Efficient Large-Deformation Medical Image Registration via Recurrent Dynamic Correlation

Tianran Li, Marius Staring, Yuchuan Qiao

***Abstract*— Deformable image registration estimates voxel-wise correspondences between images through spatial transformations, and plays a key role in medical imaging. While deep learning methods have significantly reduced runtime, efficiently handling large deformations remains a challenging task. Convolutional networks aggregate local features but lack direct modeling of voxel correspondences, promoting recent works to explore explicit feature matching. Among them, voxel-to-region matching is more efficient for direct correspondence modeling by computing local correlation features within neighbourhoods, while region-to-region matching incurs higher redundancy due to excessive correlation pairs across large regions. However, the inherent locality of voxel-to-region matching hinders the capture of long-range correspondences required for large deformations. To address this, we propose a Recurrent Correlation-based framework that dynamically relocates the matching region toward more promising positions. At each step, local matching is performed with low cost, and the estimated offset guides the next search region, supporting efficient convergence toward large deformations. In addition, we uses a lightweight recurrent update module with memory capacity and decouples motion-related and texture features to suppress semantic redundancy. We conduct extensive experiments on brain MRI and abdominal CT datasets under two settings: with and without affine pre-registration. Results show our method exhibits a strong accuracy-computation trade-off, surpassing or matching the state-of-the-art performance. For example, it achieves comparable performance on the non-affine OASIS dataset, while using only 9.5% of the FLOPs and running 96% faster than RDP, a representative high-performing method.**

***Index Terms*— Large deformation, deformable image registration, unsupervised deep learning, recurrent dynamic correlation**

I. Introduction

DEFORMABLE medical image registration aligns 3D image pairs (named the fixed image I_f and the moving image I_m) by applying spatial transformations to match their structures within a common coordinate system [1]. It involves mapping each voxel in I_f to a corresponding most relevant location in I_m. This process is crucial in clinical applications like tumor monitoring and CT/MRI/PET data fusion [2]. Traditionally, it involves iterative optimization to find the optimal deformation parameters with extensive and time-consuming searches [1]. Although GPU acceleration has made iterative registration much faster, these methods are still sensitive to starting conditions during optimization.

In contrast, deep learning methods could complete registration in seconds by estimating the deformation field from scratch [3]. However, efficiently handling large deformations remains challenging. U-shaped convolutional structures which directly predict the deformation field in an end-to-end manner [4, 5], struggle with large deformations due to the limited receptive fields. To mitigate this limitation, pyramid [6, 7], recursive [8] and hybrid architectures [9, 10] are later introduced, which decompose large deformations into smaller and more manageable steps. Although they have been proven to be effective to some extent for large deformations, the convolution operations are fundamentally the weighted aggregation within neighborhoods, which is less capable to model voxel-wise correspondences between the two images.

Several recent works introduce explicit feature matching by establishing voxel-wise dense matches within a specified region, a process that naturally aligns with the core objective of the registration task: identifying voxel-level correspondences between fixed and moving images. In this sense, such process can be viewed as *a search within specific regions to identify the most relevant target coordinates*. However, existing methods still struggle to balance accuracy and computational efficiency.

One group of these methods [11, 12, 13, 14, 15] adopts a ***region-to-region*** matching strategy, typically implemented via the attention mechanism in Transformer [8, 16], which densely computes pairwise voxel correlations within large regions of image pair. However, many of these voxel pairs are irrelevant due to the high structural similarity in medical images [17], resulting in substantial computational redundancy. In contrast, another line of work [18, 19] adopts ***voxel-to-region*** matching, where each voxel in I_f independently queries its corresponding regions in I_m. Compared to *region-to-region* strategies that correlate all voxels within a window, *voxel-to-region* matching assigns each voxel a specific search scope, reducing unnecessary pairwise correlation operations. Typically, to keep the computational cost manageable, the region in I_m is confined to a small local neighborhood, which is often insufficient to capture very large deformations. CorrMLP [19] attempts to mitigate this by applying multiple large-window MLPs, but at a cost of significantly increased computation. In addition, correlation features are often treated as auxiliary channels concatenated with the two image features [18, 19], which introduces additional semantic redundancy and reduces their effectiveness in guiding correspondence search.

This work is supported by the National Natural Science Foundation of China under Grant 82102002. Corresponding author: Yuchuan Qiao. Email: YuchuanQiao@fudan.edu.cn.

Tianran Li and Yuchuan Qiao are with the Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai, China. Marius Staring is with the Department of Radiology, Leiden University Medical Center, Leiden, The Netherlands.

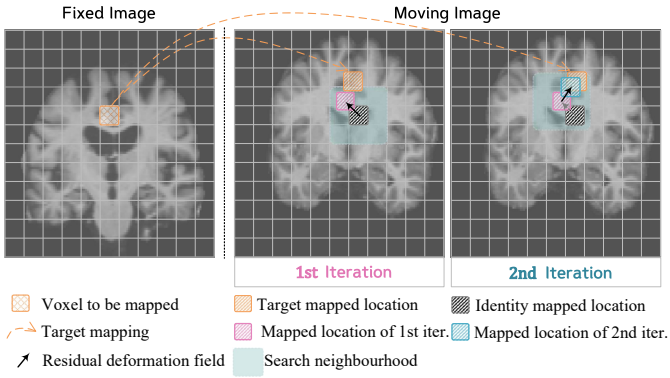

Fig. 1. *Illustration of the recurrent local search strategy of ReCorr.* A voxel from the fixed image (left) is progressively matched to its corresponding location in the moving image via recurrent local search and dynamic updates. At each iteration, the search is performed within a local neighborhood (blue), and the deformation field is incrementally updated (black arrows), guiding the search region toward more accurate correspondences.

In this work, we rethink how large deformations are modeled, and explore to efficiently establish long-range voxel correspondences. To overcome the locality of the *voxel-to-region* strategy while still leveraging its computational efficiency, we introduce an efficient Recurrent Correlation-based framework, named ReCorr, which performs dynamic local search guided by iterative search-center relocation. The basic idea is illustrated in Fig. 1. At each step, the model conducts *voxel-to-region* matching within a local neighborhood, then dynamically updates the search center to guide the next iteration. Through this recurrent process, the search windows adaptively shift toward more promising matching regions and gradually converge to the global optimal correspondence, with each step at a low computational cost. This search scheme is inspired by gradient-based optimization, where parameters are incrementally refined toward convergence.

Besides, we design a lightweight recurrent update module with memory capacity that retains useful deformation context across iterations. To help the model focus on spatial alignment without interference from redundant semantic features, instead of directly concatenating all features, we decouple the prediction into two branches: one for the motion-related information and the other for the image texture. Our module operates at a single resolution with shared parameters, reducing the inference time compared to previous iterative methods [8, 9] which jointly use multi-scale image features. ReCorr adopts a pyramid architecture, where iterative search begins at a coarse resolution to provide a reliable initialization for finer levels. Unlike traditional coarse-to-fine schemes, the iterations are not limited to the number of pyramid levels, which effectively reduces low-resolution errors by allowing continuous refinement at each scale.

To evaluate our approach for both small and large deformations, we conducted experiments in two scenarios: *regular* experiments on datasets with affine pre-registration, and *extreme* trials on datasets without such pre-registration, formulated to present more complex and challenging deformations. Across all experimental setups on two brain MRI datasets and an abdominal CT dataset, our method consistently outperforms or compares favorably with the state-of-the-art methods but at significantly lower FLOPs and inference time.

In summary, this work has the following contributions:

1) Efficient Search Scheme for Large Deformations: We propose ReCorr with an efficient search scheme for large deformations by dynamically performing voxel-to-region matching with iterative search-center relocation. The pyramid recurrent process compensates for the spatial limitation of voxel-to-region matching, enabling progressive convergence toward the global optimum at a low computational cost per step.

2) Lightweight Recurrent Update Module with Motion-texture Decoupling: A lightweight update module is introduced to retain historical deformation cues via memory across iterations. To suppress semantic redundancy and focus on alignment, we decouple the prediction into two branches for motion-related information and image texture.

3) Excellent Accuracy-computation Trade-off on Diverse Deformation scenarios: We conduct extensive evaluations on both regular (affine pre-registered) and extreme (unregistered) settings across two brain MRI datasets and one abdominal CT dataset. Our method consistently achieves superior or competitive accuracy compared to state-of-the-art approaches, while significantly reducing FLOPs and inference time.

We review related work in Section II, and present our method in Section III, including the problem definition, overall architecture, local search module, and recurrent updater. Section IV describes the experimental setup, and Section V reports the results and analysis. Finally, discussions and conclusions are provided in Section VI. Our code is available at *https://github.com/YQiaoGroup/Registration-ReCorr*.

II. Related Work

A. Non-learning Based Medical Image Registration

Image registration has traditionally been approached as an optimization problem, balancing between a similarity term and a regularization term. In the early stages, the methods include elastic registration methods [20], B-spline based free-form deformations [21], and the Demons algorithm [22]. Later, there are several techniques focusing on diffeomorphic transformations for anatomical accuracy, such as the diffeomorphic Demons [23], Large Deformation Diffeomorphic Metric Mapping (LDDMM) [24] and Symmetric Normalization (SyN) [25]. These methods, while effective and robust, tend to be time-intensive because of the time-consuming and resource-intensive optimization. With the adoption of GPU acceleration, many iterative methods have become faster, but they still remain sensitive to initialization and prone to local optima.

B. Learning Based Medical Image Registration

Pure Convolutional Networks. Recently, learning-based methods have demonstrated the potential for fast and high-precision image registration [26, 27]. A milestone is VoxelMorph [4], which introduced an unsupervised U-shaped convolutional framework for predicting deformation fields from image pairs, inspiring numerous subsequent approaches. MIDIR [5] use a similar structure to VoxelMorph but modifies the output to be the parameters of the B-spline transformation.

