444 (1038)	2657	0.512	0.604 (0.012)	2352 (986)	2819	0.523
Pyd-light-H-ALDK	0.802 (0.028)	2867 (2194)	2657	0.512	0.606 (0.014)	2759 (1368)	2819	0.523
Pyd-lighter-H	0.773 (0.025)	1714 (693)	2005	0.429	0.584 (0.012)	2070 (1182)	2141	0.448
Pyd-lighter-H-[36]	0.780 (0.026)	3852 (2586)	2005	0.429	0.588 (0.012)	3241 (2026)	2141	0.448
Pyd-lighter-H-[43]	0.782 (0.024)	**1537 (998)**	2005	0.429	0.591 (0.011)	**1478 (1123)**	2141	0.448
Pyd-lighter-H-ALDK	0.789 (0.029)	1974 (1568)	2005	0.429	0.596 (0.016)	1783 (1523)	2141	0.448
Pyd-L-CRD	**0.804 (0.024)**	1711 (1086)	**735**	**0.091**	**0.610 (0.010)**	1988 (1355)	**757**	**0.097**

on OASIS, 0.556→0.533/2809→2760 on Mindboggle-101). The model VM-lighter-H consumes much fewer GPU memories compared to VM-light-H (2665MB→1723MB on OASIS, 2735MB→1758MB on Mindboggle-101) and inference time (0.112s→0.071s on OASIS, 0.120s→0.078s on Mindboggle-101). By reason of the heavy model size shrinking, the performance of VM-lighter-H is much weaker than VM-light-H (Dice/Folds: 0.736→0.705/5100→7764 on OASIS, 0.533→0.508/2760→2966 on Mindboggle-101). Thanks to knowledge distillation, the performance of these shrunk models gets improvement without extra burdens. For example, VM-light-H-[36] achieves Dice/Folds 0.741/3836 on OASIS and 0.545/1745 on Mindboggle-101, which is obviously better than VM-light-H. VM-light-H-ALDK achieves Dice/Folds 0.747/3989 on OASIS and 0.547/2681 on Mindboggle-101. These results demonstrate that the ALDK method provides promising distillation effects. All of these models take high-resolution images as inputs and focus on shrinking model sizes. Our VM-L-CRD demands the least GPU memories (551MB on OASIS, 585MB on Mindboggle-101) and the shortest inference time (0.023s on OASIS, 0.024s on Mindboggle-101). This result proves that our CRD scheme is much more efficient than other methods based on shrinking model sizes. Meanwhile, VM-L-CRD achieves the best performance (Dice/Folds: 0.755/2762) on both OASIS and Mindboggle-101. We provide a visualization comparison example in Fig. 5 of different efficient methods based on VM on OASIS. For Pyd, the advantage of our CRD is consistent. Among all efficient methods including the ALDK methods, Pyd-L-CRD achieves the best registration performance with the least consumption of GPU memories (735MB on OASIS, 757MB on Mindboggle-101) and time (0.091s on OASIS, 0.097s on Mindboggle-101).

We also compare the results on LPBA40 and SLIVER in Table IV. Compared to the ALDK method, VM-L and Pyd-L guided by our CRD scheme can achieve better performance. From the results in Table IV, the LDR-H guided by ALDK achieves Dice 0.672 and 0.892 on LPBA40 and SLIVER separately, while the VM-L-CRD scheme achieves Dice 0.688 and 0.910. In addition, VM-L-CRD are much more efficient than LDR-H-ALDK in terms of the consumption of GPU memories (1259MB→507MB) and inference time (0.082s→0.044s).

C. Exploration on Different Downsampling Factors

In our method, we downsample the high-resolution images by a factor of $n = 2^k$ at each dimension to obtain the low-resolution images, and we set $n = 2, k = 1$ for the CRD scheme as default, which means the input resolution to the low-resolution models is $1/2$ of that to the high-resolution models and the number of shifted levels is $k = 1$.

TABLE IV
COMPARISON RESULTS ON LPBA40 AND SLIVER

Methods	Dice (LPBA40)	Dice (SLIVER)	GPU (MB)	GPU (s)
SyN-H	0.709 (0.015)	0.903 (0.038)	-	748
B-Splines-H	0.681 (0.014)	0.912 (0.035)	-	115
VTN-H	0.693 (0.014)	0.916 (0.023)	1515	0.734
Dual-PRNet-H	0.697 (0.013)	0.922 (0.020)	1271	0.146
LapIRN-H	0.709 (0.015)	0.939 (0.021)	3863	0.428
LapIRN-diff-H	0.706 (0.015)	0.935 (0.024)	4045	1.573
LDR-H	0.672 (0.017)	0.892 (0.028)	1259	0.082
LDR-H-ALDK	0.684 (0.016)	0.907 (0.026)	1259	0.082
VM-H	**0.691 (0.015)**	**0.912 (0.025)**	3103	0.125
VM-L	0.678 (0.014)	0.897 (0.032)	**507**	**0.044**
VM-L-CRD	0.688 (0.016)	0.910 (0.026)	**507**	**0.044**
Pyd-H	**0.710 (0.014)**	0.940 (0.022)	3719	0.329
Pyd-L	0.698 (0.016)	0.905 (0.026)	**665**	**0.136**
Pyd-L-CRD	0.709 (0.014)	0.931 (0.023)	**665**	**0.136**