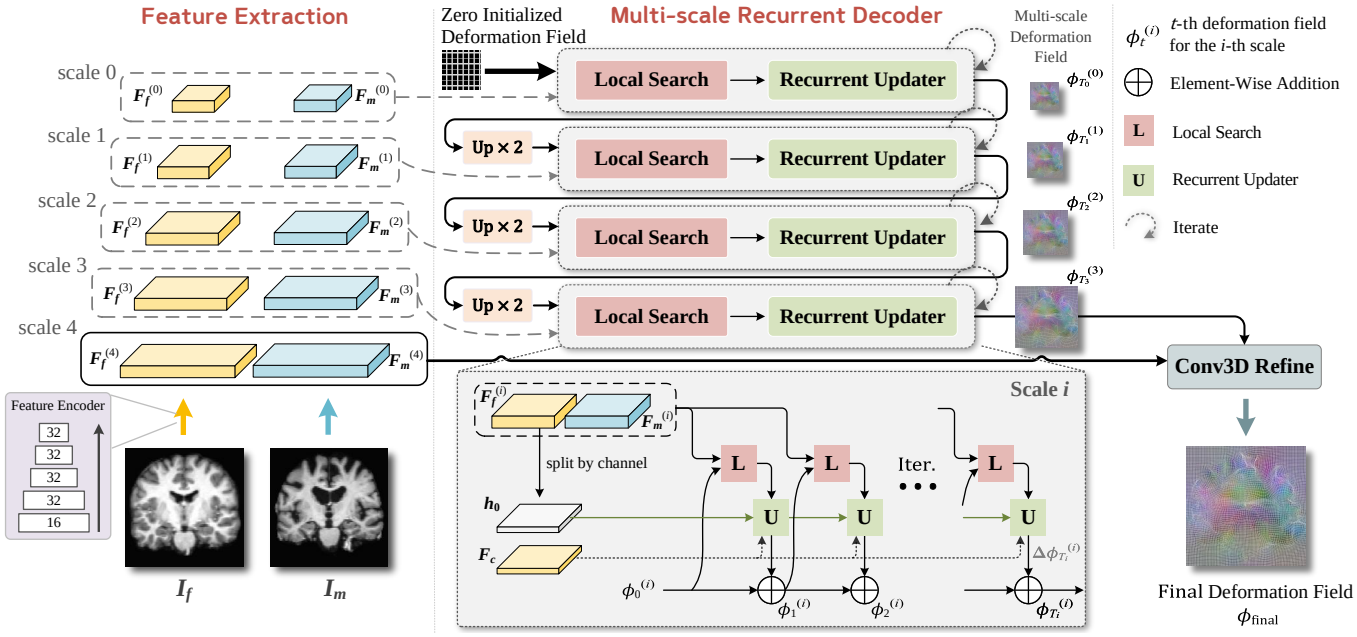

Fig. 2. Architecture overview of the proposed ReCorr. Given a pair of images, ReCorr first extracts multi-scale features using a same encoder, then performs iterative refinement from scale $\mathbf{0}$ to scale $\mathbf{3}$. At each scale i, the t-th iteration begins by conducting local search and then updates the residual deformation field $\Delta\phi_t^{(i)}$, which is added to the deformation field from the previous iteration. Except for the zero-initialized scale 0, each scale is initialized by upsampling the final output from the preceding scale. The final deformation field ϕ_{final} is refined at the original resolution using a Conv3D block.

These methods build on pure convolutional operations and predict dense deformation fields in a single pass. While some other work attempt to improve accuracy by enhancing feature representations [28, 29, 30] or reducing redundant parameters [31], they still face challenges with large deformations due to the inherently limited receptive fields. To tackle large deformations, some methods adopt progressive refinement via recursive or pyramid structures. VTN [32] stacks multiple U-Net-based subnetworks, while LapIRN [6] introduces a coarse-to-fine pyramid strategy that has become widely adopted. Other variants of pyramid-based methods include [7, 33, 34, 35]. Several works combine the pyramid structure with recurrent refinement. SDHNet [9] jointly estimates and fuses multi-scale deformation fields at each iteration, but its parallel design increases structural complexity and inference latency due to the need for synchronization across levels. In contrast, RDP [10] applies recurrent loops sequentially across pyramid levels, yet suffers from significant computational overhead due to repeated feature concatenation and heavy 3D convolutions. IIRP [36] follows a similar strategy but introduces additional cost by performing NCC-based early stopping at each iteration. However, these methods are pure convolutional architectures, which are insufficient to capture voxel-correspondences.

Region-to-region Explicit Feature Matching. Recent works introduce explicit voxel-wise matching to establish dense correspondences. A common strategy is region-to-region matching, where pairwise voxel correlations are densely computed between two predefined regions. The pioneering work [11] introduces a 6D correlation feature computed over the whole image pair. Later, the rise and popularity of Transformer architectures have led to methods such as [12, 13, 14], which adopt Transformer-based architectures for voxel-wise correlation modeling, typically within partitioned windows of the input images. Inspired by these methods, CGNet [15] introduces a modified correlation module as an alternative to the attention mechanism in Transformers, enabling efficient processing of high-resolution features. However, due to the relatively consistent contextual positions of structures in medical images [17], many voxel pairs within the window are irrelevant, leading to significant computational redundancy.

Voxel-to-region Explicit Feature Matching. Several other methods adopt a more efficient voxel-to-region strategy, where each voxel queries a local neighborhood in the other image to identify the best match, reducing redundancy by avoiding exhaustive pairwise voxel correlations. However, their locality limits the ability to capture large deformations. DualPRNet++ [33] incorporates the local correlation features as auxiliary information in a dual-stream pyramid network to enhance deformable image registration. CorrMLP [19] combines the correlation features with multi-window MLPs to capture a broader receptive field, but the fully-connected nature of MLPs leads to considerable computational overhead, especially at high resolutions. Notably, these methods treat correlation features merely as supplementary cues, simply concatenating them with image features within a feature-based framework. In contrast, we highlight the role of local correlation in explicitly modeling voxel-wise spatial relationships and decouple the update process into two branches: one for motion-related information and the other for image texture, to better focus on alignment and reduce semantic redundancy.

While our method is connected to the existing methods, our motivation is originated from the efficiency–accuracy trade-off challenges for large-deformation registration and differs substantially in design. We explicitly leverage the voxel-to-

region matching paradigm for its direct role in establishing voxel correspondences, and address its locality through pyramid recurrent refinement. This enable the model to perform dynamic local search, with search centers progressively shifting toward more promising regions, achieving accurate long-range matching at low computational cost. In addition, by decoupling prediction into motion and texture branches, our model focuses on spatial alignment without being distracted by semantic redundancy, improving both precision and efficiency.

III. Method

A. Definition

Given a moving image $I_m \in \mathbb{R}^3$ and a fixed image $I_f \in \mathbb{R}^3$, deformable image registration seeks a dense, non-linear transformation $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ from I_m to I_f, such that the deformed image $I'_m = I_m \circ \phi$ is as similar as possible to I_f. The deformation field ϕ defines voxel-wise correspondences by mapping each voxel in I_f to a corresponding location in I_m. In this context, handling large deformations means establishing long-range voxel correspondences across the two images.

B. Architecture of ReCorr

To efficiently handle large deformations in medical image registration, we propose a recurrent correlation-based framework specifically designed to establish long-range voxel correspondences with low computational cost. Our method leverages the voxel-to-region matching paradigm and decomposes the large distance into several shorter steps through a pyramid recurrent refinement strategy. At each level, the model iteratively performs dynamic local search by matching voxels in I_f to local neighborhoods in I_m, with search centers progressively updated toward more likely correspondences, thus effectively bridging the spatial gap. An overview of the proposed framework is given in Fig. 2.

Given a pair of input images I_f and I_m, we use a lightweight encoder akin to that in VoxelMorph [4] to extract multi-scale features, as shown in Fig. 2 *(left)*. This encoder is shared by the two images and generates two sets of features at scales of $1/16$, $1/8$, $1/4$, $1/2$, and the original resolution, represented as $\{\boldsymbol{F}_f^{(i)}\}_{i=0,1,2,3,4}$ and $\{\boldsymbol{F}_m^{(i)}\}_{i=0,1,2,3,4}$, respectively. Benefiting from the inductive bias of CNNs in translation equivariance and locality, these features capture local structures and texture patterns, which are suitable for voxel-wise matching. Features at $1/16$, $1/8$, $1/4$ and $1/2$ resolutions ($\{\boldsymbol{F}_f^{(i)}\}_{i=0,1,2,3}$) are used for iterative updates, while features at original resolution ($\boldsymbol{F}_m^{(4)}$) are used for final refinement. A portion of the features from the fixed image, denoted as $\boldsymbol{F}_c$, is used to preserve image texture and serves as context guidance during deformation estimation (see Section III-D for details). Note that the feature extraction process is executed only once and the extracted features are consistently reused across all iterations for efficiency.

Starting from scale 0, ReCorr performs iterative updates sequentially up to scale 3. The procedures at all the four scales are the same, and we take scale i ($i = 0, 1, 2, 3$) as an example. At scale i, there are T_i iterations. During the t-th iteration, we perform local search by querying each voxel in I_f within a local neighborhood centered at its previously estimated mapped location in I_m, resulting a dynamic 4D correlation feature $\mathcal{C}$. Subsequently, $\mathcal{C}$ is fed into a lightweight recurrent updater to produce the residual deformation field $\Delta\phi_t^{(i)}$, and the current deformation field $\phi_t^{(i)}$ is updated as

$$\phi_t^{(i)} = \phi_{t-1}^{(i)} + \Delta\phi_t^{(i)}, \ t = 1, 2, ..., T_i. \tag{1}$$

As a result, the final deformation field of the i-th scale $\phi_T^{(i)}$ is the accumulation of all learned $\Delta\phi^{(i)}$:

$$\phi_T^{(i)} = \phi_0^{(i)} + \sum_{t=1}^{T} \Delta\phi_t^{(i)}, \ t = 1, 2, ..., T_i, \tag{2}$$

where $\phi_0^{(i)}$ denotes the initial deformation field at scale i. At scale 0, $\phi_0^{(0)}$ is zero-initialized for identical mapping, while at higher scales, $\phi_0^{(i)}$ is derived by upsampling the final deformation field from the preceding scale with a factor of 2. Finally, after all iterations are completed, the deformation field is upsampled to full image resolution and refined using the high-resolution $\boldsymbol{F}_f^{(4)}$ and $\boldsymbol{F}_m^{(4)}$ to recover fine-grained details, yielding in the final deformation field ϕ_{final}.

C. Local Search Module

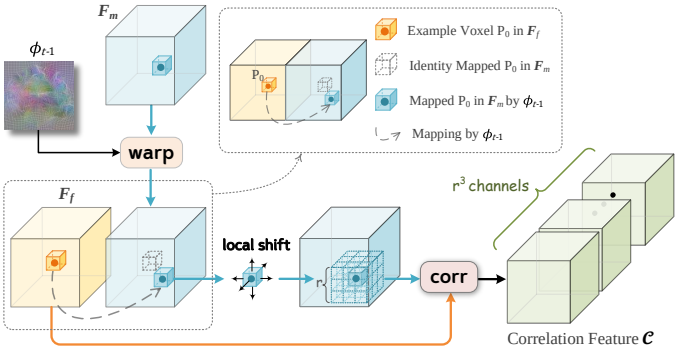

Fig. 3. The structure of the local search module. Note that we omit the scale index 'i' for simplicity.