TABLE V
CRD OF DIFFERENT DOWNSAMPLING FACTORS ON MINDBOGGLE-101. L DENOTES THE 1/2-RESOLUTION MODEL WHILE LL DENOTES THE 1/4-RESOLUTION MODEL

Methods	Dice	Folds	GPU (MB)	GPU (s)
VM-H	0.556 (0.018)	2809 (1429)	3395	0.159
VM-L	0.542 (0.014)	1826 (1215)	585	0.024
VM-LL	0.484 (0.013)	220 (1495)	**293**	**0.005**
VM-LL-CRD	0.517 (0.013)	875 (843)	**293**	**0.005**
Pyd-H	0.613 (0.011)	5261 (1316)	4165	0.628
Pyd-L	0.584 (0.009)	1593 (1482)	757	0.097
Pyd-LL	0.527 (0.009)	379 (985)	**389**	**0.018**
Pyd-LL-CRD	0.561 (0.010)	1080 (1138)	**389**	**0.018**

Theoretically, as mentioned in Section III-B-1), we can set any valid integer k, and apply our CRD scheme with k-level feature shift. Since $k = 1$ is the largest valid number of feature shift for the CRD scheme of VoxelMorph and Pyd on OASIS, we explore the case $n = 4, k = 2$ on Mindboggle-101, in which the resolution of input images of the low-resolution model is only $1/4$ of the high-resolution inputs. We give the results in Table V, where L denotes the default 1/2-resolution model introduced in our previous sections, and LL denotes the 1/4-resolution model.

From the results, we can find that the consumption of GPU memories and time of 1/4-resolution models falls significantly when we compare them with the corresponding high-resolution models (3395MB→293MB, 0.159s→0.005s for VM, 4165MB→389MB, 0.628s→0.018s for Pyd). However, along with the greatly reduced input resolution, there is a severe performance drop. We thus boost these 1/4-resolution models through our CRD scheme, and the similarity performance gets remarkable improvement (Dice: 0.484→0.517 for VM, 0.527→0.561 for Pyd). Although the performance of the 1/4-resolution models boosted by CRD is not as good as the corresponding high-resolution and 1/2-resolution models, the ultra-low consumption of GPU memories and time still endow the 1/4-resolution models strong ability to apply to extremely resource-limited scenarios.

TABLE VI
DIFFERENT TEACHERS AND CORRESPONDING STUDENTS VM-L GUIDED BY THEM

Teacher	Dice	Folds
A (VM-H)	0.756 (0.028)	3183 (1410)
B (VM-large-H)	**0.776 (0.027)**	3487 (1453)
C (2-cascade VM-L)	0.752 (0.031)	**1111 (729)**
FST	0.759 (0.028)	2056 (1175)
Student	**Dice**	**Folds**
VM-L (w/o distillation)	0.740 (0.027)	3861 (2037)
VM-L by A	0.741 (0.029)	4358 (3681)
VM-L by A (feature ↓)	0.738 (0.027)	10284 (8785)
VM-L by B	0.743 (0.027)	4887 (3995)
VM-L by C	0.746 (0.030)	3272 (1978)
VM-L by FST	**0.755 (0.028)**	**2762 (1268)**

D. Ablation Studies

We provide ablation studies on OASIS taking VoxelMorph as an example.

1) Different Teachers: To verify the necessity and effectiveness of our proposed FST, we compare it with three other teacher models A, B, C in Table VI. All of them take the same low-resolution model VM-L as their student.

Teacher A is a raw VM-H. Due to the resolution gap as depicted in Section I and illustrated in Fig. 1(a), the straightforward distillation on features is difficult to apply. Thanks to the same resolution of deformation fields after upsampling, we can still apply deformation distillation. Since our teacher FST has more parameters and performs better than VM-H, the improvement of the guided VM-L may just come from a stronger teacher. Therefore, we design teacher B which has more parameters and better performance than our FST to demonstrate that the improvement of performance comes from the superiority of our FST. Specifically, teacher B (abbreviated as VM-large-H) is obtained by doubling the number of convolutional channels of VM-H. The number of B's parameters (1.2M) is four times larger than that of VM-H (0.3M) and two times larger than that of FST (0.6M). Teacher C is a two-cascade VoxelMorph, which takes low-resolution images as inputs (abbreviated as 2-cascade VM-L), and we apply distillation on output deformation fields following [43].

From the results of Table VI, we can see the effect of straightforward distillation is not satisfactory. In terms of teacher A, the performance of the VM-L guided by A (Dice/Folds: 0.741/4358) has no obvious improvement compared to the VM-L without distillation (Dice/Folds: 0.740/3861). That is because knowledge contained in corresponding features is hard to transfer by only distilling outputs. As depicted in Section I, there is a simple solution to this problem: we downsample the features at corresponding levels (abbreviated as feature ↓) to force the features

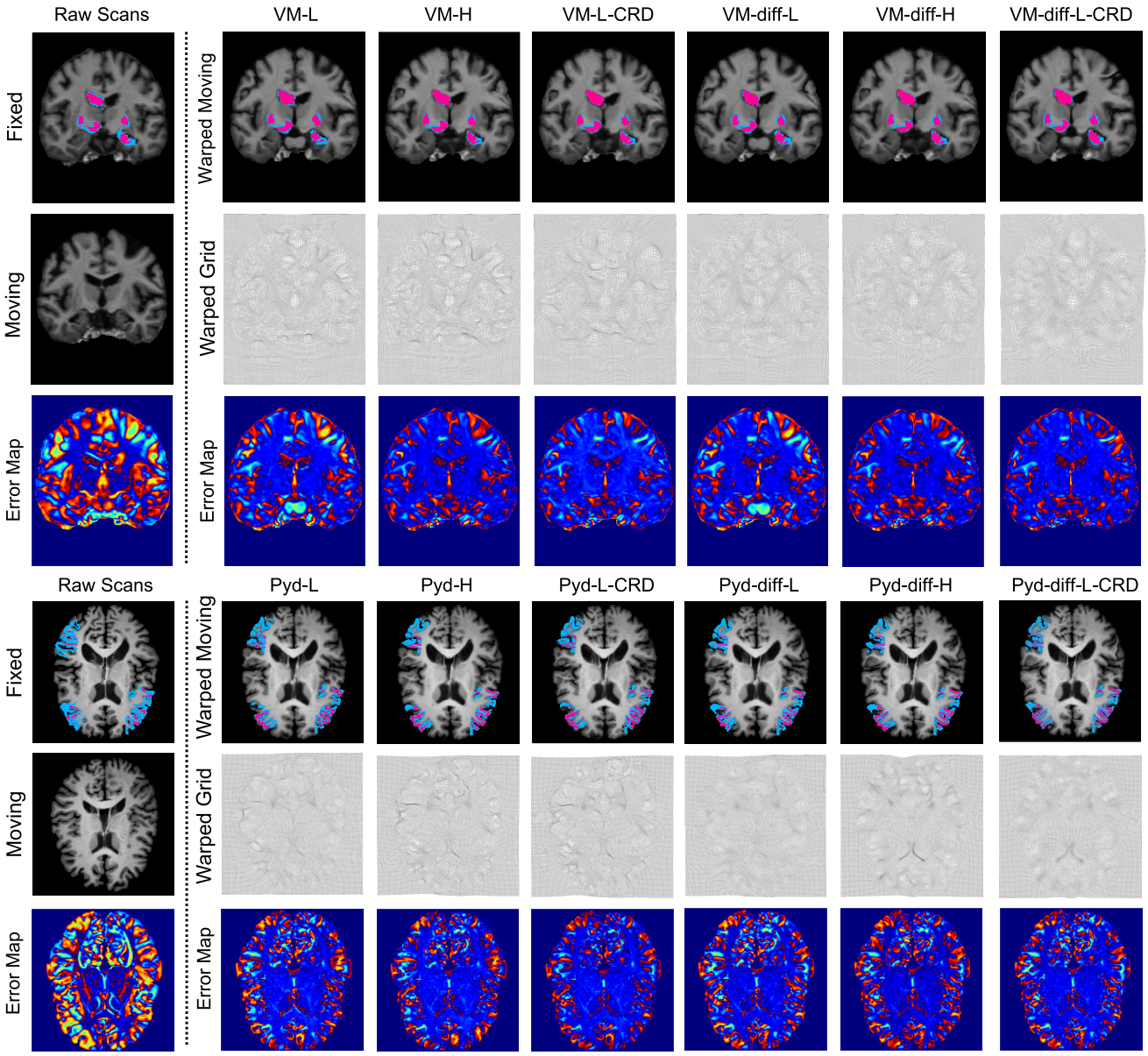

Fig. 4. Visualization of two examples on OASIS (top) and Mindboggle-101 (bottom) of different methods respectively. We provide the high/low-resolution results of VM on OASIS, and the results of Pyd on Mindboggle-101. The first row shows the target fixed image and warped moving images with five representive anatomical segments. The overlap of the fixed image and the (warped) moving image is red, and the rest part is blue. The second row shows the raw moving image on the left, and shows warped grids to visualize diffeomorphic property on the right. The third row shows error maps, which are obtained by each (warped) moving image minus the fixed image. The redder color means the larger registration error while the bluer color means the smaller.