Motivated by [18], ReCorr uses voxel-to-region local correlation to explicitly model voxel correspondences, in a manner that closely aligning with the goal of registration. This operation is referred to as *local search*, as each voxel in I_f queries within a restricted neighborhood in I_m for the best match. It takes as input the extracted features $\boldsymbol{F}_f^{(i)}$ and $\boldsymbol{F}_m^{(i)}$ (i=0,1,2,3), along with the previous deformation field $\phi_{t-1}^{(i)}$, to dynamically construct a lightweight 4D correlation feature. For simplicity, the scale index 'i' is omitted in the following descriptions. Both $\boldsymbol{F}_f$ and $\boldsymbol{F}_m$ are assumed to have the shape $B \times C \times D \times H \times W$, where B is the batch size, C is the number of channels, and D, H, W denote the spatial dimensions.

As illustrated in Fig. 3, the computation of local search during the t-th iteration involves three steps:

1) Global Localization: $\boldsymbol{F}_m$ is warped using ϕ_{t-1} to obtain the roughly mapped global locations.

2) Local Shift: Centered around the identified mapped locations, cube-shaped local neighbourhoods with a side r voxels are extracted by applying padding and spatial shifting. Specifically, the warped feature $\boldsymbol{F}_m$ is padded and then shifted

along the x, y and z axes within the range $[-r//2, r//2]$, producing a total of r^3 offsets positions.

3) Correlation and Aggregation: For each shift position, a correlation volume $\boldsymbol{Corr}$ is computed between $\boldsymbol{F}_f$ and the corresponding shifted, warped $\boldsymbol{F}_m$ using the normalized dot product of feature vectors:

$$\boldsymbol{Corr}_{bzyx} = \frac{1}{C}\left\langle (\boldsymbol{F}_f)_{b\cdot zyx}\,,\ \mathbf{Shift}(\boldsymbol{F}_m \circ \phi_{t-1})_{b\cdot zyx}\right\rangle, \quad (3)$$

where b, z, y, x index the batch and spatial dimensions B, D, H, W, respectively. The resulting $\boldsymbol{Corr}$ has shape $B \times D \times H \times W$. All correlation volumes across the r^3 offsets are then concatenated along the channel dimension to form a 4D correlation feature $\boldsymbol{\mathcal{C}} \in B \times r^3 \times D \times H \times W$. The hyper-parameter r is set to 3 based on ablation studies. A local neighbourhood with side length r at coarse scales corresponds to a much larger region in the original resolution space, enabling broader and more efficient search. In contrast, higher-scales focus the search within smaller regions, allowing more precise voxel-level matching and refinement.

The computational complexity of local correlation is $\mathcal{O}(C \cdot r^3 \cdot DHW)$. Compared to convolutional operations, which implicitly aggregate local information with complexity $\mathcal{O}(C \cdot C_{out} \cdot k^3 \cdot DHW)$ (C, and C_{out} are the input and output channel dimensions), local correlation offers both direct correspondence modeling and lower cost. Compared to region-to-region matching, which also performs explicit feature matching but requires dense correlations between all voxel pairs across two regions, resulting in much higher cost at $\mathcal{O}(C \cdot (DHW)^2)$ if applied to the full volume [8]. Even window-based variants [16, 12] that limit matching to partitioned subregions still incur substantial overhead, particularly in 3D where the number of windows grows cubically with resolution.

Overall, local correlation is both task-aligned and computationally efficient, but its performance is limited by the narrow search range. To address this, we introduce a pyramid recurrent strategy that gradually guides the search toward more accurate correspondences, while maintaining low computational cost.

D. Recurrent Updater

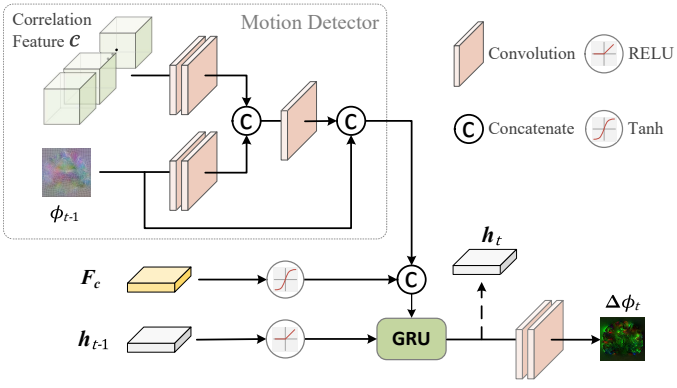

Fig. 4. The structure of the recurrent updater for each scale.

We propose a lightweight recurrent updater to iteratively update the deformation field, which in turn guides the next iteration. To focus on spatial alignment, the prediction is decoupled into motion and texture branches.

At the t-th iteration of scale i, the updater predicts a residual deformation field $\Delta\phi_t^{(i)}$ based on the previous estimate $\phi_{t-1}^{(i)}$. All iterations at the same scale share the same update module. For clarity, we omit the scale index 'i' in the following.

As show in Fig. 4, the core of the updater is a gated recurrent unit (GRU) [37], which maintains a hidden state to propagate deformation-related cues across iterations. The correlation feature $\boldsymbol{\mathcal{C}}$ and the previous deformation field ϕ_{t-1}, which both reflect spatial relationships between the fixed and moving images, are treated as motion-related inputs and processed by a motion detector to produce $\boldsymbol{m}_t$.

Meanwhile, the fixed feature map $\boldsymbol{F}_f$ is split along the channel dimension: one half initializes the GRU hidden state $\boldsymbol{h}_0$, and the other is used as the contextual guidance $\boldsymbol{F}_c$ to preserve image texture. Then, the motion feature $\boldsymbol{m}_t$, the activated context $\boldsymbol{F}_c$, and the previous hidden state $\boldsymbol{h}_{t-1}$ are concatenated and fed into the GRU. The updated hidden state $\boldsymbol{h}_t$ is computed as follows:

$$\begin{aligned}
\boldsymbol{z_t} &= \sigma\big(\mathrm{Conv}([\boldsymbol{h}_{t-1}, \boldsymbol{m}_t, \boldsymbol{F}_c], \boldsymbol{W}_z)\big),\\
\boldsymbol{r}_t &= \sigma\big(\mathrm{Conv}([\boldsymbol{h_{t-1}}, \boldsymbol{m}_t, \boldsymbol{F}_c], \boldsymbol{W}_r)\big),\\
\tilde{\boldsymbol{h}}_t &= \tanh\big(\mathrm{Conv}([\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}, \boldsymbol{m}_t, \boldsymbol{F}_c], \boldsymbol{W}_h)\big),\\
\boldsymbol{h}_t &= (1 - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t.
\end{aligned} \quad (4)$$

Inspired by [38], the convolution operators within the GRU unit are designed to be separable, implemented as sequential GRU branches with $1 \times 1 \times 5$, $1 \times 5 \times 1$, and $5 \times 1 \times 1$ convolutions. This design enlarges the receptive field without significantly increasing the model complexity. Finally, the updated hidden state $\boldsymbol{h}_t$ is passed through a convolution layer to produce the residual deformation field $\Delta\phi_t$.

E. Unsupervised Loss Function

ReCorr produces T_i deformation fields $\{\phi_1^{(i)}, \phi_2^{(i)}, \ldots, \phi_T^{(i)}\}$ at each scale i $(i = 0, 1, 2, 3)$. Together with the full-resolution refined result ϕ_{final}, the complete sequence of predicted deformation fields is as follows:

$$\{\{\phi_1^{(0)}, \ldots, \phi_T^{(0)}\}, \cdots, \{\phi_1^{(3)}, \ldots, \phi_T^{(3)}\}, \phi_{\text{final}}\}.$$

For loss computation, all deformation fields are upsampled to the original resolution, yielding a unified sequence $\{\phi_1, \phi_2, \ldots, \phi_t, \ldots, \phi_T\}$, where $T = T_0 + T_1 + T_2 + T_3 + 1$. Each ϕ_t is supervised using two terms: an image similarity loss and a deformation field regularization loss, following [4]. The combined loss function is denoted as $\mathcal{L}_{\text{single}}$, defined as:

$$\mathcal{L}_{\text{single}}(I_f, I_m, \phi_t) = \mathcal{L}_{\text{sim}}(I_f, I_m \circ \phi_t) + \lambda \mathcal{L}_{\text{reg}}(\phi_t), \quad (5)$$

where $\mathcal{L}_{\text{sim}}$ denotes the similarity loss, implemented using either mean squared intensity error (MSE) or normalized cross-correlation (NCC), and $\mathcal{L}_{\text{reg}}$ is an L2 penalty on the spatial gradients of ϕ_t. The hyperparameter λ balances the trade-off of two terms .

To ensure effective supervision across multiple outputs, we apply exponentially increasing weights to the sequence $\{\phi_1, \phi_2, \ldots, \phi_t, \ldots, \phi_T\}$, following [39]:

$$\mathcal{L} = \sum_{t=1}^{T} \gamma^{T-t} \mathcal{L}_{\text{single}}(I_f, I_m, \phi_t), \quad (6)$$

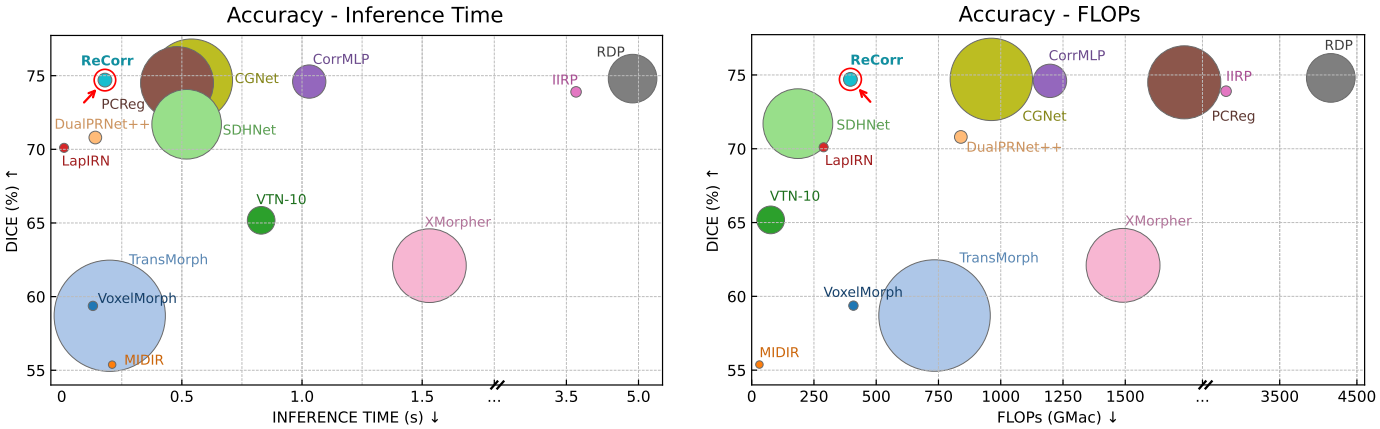

Fig. 5. The two scatter plots compare registration accuracy (Dice) against inference time (left), and FLOPs (right) on the OASIS dataset without affine pre-registration. Circle size represents model parameters, where a **larger** circle indicates **more** parameters. Arrows (↑ ↓) indicate the preferable direction of metrics. The proposed ReCorr achieves a favorable trade-off between accuracy and efficiency in both views.

where $\gamma \in (0, 1]$ and is set to 0.7 in our experiments. This strategy assigns lower weights to early predictions and progressively increases the weights for later ones, ensuring greater emphasis on the later stages of deformation field prediction, which contribute more to the final result.

F. Diffeomorphic Version of ReCorr

In addition to the standard version, we offer a diffeomorphic variant of ReCorr, where the network predicts residual velocity fields at each iteration. To perform local search in this setting, several modifications are required: the velocity field is first transformed into a deformation field using the scaling and squaring algorithm [40] with a time step of 5, after which $\boldsymbol{F}_m$ is warped accordingly. The final deformation field is also obtained through the scaling and squaring applied to the accumulated velocity field.

IV. Experiments

A. Datasets

The evaluation is performed on 3 datasets: two brain MRI datasets OASIS [41] and IXI[1], and an abdominal CT dataset BTCV [42]. We conduct atlas-based experiments on the OASIS and the IXI datasets, and subject-to-subject experiments on the BTCV dataset. To comprehensively evaluate the effectiveness of ReCorr for both *regular* and *extremely large* deformations, dual-setting experiments are conducted on the OASIS dataset: one with affine pre-registration and the other without. Besides, the IXI dataset preprocessed by TransMorph [12] (including affine pre-registration) is used to test *regular* deformations, while the BTCV dataset without affine pre-registration is used to test *extremely substantial* deformations.

The OASIS dataset includes 414 scans with segmentation annotations on 35 brain regions. One scan is randomly selected as the atlas, and the rest are split into 330 for training, 28 for validation, 55 for testing. For the two experiment settings (with and without affine pre-registration), different preprocessing steps are performed after the skull stripping. Specifically, the OASIS (affine) scans are resampled to a size of $256 \times 256 \times 256$ with 1mm isotropic voxels, then affine pre-registerd and cropped to $160 \times 224 \times 192$. In contrast, the OASIS (non-affine) scans are directly resampled to $192 \times 192 \times 192$ with 1.33 mm isotropic voxels.

The IXI dataset includes 576 T1-weighted brain MRI im-

TABLE I

COMPARISON RESULTS ON THE BTCV DATASET WITHOUT AFFINE PRE-REGISTRATION. MSE SERVES AS THE SIMILARITY LOSS FOR ALL METHODS. ARROWS (↑, ↓) INDICATE THE PREFERABLE DIRECTION OF METRICS. THE STANDARD DEVIATION FOR EACH METRIC IS SHOWN IN PARENTHESES. **BOLD** FONT INDICATES THE BEST VALUE, UNDERLINE FONT INDICATES THE SECOND-BEST VALUE. SUPERSCRIPTS * AND † DENOTE DIFFERENT WILCOXON RANK LEVELS AGAINST *the best of comparison methods*, DENOTED IN BROWN.

Methods	**BTCV** (*w/o Pre-affine*) (MSE)			
	Dice(%)↑	HD95(mm)↓	ASSD(mm)↓	%fold ↓
Initial	33.5 (15.1)	41.33 (15.50)	17.69 (12.52)	-
○ SyN	61.2 (11.2)	25.57 (7.70)	8.21 (2.98)	1.9e-5
○ Demons	58.0 (12.4)	27.34 (8.04)	8.82 (3.39)	9.0e-3
○ B-Spline	51.6 (12.0)	30.93 (10.19)	10.33 (3.59)	5.8e-4
● VoxelMorph	48.1 (9.2)	30.76 (7.35)	11.02 (3.04)	7.1e-2
● MIDIR	46.4 (8.1)	30.85 (7.19)	11.25 (2.79)	1.3e-3
● VTN-10	52.4 (10.2)	28.95 (8.33)	10.01 (3.61)	2.8e-2
● LapIRN	56.9 (9.5)	26.41 (7.19)	8.81 (2.77)	5.9e-3
● SDHNet	63.1 (11.8)	26.31 (9.42)	8.27 (2.90)	5.9e-2
● PCReg	58.7 (14.4)	28.59 (7.06)	9.51 (4.84)	4.7e-2
● IIRP	65.5 (9.3)	**24.43** (7.06)	7.39 (2.52)	1.7e-2
● RDP	64.2 (10.9)	24.64 (7.86)	7.72 (3.10)	5.3e-3
● TransMorph	47.0 (9.9)	30.76 (7.43)	11.11 (3.12)	8.4e-2
● XMorpher	51.1 (8.6)	28.40 (7.80)	9.98 (3.05)	4.5e-2
● CGNet	64.6 (9.8)	24.51 (7.74)	7.51 (2.85)	2.4e-2
● DualPRNet++	59.8 (9.6)	26.73 (7.59)	7.74 (2.83)	2.9e-2
● CorrMLP	63.7 (9.5)	24.95 (7.96)	7.91 (2.96)	3.2e-2
● ReCorr-S	64.0*(9.1)	25.18†(7.34)	7.55*(2.59)	2.0e-2
● ReCorr	**66.3**†(8.9)	24.48*(7.60)	**7.08**†(2.45)	1.9e-2
● ReCorr-S-*diff*	63.8†(9.6)	24.83†(7.32)	7.38 (2.40)	**1.1e-5**
● ReCorr-*diff*	64.5(10.2)	24.82†(6.97)	7.29†(2.56)	3.0e-5

[1] ○ *Traditional Methods* ● *Pure Convolutional Networks* ● *Region-to-region Methods* ● *Voxel-to-region Methods*

[2] $*: p < 0.05, \quad \dagger: p < 5e\text{-}5$

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

TABLE II

QUANTITATIVE RESULTS ON THE OASIS DATASET, EITHER WITH AFFINE PRE-REGISTRATION (SMALL DEFORMATION) OR WITHOUT AFFINE PRE-REGISTRATION (LARGE DEFORMATION). MSE SERVES AS THE TRAINING SIMILARITY LOSS FOR ALL METHODS. ARROWS (↑, ↓) INDICATE THE PREFERABLE DIRECTION OF METRICS. THE STANDARD DEVIATION FOR EACH METRIC IS SHOWN IN PARENTHESES. AVERAGE TIME AND GPU ARE TESTED DURING THE INFERENCE STAGE ON THE NON-AFFINE OASIS DATASET WITH THE RESOLUTION OF 192 × 192 × 192. **BOLD** FONT INDICATES THE BEST VALUE, UNDERLINE FONT INDICATES THE SECOND-BEST VALUE. THE *best-performing baseline* IS HIGHLIGHTED IN BROWN. SUPERSCRIPTS * AND † DENOTE DIFFERENT WILCOXON SIGNED-RANK SIGNIFICANCE LEVELS IN COMPARISON WITH THIS BASELINE.

Methods	**OASIS** (MSE) *Pre-affine*				*w/o Pre-affine*				Time	GPU	Param.	FLOPs
	Dice(%)↑	HD95(mm)↓	ASSD (mm)↓	%fold ↓	Dice(%)↑	HD95(mm)↓	ASSD(mm)↓	%fold ↓	(s)	(MB)		(GMac)
Initial	53.8 (5.7)	4.08 (0.67)	1.76 (0.31)	-	13.4 (10.3)	17.00 (7.81)	9.85 (5.96)	-	-	-	-	-
○ SyN	77.5 (3.1)	2.33 (0.49)	0.78 (0.13)	**<1e-5**	69.1 (3.8)	2.67 (0.51)	1.00 (0.15)	**<1e-5**	178	-	-	-
○ Demons	77.9 (2.6)	2.37 (0.45)	0.75 (0.11)	3.8e-3	53.7 (22.5)	5.84 (5.41)	2.62 (3.25)	8.4e-4	114	-	-	-
○ B-Spline	65.0 (5.4)	3.30 (0.74)	1.24 (0.26)	**<1e-5**	60.7 (6.2)	3.42 (0.83)	1.35 (0.31)	**<1e-5**	3.75	-	-	-
● VoxelMorph	78.3 (2.3)	2.08 (0.31)	0.74 (0.09)	2.6e-3	59.4 (5.8)	3.97 (1.06)	1.45 (0.30)	1.2e-3	0.13	6318	0.33 M	408.5
● MIDIR	73.6 (2.5)	2.38 (0.31)	0.89 (0.10)	**<1e-5**	55.4 (5.9)	4.22 (1.13)	1.60 (0.37)	**<1e-5**	0.66	3712	0.21 M	31.1
● VTN-10	77.7 (1.8)	2.13 (0.31)	0.76 (0.08)	1.0e-4	65.2 (4.2)	3.02 (0.52)	1.14 (0.19)	4.3e-4	0.83	5940	2.89 M	76.2
● LapIRN	77.9 (2.1)	2.12 (0.31)	0.75 (0.09)	**<1e-5**	70.1 (3.3)	2.84 (0.72)	1.01 (0.18)	4.4e-5	0.01	5340	0.31 M	288.9
● SDHNet	79.9 (2.4)	1.87 (0.28)	0.68 (0.08)	3.2e-4	71.7 (2.3)	2.35 (0.36)	**0.77** (0.68)	1.9e-2	0.52	3510	18.29 M	185.5
● PCReg	81.3 (2.2)	1.81 (0.28)	0.65 (0.09)	2.2e-3	74.5 (2.0)	2.95 (0.48)	1.09 (0.13)	5.8e-3	0.48	18530	20.09 M	2264.2
● IIRP	81.0 (2.2)	1.85 (0.28)	0.66 (0.10)	5.4e-4	73.9 (2.3)	3.23 (0.53)	1.18 (0.15)	2.3e-4	3.42	4694	0.42 M	2805.6
● RDP	**81.6** (2.1)	1.78 (0.28)	0.65 (0.09)	**<1e-5**	**74.8** (1.8)	2.91 (0.46)	1.08 (0.12)	4.8e-5	4.54	4612	8.92 M	4161.8
● TransMorph	80.3 (2.2)	1.88 (0.27)	0.67 (0.08)	2.7e-3	58.7 (13.8)	5.46 (4.46)	1.56 (1.59)	2.5e-3	0.20	7648	46.77 M	734.2
● XMorpher	79.3 (2.0)	2.01 (0.25)	0.73 (0.07)	2.5e-3	62.1 (5.1)	3.37 (0.76)	1.28 (0.24)	7.8e-4	1.53	6456	20.51 M	1491.2
● CGNet	81.3 (2.1)	1.79 (0.28)	0.67 (0.09)	1.9e-3	74.7 (2.0)	2.89 (0.47)	0.83 (0.47)	5.6e-3	0.57	8930	25.53 M	962.2
● DualPRNet++	79.5 (2.1)	1.89 (0.29)	0.70 (0.09)	2.4e-3	70.8 (2.2)	3.02 (0.53)	0.97 (0.61)	5.3e-3	0.48	13324	0.61 M	839.8
● CorrMLP	81.5 (2.2)	**1.77** (0.28)	**0.64** (0.09)	2.1e-3	74.6 (2.1)	2.94 (0.36)	1.09 (0.68)	5.3e-3	1.03	13324	4.19 M	1197.6
● ReCorr-S	81.2†(2.1)	1.81†(0.28)	**0.64** (0.08)	1.6e-3	74.0*(2.0)	2.30*(0.39)	0.83†(0.09)	7.8e-4	0.11	5626	0.72 M	243.2
● ReCorr	81.4†(2.0)	1.79*(0.27)	**0.64** (0.08)	1.6e-3	74.7 (1.9)	**2.24**†(0.37)	0.81†(0.08)	7.6e-4	0.18	5626	0.72 M	396.4
● ReCorr-S-*diff*	80.6†(2.0)	1.86†(0.28)	0.66*(0.08)	**<1e-5**	71.8†(2.6)	2.40†(0.42)	0.88†(0.11)	**<1e-5**	0.24	5894	0.72 M	243.4
● ReCorr-*diff*	80.9†(2.0)	1.83†(0.27)	0.65†(0.08)	**<1e-5**	72.5†(2.4)	2.37*(0.42)	0.86†(0.10)	**<1e-5**	0.29	6166	0.72 M	396.5