of the teacher and student to have the same resolution, so that we can apply distillation on both deformation fields and features. The VM-L guided by A (feature ↓) in this way has the worst Dice/Folds 0.738/10284, which demonstrates the strategy of straightforward downsampling features harms knowledge and results in the bad distillation effect. In terms of teacher B, the performance of B (Dice/Folds: 0.776/3487) is much stronger than that of our FST (Dice/Folds: 0.759/2056), however, the performance of the VM-L guided by B (Dice/Folds: 0.743/4887) is lower than VM-L guided by our FST (Dice/Folds: 0.755/2762). Therefore, we can exclude the possibility that the performance improvement of the guided VM-L just comes from a stronger teacher with more parameters. In terms of teacher C, although C (Dice/Folds: 0.752/1111) performs worse than A and B since it takes low-resolution images as inputs, C does not have the problem of the knowledge gap and can transfer knowledge effectively to its student (Dice/Folds: 0.746/3272). However, as C lacks of knowledge learned from high-resolution inputs, the VM-L guided by C is still worse than our VM-L guided by FST. The VM-L guided by FST performs best in all guided student models.

2) Resolution Cooperative Training & Feature Shift & Feature Fusion: The procedure to obtain our FST includes resolution cooperative training (denoted as RC), feature shift (denoted as FS), and feature fusion (denoted as FF). We carry the ablation studies of each step of the teacher models and their impacts on guiding students in Table VII. Teacher D denotes a

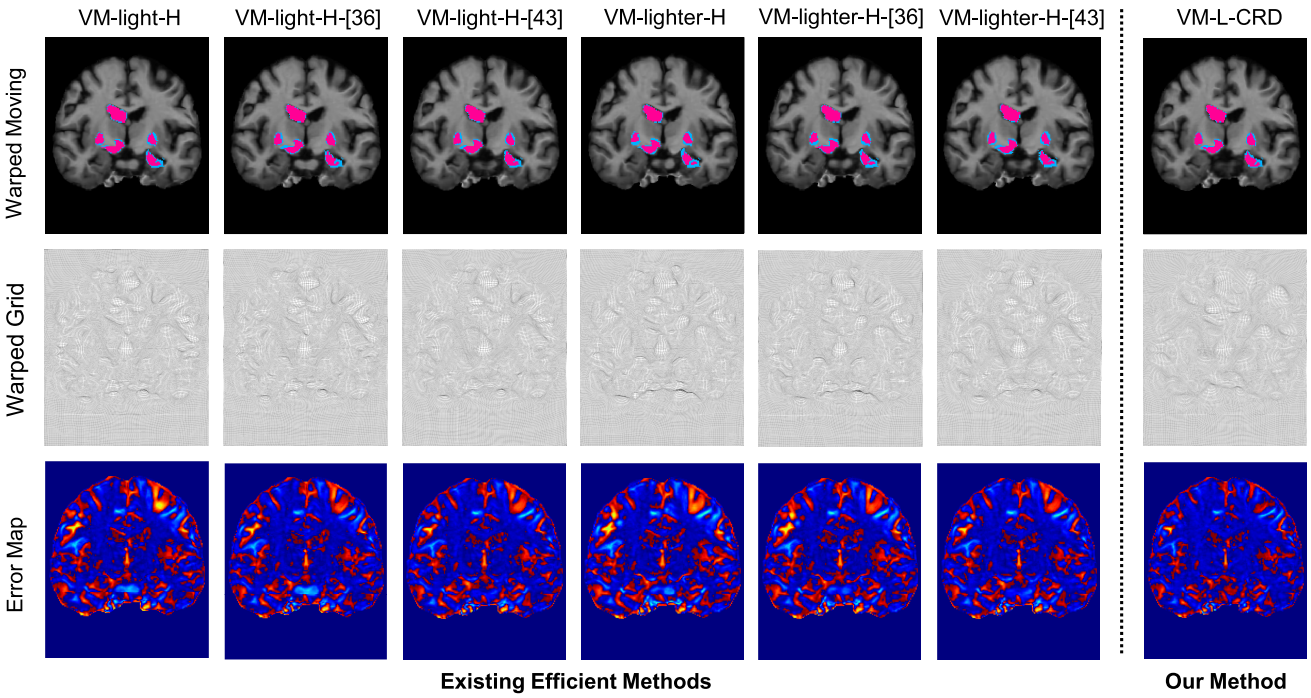

Fig. 5. Visualization of an example on OASIS for comparison of existing efficient registration methods based on VM (left) and our CRD scheme (right).

TABLE VII
ABLATION STUDIES ON RESOLUTION COOPERATIVE TRAINING (RC), FEATURE SHIFT (FS) AND FUSION (FF)

Teacher	RC	FS	FF	Dice	Folds
D	✓			0.750 (0.023)	1986 (892)
E		✓	✓	0.754 (0.026)	2953 (1168)
FST	✓	✓	✓	**0.759 (0.028)**	**2056 (1175)**
Student				Dice	Folds
VM-L (w/o distillation)				0.740 (0.027)	3861 (2037)
VM-L by D				0.743 (0.023)	2890 (1374)
VM-L by E				0.743 (0.024)	3204 (1873)
VM-L by FST				**0.755 (0.028)**	**2762 (1268)**

VM-H with RC, which means D is a VM that trained with both high/low-resolution input images. Although RC can eliminate the knowledge gap between high/low-resolution models as depicted in Section III-B-1), the problem of the resolution gap is still not solved, which makes it difficult to apply distillation on features of at corresponding levels. Therefore, we employ D to guide the low-resolution student VM-L with only deformation distillation. Teacher E denotes a VM-H with both FS and FF. With FS, the features of VM-H can have the same resolution with the features of VM-L. In spite of this, the number of channels of the shifted low-resolution features is still different from the corresponding high-resolution features. Therefore, we use the feature fusion as illustrated in the third part of Fig. 2 to solve this problem. After that, the number of channels of high/low-resolution features is equal, and we can apply both deformation distillation and feature distillation.

From the results of Table VII, we can see the performance of the VM-L guided by D is improved compared to the VM-L without distillation (Dice/Folds: 0.740→0.743/3861→2890). The reason is that D is trained with both high/low-resolution images, which makes features in D contain knowledge for a low-resolution model and contributes to knowledge transfer. For teacher E, E solves the problem of the resolution gap between high/low-resolution models with the help of FS and FF. Nevertheless, as there is no RC and the knowledge gap is not filled yet, the improvement of performance (Dice/Folds: 0.740→0.743/3861→3204) for the VM-L guided by E is less than the VM-L guided by FST. Our FST with RC, FS and FF achieves the best distillation effect with Dice/Folds 0.755/2762.

3) Separate Training: Our method consists of three-step training: the first step is resolution cooperative training (denoted as S1), the second step is feature shift and fusion (denoted as S2), and the third step is cross-resolution distillation (denoted as S3). We train an FST by the first and the second step, and then use it to guide the learning of low-resolution models following the third step. We train these three steps separately and fix the parameters of models in previous steps when training the current step. We give the comparison results of separate training and end-to-end training in Table VIII. We explore training these three steps together, where the whole teacher model and the student model are optimized jointly (named as teacher F). We also explore training the first and the second step together, which means the teacher model is obtained in an end-to-end manner, and the distillation part separates from them (named as teacher G).

We find the performance of teacher F is obviously poor (Dice/Folds: 0.731/4898) and much lower than FST.

TABLE VIII

ABLATION STUDIES ON SEPARATE TRAINING. THE STEPS IN THE SAME CURLY BRACES MEAN THAT THEY ARE TRAINED JOINTLY WHILE THE STEPS IN THE DIFFERENT CURLY BRACES MEAN THAT THEY ARE TRAINED SEPARATELY

Teacher	Training	Dice	Folds
F	{S1, S2, S3}	0.731 (0.032)	4898 (3259)
G	{S1, S2}, {S3}	0.755 (0.029)	2875 (1633)
FST	{S1},{S2},{S3}	**0.759 (0.028)**	**2056 (1175)**
Student		Dice	Folds
VM-L (w/o distillation)		0.740 (0.027)	3861 (2037)
VM-L by F		0.728 (0.033)	5861 (3576)
VM-L by G		0.746 (0.028)	3109 (2150)
VM-L by FST		**0.755 (0.028)**	**2762 (1268)**

TABLE IX

ABLATION STUDIES ON DISTILLATION LOSSES

Student	Def.	Fea.	Dice	Folds
VM-L	✓		0.751 (0.027)	2989 (1563)
VM-L		✓	0.742 (0.029)	5891 (3527)
VM-L	✓	✓	**0.755 (0.028)**	**2762 (1268)**

F's student performs even worse (Dice/Folds:0.728/5861) than the VM-L without distillation. The reason is that registration and distillation are very different procedures, and the optimization of them is different as well. The first and the second steps to obtain a teacher are responsible for registration, and the third step to guide the student is responsible for distillation.