[1] ○ *Traditional Methods* ● *Pure Convolutional Methods* ● *Region-to-region Matching Methods* ● *Voxel-to-region Matching Methods*
[2] $* : p < 0.05, \quad \dagger : p < 5e\text{-}5$

ages, and has been preprocessed by TransMorph, with steps include skull stripping, resampling, and affine transformation. All preprocessed images are cropped to a size of $160 \times 192 \times 224$ and 36 segmentation labels are used for evaluation. The scans are split into 403 training , 58 validation, and 115 test images. Scans from the IXI dataset serve as the moving images and are registered to an atlas brain MRI from [43].

The BTCV dataset includes 50 abdominal multi-organ CT scans from patients with metastatic liver cancer or postoperative abdominal hernia. All scans are resampled to a voxel spacing of $2 \times 2 \times 2.5$ mm, with intensity values clipped to [-900,1000] Hounsfield units and normalized to [0,1]. The scans are manually cropped to ensure consistent anatomical coverage and zero-padded to a size of $192 \times 160 \times 192$. The dataset is split into 35 training, 5 validation, and 10 test scans. For training, each scan is randomly paired with 10 others, resulting in 35×10 training pairs. Validation and test scans are inter-paired, resulting in 5×4 and 10×9 pairs, respectively. Five organ labels are used for evaluation: the spleen, left kidney, right kidney, liver, and stomach.

B. Evaluation Metrics

We evaluate registration accuracy using Dice coefficient, 95% Hausdorff Distance (HD95), and Average Symmetric Surface Distance (ASSD). Dice quantifies the overlap between anatomical segmentations, where the higher value means the better performance, and the maximum is 1. For multiple labels, we calculate an average Dice score. HD95 calculates the 95% largest distance between the closest points of two anatomical outlines (lower is better), offering more robustness against outliers than standard Hausdorff Distance. ASSD measures the average point-to-point surface distance, which provides a more comprehensive perspective and serves as a balanced quality metric. HD95 and ASSD are reported in millimeters (mm), computed using the voxel spacing information provided with each image.

Additionally, we assess deformation smoothness using the determinant of the Jacobian matrix, $|J_\phi(\boldsymbol{p})| = |\nabla\phi(\boldsymbol{p})| \in \mathbb{R}^{3\times3}$, which reflects the local characteristics of ϕ around voxel $\boldsymbol{p}$ where negative values suggest image folding. The percentage of non-positive values in $|J_\phi(\boldsymbol{p})|$, denoted as $\%_{\text{fold}}$, is used to quantify folding. Besides, we evaluate inference time, inference memory usage, parameter number, and FLOPs. For traditional methods, runtime is measured on CPU, while deep learning models are tested on an NVIDIA L40 GPU.

To assess statistical significance, we conduct Wilcoxon signed-rank tests between our method and the best-performing comparison method under each metric, using per-patient scores.

C. Baseline Methods

1) Traditional Methods: We select three widely-used algorithms for traditional methods: the Symmetric Normalization

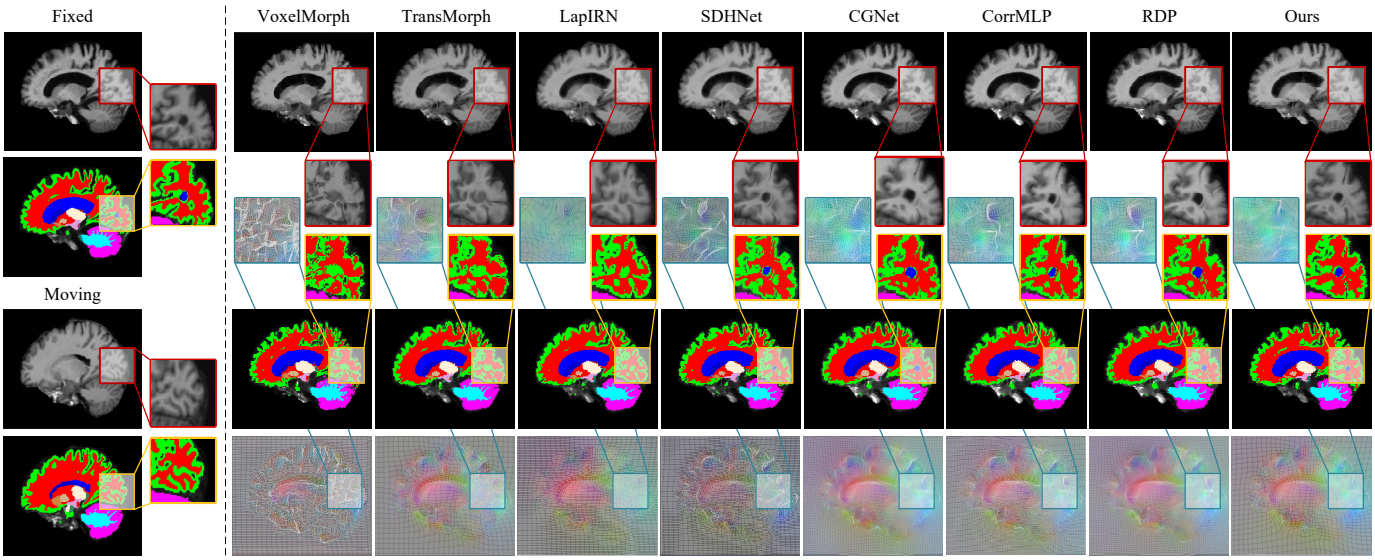

Fig. 6. A visualization example from the OASIS dataset *with affine pre-registration*. Left: the fixed image, the moving image, and their anatomical segmentation, along with zoomed-in views of specific areas at the same location. Right: the 1st, 3rd, and 4th rows show each method's warped moving images, the anatomical segmentation of these images, and the deformation field (shown by RGB images with overlaid grids), respectively. The 2nd row displays zoomed-in views of the same areas in the 1st, 3rd, and 4th rows.

(SyN) [25], Demons [22], and B-Spline [21]. Standard SyN is implemented using the ANTsPy Python package, with iterations set to (160, 80, 40). Demons registration is executed using the SimpleITK [44] Python package in a multi-scale fashion, with the iteration number set to 10. The B-Spline algorithm is applied using elastix [45], with iterations maximized at 500. For both SyN and B-Spline, the choice of metric is dataset-specific: MSE for the OASIS and BTCV datasets, and NCC for the IXI dataset, whose atlas and subjects exhibit wide variance in contrast and intensity distribution.

TABLE III

COMPARISON RESULTS ON THE IXI DATASET WHICH HAS UNDERGONE AFFINE PRE-REGISTRATION. NCC SERVES AS THE TRAINING SIMILARITY LOSS FOR ALL METHODS. ARROWS (↑, ↓) INDICATE THE PREFERABLE DIRECTION OF METRICS. THE STANDARD DEVIATION FOR EACH METRIC IS SHOWN IN PARENTHESES. **BOLD** FONT INDICATES THE BEST VALUE, UNDERLINE FONT INDICATES THE SECOND-BEST VALUE. SUPERSCRIPTS * AND † DENOTE DIFFERENT WILCOXON RANK LEVELS AGAINST *the best of comparison methods*, DENOTED IN BROWN.