We also find the performance of teacher G (Dice/Folds: 0.755/2875) and G's student (0.746/3109) is lower than our teacher FST and FST's student. An explanation is that the completely separate training strategy of FST can guarantee the features from both high/low-resolution models are already optimized well, so that the fusion part of the teacher can focus on fusing these well-trained features and combine their strengths.

4) Distillation Loss: Our distillation is based on both the deformation loss and the feature loss while previous works are only based on the deformation loss. We carry ablation studies to explore the respective impact of these two loss functions, and the results are shown in Table IX. First, we study the case when there is only one type of distillation. Although promising distillation performance can be achieved by only using the deformation loss, it is not sufficient for exploring the teacher's knowledge in features. And if we only use the feature loss, the distillation effect is not obvious (Dice/Folds: 0.742/5891). We think the reason is that the feature loss can only provide an indirect guidance to learn a deformation field. After applying the feature loss together with the deformation loss, the performance can be boosted which demonstrates that the direct feature distillation can be a good complement for the deformation distillation.

What is more, we also explore different hyperparameters α and β for deformation and feature distillation in the loss function in Equation (10). As shown in Table X, we can see our method is robust to the change of hyperparameters. Generally speaking, each set of parameters can lead to promising distillation effects. As the feature loss is a complement to the deformation loss, it should be assigned a small weight. Our default setting $\alpha = 0.1$ and $\beta = 0.1$ performs the best among all settings.

TABLE X

DIFFERENT HYPERPARAMETERS OF DISTILLATION LOSSES

Student	Hyperparams.		Dice	Folds
VM-L	α=0.1	β=0.1	0.750 (0.027)	3592 (2389)
VM-L	α=0.1	β=1	0.745 (0.031)	7791 (5434)
VM-L	α=**1**	β=**0.1**	**0.755 (0.028)**	**2762 (1268)**
VM-L	α=1	β=1	0.751 (0.029)	4689 (4021)

VI. CONCLUSION

We solve the problem of heavy computational burdens in medical image registration in a new perspective. Instead of shrinking the model size as in previous works, we turn to reducing the input resolution of existing registration models and boosting their performance through knowledge distillation. We propose a CRD scheme, which is designed to train low-resolution models under the guidance of corresponding high-resolution models. Experimental results on four representative 3D medical image datasets, OASIS, Mindboggle-101, LPBA40 and SLIVER, demonstrate that the low-resolution models trained through our CRD scheme use fewer than 20% GPU memories and less than 20% inference time while achieving competitive performance compared with corresponding high-resolution models.

REFERENCES

[1] I.-H. Lee and T.-S. Choi, "Accurate registration using adaptive block processing for multispectral images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 9, pp. 1491–1501, Sep. 2013.

[2] Z. Zhang, J. Sun, Y. Dai, B. Fan, and M. He, "VRNet: Learning the rectified virtual corresponding points for 3D point cloud registration," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jan. 14, 2022, doi: 10.1109/TCSVT.2022.3143151.

[3] J. Du, W. Li, K. Lu, and B. Xiao, "An overview of multi-modal medical image fusion," *Neurocomputing*, vol. 215, pp. 3–20, Nov. 2016.

[4] F. Alam, S. U. Rahman, S. Ullah, and K. Gulati, "Medical image registration in image guided surgery: Issues, challenges and research opportunities," *Biocybern. Biomed. Eng.*, vol. 38, no. 1, pp. 71–89, 2018.

[5] S. Mo *et al.*, "Mutual information-based graph co-attention networks for multimodal prior-guided magnetic resonance imaging segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2512–2526, May 2022.

[6] M. Li, S. Zhou, C. Chen, Y. Zhang, D. Liu, and Z. Xiong, "Retinal vessel segmentation with pixel-wise adaptive filters," in *Proc. IEEE 19th Int. Symp. Biomed. Imag. (ISBI)*, Mar. 2022, pp. 1–5.

[7] T. C. W. Mok and A. C. S. Chung, "Fast symmetric diffeomorphic image registration with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4644–4653.

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

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

[10] X. Cao *et al.*, "Deformable image registration based on similarity-steered CNN regression," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2017, pp. 300–308.

[11] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning for fast probabilistic diffeomorphic registration," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018, pp. 729–738.

[12] X. Hu, M. Kang, W. Huang, M. R. Scott, R. Wiest, and M. Reyes, "Dual-stream pyramid registration network," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2019, pp. 382–390.

[13] S. Zhou *et al.*, "Fast and accurate electron microscopy image registration with 3D convolution," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2019, pp. 478–486.

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

[15] S. Zhao, Y. Dong, E. Chang, and Y. Xu, "Recursive cascaded networks for unsupervised medical image registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10600–10610.