Methods	**IXI** (*Pre-affine*) (NCC)			
	Dice(%) ↑	HD95 (mm) ↓	ASSD (mm) ↓	%fold ↓
Initial	35.5 (3.4)	8.00 (0.78)	3.11 (0.35)	-
○ SyN	65.7 (2.9)	4.97 (0.50)	1.46 (0.16)	**<1e-5**
○ Demons	56.8 (5.3)	5.81 (0.68)	1.80 (0.26)	2.8e-3
○ B-Spline	64.0 (3.6)	5.11 (0.66)	1.52 (0.20)	1.1e-4
● VoxelMorph	70.6 (2.4)	4.53 (0.51)	1.25 (0.13)	6.0e-3
● MIDIR	68.1 (2.4)	4.71 (0.52)	1.36 (0.15)	**<1e-5**
● VTN-10	70.5 (2.2)	4.70 (0.50)	1.27 (0.13)	7.0e-3
● LapIRN	71.2 (2.3)	4.44 (0.49)	1.23 (0.13)	2.7e-4
● SDHNet	70.4 (3.4)	4.59 (0.49)	1.26 (0.12)	1.3e-2
● PCReg	70.4 (2.1)	4.52 (0.49)	1.57 (0.18)	4.0e-3
● IIRP	71.6 (1.9)	4.44 (0.48)	1.54 (0.17)	1.9e-3
● RDP	71.4 (2.1)	4.47 (0.49)	1.53 (0.18)	2.7e-4
● TransMorph	70.5 (2.4)	4.52 (0.51)	1.24 (0.13)	6.3e-3
● XMorpher	70.7 (2.3)	4.49 (0.48)	1.24 (0.13)	5.1e-3
● CGNet	70.5 (2.2)	4.57 (0.49)	1.21 (0.52)	4.3e-3
● DualPRNet++	70.1 (3.2)	4.63 (0.49)	1.26 (0.12)	2.1e-2
● CorrMLP	70.4 (3.4)	4.59 (0.50)	1.25 (0.15)	1.3e-2
● ReCorr-S	72.5†(1.8)	4.37†(0.48)	1.19†(0.12)	3.4e-3
● ReCorr	**72.6**†(1.8)	4.37†(0.47)	**1.18**†(0.12)	2.9e-3
● ReCorr-S-*diff*	72.1†(2.0)	4.38†(0.48)	1.20†(0.11)	**<1e-5**
● ReCorr-*diff*	72.2†(1.9)	**4.36**†(0.48)	1.19†(0.12)	1.2e-5

[1] ○ *Traditional Methods* ● *Pure Convolutional Networks* ● *Region-to-region Methods* ● *Voxel-to-region Methods*

[2] $* : p < 0.05$, $\dagger : p < 5e\text{-}5$

2) Learning-based Methods: We reimplement thirteen learning-based approaches across three categories: pure convolutional methods, region-to-region explicit matching methods, and voxel-to-region explicit matching methods.

For pure convolutional methods, we include VoxelMorph [4] and MIDIR [5] as direct registration baselines; VTN [32] with 10 cascaded subnetworks as a representative recursive method; LapIRN [6] and PCReg [7] as representative pyramid-based methods; SDHNet [9] (with 6 iterations), RDP [10] and IIRP [36] as examples of pyramid-recurrent networks. For region-to-region matching methods, we compare TransMorph [12], which integrates both Transformer and convolution modules; XMorpher [13], a fully Transformer-based model; and CGNet [15], which adopts a pyramid structure with cross-window correlation. For voxel-to-region matching methods, we include DualPRNet++ [18], and CorrMLP [19], which expands the receptive field using multi-window MLPs.

For all the comparison methods, MSE is used as the similarity loss on the OASIS and BTCV datasets, with a regularization weight of 0.02. For the IXI dataset, due to substantial differences in data distribution, we choose NCC with a window size of 9 and a regularization weight of 1. The learning rate is set to 0.0001 with a weight decay rate of 0.0001, and the batch size is 1. Each method is trained for 200 epochs with validation performed after each epoch. The model achieving the best performance on the validation set is selected for final testing.

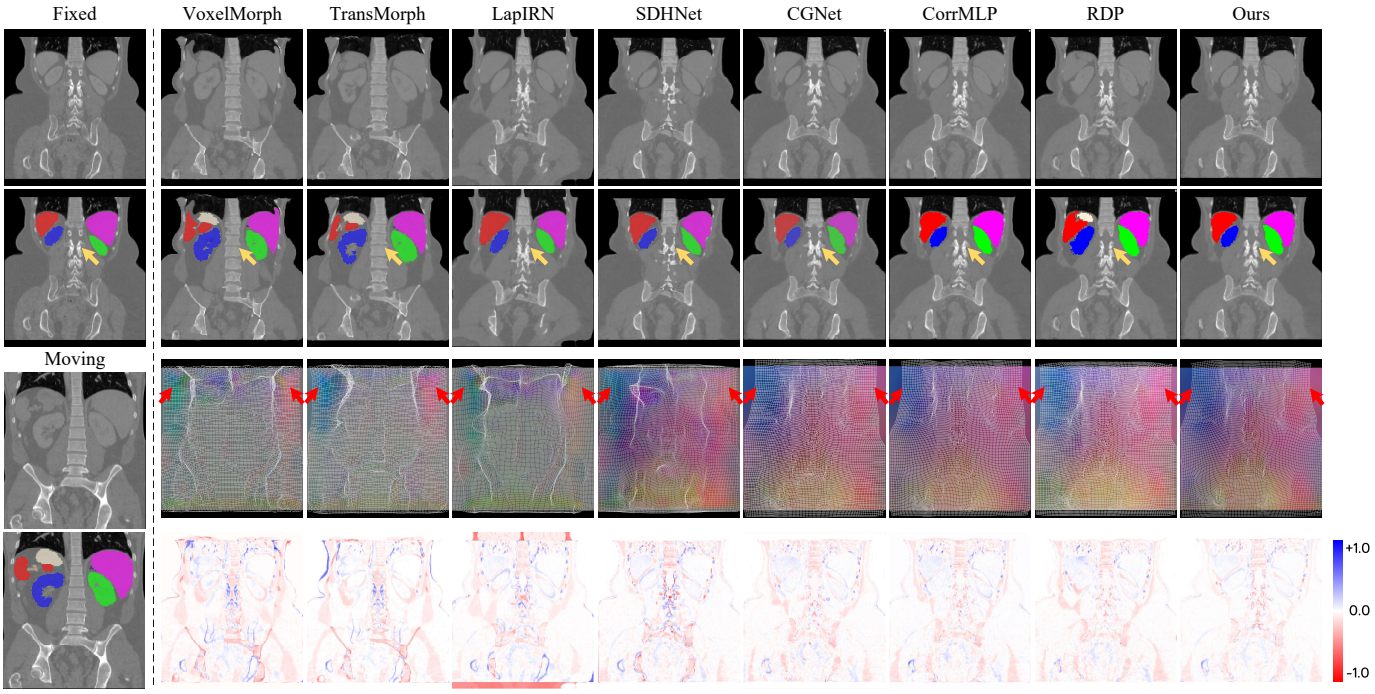

Fig. 7. A visualization example in the BTCV dataset without affine pre-registration. The first column shows the fixed and moving images with their segmentations; the remaining columns show results from different methods. The rows display, from top to bottom: warped images, warped segmentations, deformation fields (RGB grids), and difference maps. Yellow arrows indicate a location to distinguish performance. Red arrows indicate the locations where the deformation field generated by our method can demonstrate large linear stretching.

D. Implementation Details

ReCorr is implemented in PyTorch and training using the AdamW optimizer with a learning rate of 0.0007, a weight decay rate of 0.0004 and a batch size of 1. Consistent with other learning-based methods, our method is trained for 200 epochs. ReCorr performs $\{3,3,2,2\}$ recurrent iterations across four resolution scales ($1/16$, $1/8$, $1/4$, $1/2$). To accelerate inference in scenarios with relatively small deformations, we also provide ReCorr-S, a faster variant that uses fewer iterations $\{1,1,1,1\}$ at each scale. The diffeomorphic versions ReCorr-*diff* and ReCorr-S-*diff* are implemented under the same configurations.

The similarity loss function and its corresponding regularization weight are selected to match those used in other comparison methods: MSE with $\lambda = 0.02$ for the OASIS and BTCV datasets, and NCC with $\lambda = 1$ for the IXI dataset. The iteration decay hyperparameter γ is set to 0.7.

V. Results

To demonstrate the efficiency and accuracy of our method across different deformation scales, we conduct comprehensive comparisons with the existing methods in two settings: 1) *Regular* experiments with affine pre-registration for small deformations, and 2) *Extreme* tests on datasets without affine pre-registration for large deformations.

A. Compare to Start-of-the-art Methods

We evaluate our method across four benchmark datasets: OASIS (pre-affine and non-affine), IXI, and BTCV. The quantitative results are shown in Tables II, III and I, our proposed ReCorr consistently achieves competitive or superior accuracy compared with recent state-of-the-art methods, while offering a strong balance between accuracy and efficiency.

On the non-affine OASIS and BTCV datasets, which involve large anatomical deformations, ReCorr achieves strong results in all metrics. On BTCV, ReCorr achieves the highest Dice score (66.3%) as well as the best HD95 (24.48 mm) and ASSD (7.08 mm) among all methods, confirming its advantage under large-deformation conditions. While on OASIS (non-affine), ReCorr reaches a Dice score of 74.7%, ranking second only to RDP (74.8%), but yields a lower HD95 (2.24 mm *vs.* 2.91 mm) and significantly better efficiency in terms of parameters (0.72M *vs.* 8.92M), FLOPs (396.4G *vs.* 4161.8G), and inference time (0.18 s *vs.* 4.54 s). In contrast to most methods with comparable accuracy, ReCorr achieves these results with only 0.72M parameters and moderate computational cost, demonstrating a strong accuracy-efficiency trade-off under large-deformation scenarios. Fig. 5 visualizes the relationship between registration accuracy (Dice), inference time, FLOPs, and model parameters on the non-affine OASIS dataset. ReCorr achieves one of the highest Dice scores while maintaining moderate inference time, FLOPs, and parameter count. Compared to methods with similar accuracy, such as RDP and CorrMLP, ReCorr requires substantially fewer computational resources. Overall, our method achieves an excellent accuracy-efficiency trade-off.

On the pre-affine datasets OASIS and IXI, where deformations are relatively mild, our method remains highly competitive. ReCorr-S achieves Dice scores of 81.2% on OASIS and 72.5% on IXI, performing on par with or surpassing most existing methods. On OASIS, some approaches such as

CorrMLP (81.5%) and RDP (81.6%) report slightly higher Dice scores, but incur significantly greater computational cost. In contrast, ReCorr-S achieves comparable accuracy with only 0.72M parameters, 243G FLOPs, and a fast 0.11s inference time. On the IXI dataset, ReCorr-S not only surpasses all previous methods in Dice, outperforming RDP (71.4%) and IIRP (71.6%), but also maintains the lowest computational burden among top-performing methods. These results highlight the strong accuracy-efficiency trade-off of ReCorr-S under small-deformation conditions, making it a practical and effective choice for time- and resource-sensitive scenarios.

Fig. 6. visualizes the comparison results of ReCorr-S with some representative methods of pure convolutional methods, region-to-region matching methods, and voxel-to-region matching methods. Compared to other methods, our approach produces a warped image with more accurate details, while maintaining smoother and more coherent deformation fields. Fig. 7 shows qualitative results of ReCorr on the non-affine BTCV dataset. As seen in the third row, our model effectively captures rigid, linear displacements from the moving to fixed image, despite not explicitly modeling affine transformations. Additionally, the deformation fields generated by our method are visually smoother than those from other methods.

B. Trade-off between ReCorr and ReCorr-S

While both ReCorr and ReCorr-S are built upon the same framework, their performance trends across datasets reveal practical insights. On datasets with small deformations (*i.e.*, pre-affine OASIS and IXI), the performance gap is marginal: ReCorr-S achieves nearly the same Dice scores and in some cases even slightly lower HD95, suggesting that a single iteration per scale is sufficient to achieve accurate alignment.

TABLE IV
COMPARISON RESULTS UNDER THE SEMI-SUPERVISED SETTING ON THE NON-AFFINE OASIS DATASET AND IXI DATASET. ARROWS (↑, ↓) INDICATE THE PREFERABLE DIRECTION OF METRICS. THE STANDARD DEVIATION FOR EACH METRIC IS SHOWN IN PARENTHESES. **BOLD** FONT INDICATES THE BEST VALUE, UNDERLINE FONT INDICATES THE SECOND-BEST VALUE.

Methods	*non-affine* **OASIS** (MSE)		**IXI** (NCC)	
	Dice(%)↑	HD95(mm)↓	Dice(%)↑	HD95(mm)↓
Initial	13.4 (10.3)	17.00 (7.81)	35.5 (3.4)	8.00 (0.78)
● VoxelMorph	73.8 (2.2)	3.58 (0.57)	82.1 (1.5)	3.71 (0.44)
● MIDIR	71.9 (5.2)	3.69 (1.01)	79.4 (1.7)	3.92 (0.47)
● VTN-10	85.1 (1.8)	2.24 (0.47)	80.2 (1.6)	3.84 (0.45)
● SDHNet	86.2 (1.3)	1.90 (0.29)	83.6 (1.6)	3.61 (0.46)
● PCReg	87.2 (1.2)	2.01 (0.38)	83.6 (1.7)	3.59 (0.45)
● IIRP	86.4 (1.1)	1.98 (0.32)	83.7 (1.8)	3.55 (0.42)
● RDP	**89.1** (1.2)	2.09 (0.41)	84.3 (1.4)	3.60 (0.42)
● TransMorph	74.3 (5.4)	3.74 (1.27)	82.1 (1.5)	3.71 (0.41)
● XMorph	79.6 (1.5)	2.85 (0.55)	83.3 (1.7)	3.66 (0.44)
● CGNet	88.7 (1.2)	1.88 (0.33)	82.9 (1.5)	3.67 (0.44)
● DualPRNet++	88.3 (1.3)	1.96 (0.42)	83.2 (1.6)	3.65 (0.43)
● CorrMLP	88.3 (1.2)	1.92 (0.34)	83.7 (1.8)	3.55 (0.45)
● ReCorr	88.9 (1.2)	**1.82** (0.30)	**85.8** (1.6)	**3.52** (0.43)

[1] ○ *Traditional Methods* ● *Pure Convolutional Networks* ● *Region-to-region Methods* ● *Voxel-to-region Methods*

TABLE V
ABLATION RESULTS ON LOCAL SEARCH. SETTINGS USED IN OUR FINAL MODEL ARE UNDERLINED. BOLD FONT INDICATES THE BEST VALUE.

Experiment	*non-affine* **OASIS**		**BTCV**		**FLOPs**
	Dice(%)↑	HD95 (mm)↓	Dice(%)↑	HD95 (mm)↓	(GMac)
feature only	72.9 (2.1)	2.38 (0.43)	55.4 (10.7)	28.85 (7.56)	396.2
r=1(corr)	73.5 (2.1)	2.34 (0.42)	63.6 (9.3)	25.75 (7.32)	395.5
r=3(corr)	**74.7** (1.9)	**2.24** (0.37)	66.3 (8.9)	**24.48** (7.60)	396.4
r=5(corr)	74.7 (1.9)	2.24 (0.36)	**66.4** (9.0)	24.53 (7.69)	399.5

TABLE VI
ABLATION RESULTS ON THE RECURRENT UPDATER. BOLD FONT INDICATES THE BEST VALUE.

Feature		Components			Param. Share		*non-affine* **OASIS**	
Decouple	Concat	Conv	GRU	LSTM	Scale	All	Dice(%)↑	HD95 (mm)↓
	✓	✓			✓		73.6 (2.2)	3.21 (0.57)
✓		✓			✓		74.1 (2.0)	2.26 (0.38)
✓			✓		✓		74.5 (1.9)	2.29 (0.47)
✓		✓				✓	74.4 (2.0)	2.25 (0.37)
✓		✓			✓		**74.7** (1.9)	**2.24** (0.37)

Given their identical parameter count, ReCorr-S is preferred in such cases due to its shorter inference time.

In contrast, on datasets with large deformations (*i.e.*, non-affine OASIS and BTCV), ReCorr shows clearer improvements over ReCorr-S. It achieves a 0.7%–2.3% higher Dice score, along with noticeably improved HD95 and ASSD, demonstrating the benefit of additional recurrent updates in handling complex anatomical variability. These observations suggest a flexible usage strategy: ReCorr-S is well-suited for efficiency-critical applications and performs effectively under small deformations, while ReCorr is better suited for scenarios requiring maximum accuracy in large-deformation cases.

C. Semi-supervised Setting

Optionally, to further improve performance, we adopt a semi-supervised training strategy following [4], where an additional Dice loss is used on labeled data alongside the unsupervised similarity and regularization losses. We evaluate this setup on the non-affine OASIS and IXI datasets, which contain varying degrees of deformation.

As shown in Table IV, ReCorr achieves state-of-the-art performance under this setting. On the OASIS dataset, ReCorr attains a Dice score of 88.9%, surpassing the previous best result (89.1% by RDP) with a notably lower HD95 (1.82 mm vs. 2.09 mm). On the IXI dataset, which exhibits smaller deformations, ReCorr reaches the highest Dice score (85.8%) and the lowest HD95 (3.52 mm) among all methods. These results demonstrate that our model benefits from the additional supervised signal and remains robust across both small and large deformation scenarios.

D. Ablation Study

Our ablation experiments assess the effectiveness of the local search module, recurrent updater, multi-scale architecture, operations at the original resolution, and sequence supervision

TABLE VII
ABLATION STUDIES FOR ITERATION NUMBERS ON 4 SCALES. THE BOLD FONT INDICATES THE BEST VALUE.

Iterations				OASIS (*w/o Pre-affine*)	
1/16	1/8	1/4	1/2	Dice (%)↑	HD95 (mm)↓
0	1	0	0	59.9 (5.4)	3.65 (0.83)
0	1	1	0	70.3 (2.9)	2.69 (0.52)
1	1	1	0	71.6 (2.0)	2.50 (0.38)
1	1	1	1	**74.0** (2.0)	**2.30** (0.39)
2	1	1	1	74.3 (1.9)	2.26 (0.37)
1	2	1	1	74.3 (2.0)	2.27 (0.38)
1	1	2	1	74.2 (1.9)	2.27 (0.36)
1	1	1	2	74.4 (1.9)	2.26 (0.37)
2	2	2	1	74.3 (2.0)	2.26 (0.38)
3	3	1	1	74.5 (2.0)	2.25 (0.37)
2	2	2	2	74.6 (2.0)	2.26 (0.41)
2	3	2	2	74.6 (1.9)	2.25 (0.38)
3	3	2	2	**74.7** (1.9)	**2.24** (0.37)

TABLE VIII
ABLATION STUDIES FOR OPERATIONS AT ORIGINAL RESOLUTION. SETTINGS USED IN OUR FINAL MODEL ARE UNDERLINED. THE BOLD FONT INDICATES THE BEST VALUE.

Ori-Res. Setting			OASIS (*w/o Pre-affine*)		GPU	Time	FLOPs
Plain	Refine	Iter.(1)	Dice (%)↑	HD95 (mm)↓	(MB)	(s)	(GMac)
✓			74.3 (1.9)	2.26 (0.37)	5346	0.16	376.3
	✓		74.7 (1.9)	2.24 (0.37)	5626	0.18	396.4
		✓	**75.4** (1.9)	**2.20** (0.35)	14586	0.49	1103.8

strategy, as well as the hyperparameters of iteration numbers and γ. Unless specifically emphasized, we default to conducting ablation experiments for ReCorr on the non-affine OASIS dataset with Dice and HD95.

1) Local Search: We conduct ablation studies to assess the design choices in the local search module, as shown in Table V. The "feature only" variant skips correlation computation and directly uses concatenated features from the fixed and moving images. Using correlation yields clear improvements (e.g., 74.7% vs. 72.9% Dice on non-affine OASIS), indicating its advantage in capturing voxel correspondences. We further evaluate different search scopes (r=1, 3, 5) using correlation-based matching. Increasing the scope substantially improves performance from r=1 to r=3, especially on the BTCV dataset with large deformations. The performance at $r = 5$ shows no clear improvement over $r = 3$ (e.g., 66.4% vs. 66.3% Dice on BTCV), and the computational cost remains nearly the same. This suggests that $r = 3$ already provides sufficient search flexibility, and further increasing the radius offers little benefit.

2) Recurrent Updater: We ablate the design of the recurrent updater from three aspects: feature integration, update components, and parameter sharing, as shown in Table VI. First, we compare two strategies for feature integration: direct concatenation *vs.* explicit decoupling into motion-related and texture branches. The decoupled design yields better performance (74.7% *vs.* 73.6% Dice), suggesting improved alignment by reducing semantic interference. Second, we test different updater architectures. Replacing GRU with plain convolution reduces performance, highlighting the benefit of temporal memory. LSTM performs slightly worse than GRU, possibly attributed to the coarse-to-fine nature of our registration strategy, where large deformation trends are captured in early stages. In later stages, the memory unit primarily serves to refine local updates, and thus does not require the more complex gating mechanisms of LSTM. The GRU with simpler structure proves sufficient for propagating deformation cues across iterations. Lastly, we evaluate parameter sharing. Sharing parameters across all scales leads to degraded performance, while within-scale sharing offers a better balance of accuracy and efficiency. Our final model adopts feature decoupling, GRU, and scale-specific sharing.