[16] S. Zhao, T. Lau, J. Luo, I. Eric, C. Chang, and Y. Xu, "Unsupervised 3D end-to-end medical image registration with volume tweening network," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 5, pp. 1394–1404, May 2020.

[17] M. Q. Tran, T. Do, H. Tran, E. Tjiputra, Q. D. Tran, and A. Nguyen, "Light-weight deformable registration using adversarial learning with distilling knowledge," *IEEE Trans. Med. Imag.*, early access, Jun. 6, 2022, doi: 10.1109/TMI.2022.3141013.

[18] R. Bajcsy and S. Kovačič, "Multiresolution elastic matching," *Comput. Vis., Graph., Image Process.*, vol. 46, no. 1, pp. 1–21, 1989.

[19] D. Shen and C. Davatzikos, "HAMMER: Hierarchical attribute matching mechanism for elastic registration," *IEEE Trans. Med. Imag.*, vol. 21, no. 11, pp. 1421–1439, 2002.

[20] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: Application to breast MR images," *IEEE Trans. Med. Imag.*, vol. 18, no. 8, pp. 712–721, Aug. 1999.

[21] J.-P. Thirion, "Image matching as a diffusion process: An analogy with Maxwell's demons," *Med. Image Anal.*, vol. 2, no. 3, pp. 243–260, 1998.

[22] D. Rueckert, P. Aljabar, R. A. Heckemann, J. V. Hajnal, and A. Hammers, "Diffeomorphic registration using B-splines," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2006, pp. 702–709.

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

[24] J. Glaunès, A. Qiu, M. I. Miller, and L. Younes, "Large deformation diffeomorphic metric curve mapping," *Int. J. Comput. Vis.*, vol. 80, no. 3, pp. 317–336, 2008.

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

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

[27] J. Krebs *et al.*, "Robust non-rigid registration through agent-based action learning," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2017, pp. 344–352.

[28] N. Gunnarsson, J. Sjölund, and T. B. Schön, "Learning a deformable registration pyramid," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2020, pp. 80–86.

[29] B. Hu, S. Zhou, Z. Xiong, and F. Wu, "Self-recursive contextual network for unsupervised 3D medical image registration," in *Proc. Int. Workshop Mach. Learn. Med. Imag.* Cham, Switzerland: Springer, 2020, pp. 60–69.

[30] K. A. J. Eppenhof, M. W. Lafarge, M. Veta, and J. P. W. Pluim, "Progressively trained convolutional neural networks for deformable image registration," *IEEE Trans. Med. Imag.*, vol. 39, no. 5, pp. 1594–1604, May 2020.

[31] T. C. Mok and A. C. Chung, "Large deformation diffeomorphic image registration with Laplacian pyramid networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2020, pp. 211–221.

[32] J. Fan, X. Cao, P.-T. Yap, and D. Shen, "BIRNet: Brain image registration using dual-supervised fully convolutional networks," *Med. Image Anal.*, vol. 54, pp. 193–206, May 2019.

[33] Z. Shen, X. Han, Z. Xu, and M. Niethammer, "Networks for joint affine and non-parametric image registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4224–4233.

[34] T. C. Mok and A. Chung, "Conditional deformable image registration with convolutional neural network," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2021, pp. 35–45.

[35] A. Hoopes, M. Hoffmann, B. Fischl, J. Guttag, and A. V. Dalca, "Hypermorph: Amortized hyperparameter learning for image registration," in *Proc. Int. Conf. Inf. Process. Med. Imag.* Cham, Switzerland: Springer, 2021, pp. 3–17.

[36] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[37] R. Adriana, B. Nicolas, K. S. Ebrahimi, C. Antoine, G. Carlo, and B. Yoshua, "Fitnets: Hints for thin deep nets," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015, pp. 1–13.

[38] N. Komodakis and S. Zagoruyko, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2017, pp. 1–10.

[39] L. Qi *et al.*, "Multi-scale aligned distillation for low-resolution detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14443–14453.

[40] K. Zhang, C. Zhang, S. Li, D. Zeng, and S. Ge, "Student network learning via evolutionary knowledge distillation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2251–2263, Apr. 2022.

[41] X. Xing, Y. Hou, H. Li, Y. Yuan, H. Li, and M. Q.-H. Meng, "Categorical relation-preserving contrastive knowledge distillation for medical image classification," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2021, pp. 163–173.

[42] D. Qin *et al.*, "Efficient medical image segmentation based on knowledge distillation," *IEEE Trans. Med. Imag.*, vol. 40, no. 12, pp. 3820–3831, Dec. 2021.

[43] H. Yu, X. Chen, H. Shi, T. Chen, T. S. Huang, and S. Sun, "Motion pyramid networks for accurate and efficient cardiac motion estimation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2020, pp. 436–446.

[44] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[45] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 28, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Dec. 2015, pp. 1–8. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/33ceb07bf4eeb3da587e268d663aba1a-Paper.pdf

[46] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults," *J. Cogn. Neurosci.*, vol. 19, no. 9, pp. 1498–1507, Sep. 2007.

[47] Dalca. *Oasis*. [Online]. Available: https://github.com/adalca/medical-datasets/blob/master/neurite-oasis.md

[48] A. Klein and J. Tourville, "101 labeled brain images and a consistent human cortical labeling protocol," *Frontiers Neurosci.*, vol. 6, p. 171, Dec. 2012.

[49] D. W. Shattuck *et al.*, "Construction of a 3D probabilistic atlas of human cortical structures," *NeuroImage*, vol. 39, no. 3, pp. 1064–1080, Feb. 2008.

[50] 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.

[51] A. D. Martino *et al.*, "The autism brain imaging data exchange: Towards a large-scale evaluation of the intrinsic brain architecture in autism," *Mol. Psychiatry*, vol. 19, no. 6, pp. 659–667, Jun. 2014.

[52] P. Bellec, C. Chu, F. Chouinard-Decorte, Y. Benhajali, D. S. Margulies, and R. C. Craddock, "The neuro bureau ADHD-200 preprocessed repository," *NeuroImage*, vol. 144, pp. 275–286, Jan. 2017.

[53] T. Heimann *et al.*, "Comparison and evaluation of methods for liver segmentation from CT datasets," *IEEE Trans. Med. Imag.*, vol. 28, no. 8, pp. 1251–1265, Aug. 2009.

[54] MSD. *Medical Segmentation Decathlon*. [Online]. Available: http://medicaldecathlon.com/
[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[56] B. B. Avants, N. Tustison, and G. Song, "Advanced normalization tools (ANTs)," *Insight J.*, vol. 2, pp. 1–35, Jun. 2009.
[57] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim, "Elastix: A toolbox for intensity-based medical image registration," *IEEE Trans. Med. Imag.*, vol. 29, no. 1, pp. 196–205, Jan. 2010.

Bo Hu received the B.S. degree in electronic information engineering from the University of Electronic Science and Technology of China in 2019. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China. His research interest includes biomedical image processing.

Shenglong Zhou received the B.S. degree in electronic information engineering from the Hefei University of Technology in 2018. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China. His research interests include image processing and computer vision.

Zhiwei Xiong (Member, IEEE) received the B.S. and Ph.D. degrees in electronic engineering from the University of Science and Technology of China (USTC) in 2006 and 2011, respectively. He has been a Professor with USTC since 2016. Before that, he was a Researcher with Microsoft Research Asia (MSRA). He has authored or coauthored more than 100 papers in premium journals and conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MEDICAL IMAGING, IEEE TRANSACTIONS ON IMAGE PROCESSING, CVPR, ICCV, and ECCV. His research interests include computational photography, low-level vision, and biomedical image analysis. He received the Best Paper Award of the 2016 IEEE VCIP and the 2009 MSRA Fellowship. He and his students were the winners of eight technical challenges held in CVPR, ICCV, ECCV, MM, ICME, and ISBI.

Feng Wu (Fellow, IEEE) received the B.S. degree in electronic engineering from Xidian University in 1992 and the M.S. and Ph.D. degrees from the Harbin Institute of Technology in 1996 and 1999, respectively. He joined Microsoft Research Asia (MSRA) in 1999, where he is working as an Associate Researcher, a Researcher, and a Principle Researcher. He joined the University of Science and Technology of China (USTC) in 2014, where he is currently the Vice President, the Chair of the Division of Information and Intelligence, and the Director of the National Engineering Laboratory for Brain-Inspired Intelligence Technology and Application (NEL-BITA). He has authored or coauthored over 450 high quality papers (including more than 190 journal articles) and top conference papers on MOBICOM, SIGIR, CVPR, and ACM MM. He has 120 granted U.S. patents. His 15 techniques have been adopted into international video coding standards. His work in Google Scholar has been cited more than 19900 (H-index as 69) to date. His research interests include image and video compression, media communication, and media analysis and synthesis. As a coauthor, he got the Best Paper Award in the 2009 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the 2016 IEEE VCIP, the 2008 PCM, and the 2007 SPIE VCIP. He received the Mac Van Valkenburg Award, which is the most prestigious awards in IEEE CAS Society. He has gained international reputation in the field of media streaming. He is the Editor-in-Chief of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, which is a top-tier journal in the video processing area and the Chair of IEEE Data Compression Standard Committee (IEEE DCSC).