3) Multi-scale for Iteration: We evaluate the impact of iterative refinement across multiple scales: 1/16, 1/8, 1/4, 1/2. As shown in Table VII, performing iteration only at the 1/8 scale leads to poor performance (Dice 59.9%). Adding iterations at the 1/4 scale significantly improves the Dice score to 70.3%, and incorporating 1/16 further boosts performance to 74.0%. These results suggest that multi-scale refinement is essential for capturing deformations of different magnitudes.

We also study the effect of increasing iteration counts at each scale. Results show that more iterations improve performance, especially at higher-resolution scales like 1/2. However, the gain from combining additional iterations is sublinear, suggesting that benefits are synergistic rather than additive. Based on the trade-off between accuracy and computational cost, we set the iteration numbers to 3,3,2,2 for ReCorr and 1,1,1,1 for ReCorr-S.

4) Original Resolution Operations: As shown in Table VIII, we evaluate three strategies at the original resolution: no operation (Plain), feature-based refinement (Refine), and one iteration (Iter.(1)). Adding a refinement step improves performance from 74.3% to 74.7% Dice with minimal increase in memory and time. Performing one full-resolution iteration yields further improvement (75.4% Dice, 2.20 mm HD95), but incurs a substantial increase in GPU usage ($\times$2.6 memory) and FLOPs (376G $\rightarrow$ 1103G). Considering the accuracy-efficiency trade-off, we adopt feature refinement at full resolution while

TABLE IX
ABLATION RESULTS FOR ROBUSTNESS EVALUATION UNDER SYNTHETIC PERTURBATIONS.

Deformation Type	Level	OASIS (*w/o Pre-affine*)	
		Dice(%) ↑	HD95 ↓
No Augmentation		74.7 (2.1)	2.94 (0.36)
SVF	s=2	74.2 (2.0)	3.03 (0.40)
	s=4	74.5 (2.1)	2.96 (0.49)
	s=8	74.4 (1.9)	2.95 (0.47)
Affine	Offset (-0.2)	71.7 (1.8)	3.29 (0.47)
	Offset (-0.1)	74.5 (2.3)	2.98 (0.52)
	Offset (0.1)	74.1 (2.3)	3.20 (0.57)
	Offset (0.2)	72.1 (1.9)	3.18 (0.44)
	Scale (-0.2)	73.4 (2.0)	3.07 (0.49)
	Scale (-0.1)	74.0 (2.0)	3.12 (0.50)
	Scale (0.1)	73.3 (2.3)	3.27 (0.59)
	Scale (0.2)	72.3 (2.3)	3.38 (0.75)

TABLE X

ABLATION OF SEQUENCE SUPERVISION STRATEGY. WE COMPARE TWO DESIGNS FOR THE SEQUENCE LOSS WITH EXPONENTIALLY INCREASING WEIGHTS: USING ALL INTERMEDIATE DEFORMATION FIELDS ("FULL SEQUENCE") OR ONLY THE FINAL OUTPUT AT EACH SCALE ("LAST OF SCALE"). SETTINGS USED IN OUR FINAL MODEL ARE UNDERLINED. THE BEST VALUE IS SHOWN IN BOLD.

Experiment		OASIS *(w/o Pre-affine)*		Train GPU
Full Sequence	Last of scale	Dice (%)↑	HD95 (mm) ↓	(MB)
✓		**74.7** (1.9)	**2.24** (0.37)	23544
	✓	74.5 (2.0)	2.25 (0.38)	19266

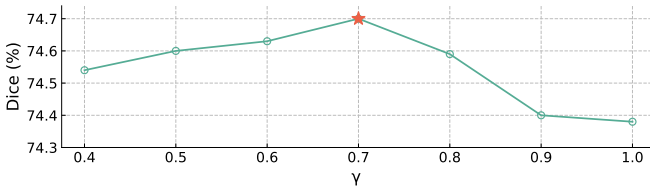

Fig. 8. Line graph of ablation results for the hyperparameter γ of loss on the OASIS dataset without affine pre-registration. The orange star indicates the setting used in our final model.

keeping recurrent iterations at lower scales in our final model.

5) Sequence Supervision Strategy: We evaluate two strategies for applying sequence loss during training: one that supervises all intermediate deformation fields across iterations ("Full Sequence") and another that only supervises the final output at each scale ("Last of scale"). As shown in Table X, the "Full Sequence" strategy yields slightly better accuracy (Dice: 74.7% *vs.* 74.5%) and lower HD95. Although it incurs higher GPU memory, the additional supervision provides more consistent optimization signals across iterations. Thus, we adopt the "Full Sequence" strategy for better optimization guidance and accuracy.

We further evaluate the impact of γ, which balances the relative weight of earlier and later predictions in the sequence loss. As shown in Fig. 8, a value of $\gamma = 0.7$ achieves the best Dice score, indicating an effective trade-off between emphasizing final predictions and preserving intermediate supervision. Therefore, we set $\gamma = 0.7$ as the default for all experiments.

6) Robustness Evaluation under Synthetic Perturbations: To evaluate the robustness of our model under varying deformation conditions, we conduct additional experiments on the non-affine OASIS dataset with synthetic perturbations. Specifically, we apply both non-linear and affine transformations to simulate diverse and challenging deformation scenarios. For non-linear perturbations, we introduce SVF-based fields at $s = 2$, 4, and 8, corresponding to resolutions of (½, ¼, ⅛), respectively. The $s = 2$ setting introduces more local and fine-grained deformations, allowing us to assess the model's sensitivity to subtle spatial variations. For affine perturbations, we apply controlled global shifts (offsets of ± 0.1 and ± 0.2) and scaling factors along the x-axis.

As shown in Table IX, the model maintains stable performance across most settings, with only moderate drops under strong perturbations. For SVF-based deformations, performance remains comparable to the baseline, confirming the robustness of the model to local non-linear distortions. In the affine setting, small offsets and scalings have minimal impact, while larger transformations (*e.g.*, offset = -0.2 or scale = 0.2) lead to a more noticeable decline in Dice, indicating the increased challenge posed by global shifts.

VI. DISCUSSION & CONCLUSION

This paper addresses the challenge of large deformation registration by proposing ReCorr, an efficient recurrent framework based on explicit voxel-to-region matching. While explicit feature matching strategies better align with the goal of voxel-level correspondence modeling, existing methods often suffer from high computational cost or limited search regions. ReCorr tackles this by performing iterative local search, where each step performs voxel-to-region matching within a small neighborhood and updates the search center accordingly. This enables the progressive convergence toward globally optimal alignment at minimal cost per iteration. To further reduce redundancy and improve alignment quality, ReCorr decouples motion-related and texture-related information into separate branches, allowing the network to focus on spatial correspondence without interference from irrelevant semantic cues. Compared to prior works, ReCorr achieves superior or competitive accuracy across diverse experimental setups, while maintaining high efficiency in both computational cost and inference speed.

However, several aspects remain to be explored. Adaptive strategies for the size of the search region could be used to improve the efficiency and accuracy of the network, such as using a larger search scope for the first iteration and a smaller one for the subsequent iterations. In addition, while local matching at low resolution is effective in many large deformation cases, it may struggle in extreme scenarios such as large-angle rotations or flipped structures, where local cues become ambiguous. In such cases, introducing sparse global matching may offer more reliable guidance and help improve convergence. Finally, adaptive attention mechanisms could help the network focus on informative regions rather than fixed local neighborhoods, improving alignment in ambiguous or low-contrast areas. These aspects could be further investigated to improve the performance of the network and its applicability in real-clinical scenarios.

In summary, our work offers a robust and efficient solution for large deformation registration within the explicit matching paradigm, achieving a favorable balance between accuracy and computation.

REFERENCES

[1] A. Sotiras *et al.*, "Deformable medical image registration: A survey," *IEEE Transactions on Medical Imaging*, 2013.
[2] F. Giesel *et al.*, "Image fusion using CT, MRI and PET for treatment planning, navigation and follow up in percutaneous RFA," *Experimental Oncology*, 2009.
[3] J. Chen *et al.*, "A survey on deep learning in medical image registration: New technologies, uncertainty, evaluation metrics, and beyond," *Arxiv:2307.15615*, 2023.
[4] G. Balakrishnan *et al.*, "VoxelMorph: a learning framework for deformable medical image registration," *IEEE Transactions on Medical Imaging*, 2019.

[5] H. Qiu *et al.*, "Learning diffeomorphic and modality-invariant registration using B-splines," in *MIDL*, 2021.

[6] T. C. Mok *et al.*, "Large deformation diffeomorphic image registration with laplacian pyramid networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2020.

[7] W. Yin *et al.*, "PC-Reg: A pyramidal prediction–correction approach for large deformation image registration," *Medical image analysis*, 2023.

[8] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *Arxiv Preprint Arxiv:2010.11929*, 2020.

[9] S. Zhou *et al.*, "Self-distilled hierarchical network for unsupervised deformable image registration," *IEEE Transactions on Medical Imaging*, 2023.

[10] H. Wang *et al.*, "Recursive deformable pyramid network for unsupervised medical image registration," *IEEE Transactions on Medical Imaging*, 2024.

[11] M. P. Heinrich, "Closing the gap between deep and conventional image registration using probabilistic dense displacement networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2019.

[12] J. Chen *et al.*, "TransMorph: Transformer for unsupervised medical image registration," *Medical image analysis*, 2022.

[13] J. Shi *et al.*, "XMorpher: Full Transformer for deformable medical image registration via cross attention," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2022.

[14] Z. Chen *et al.*, "TransMatch: a transformer-based multilevel dual-stream feature matching network for unsupervised deformable image registration," *IEEE transactions on medical imaging*, 2023.

[15] Y. Chang *et al.*, "CGNet: A correlation-guided registration network for unsupervised deformable image registration," *IEEE Transactions on Medical Imaging*, 2024.

[16] Z. Liu *et al.*, "Swin Transformer: Hierarchical vision Transformer using shifted windows," in *ICCV*, 2021.

[17] L. Wu, J. Zhuang, and H. Chen, "VoCo: A simple-yet-effective volume contrastive learning framework for 3D medical image analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[18] M. Kang *et al.*, "Dual-stream pyramid registration network," *Medical image analysis*, 2022.

[19] M. Meng *et al.*, "Correlation-aware coarse-to-fine MLPs for deformable medical image registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[20] R. Bajcsy and S. Kovačič, "Multiresolution elastic matching," *Computer Vision, Graphics, and Image Processing*, 1989.

[21] D. Rueckert *et al.*, "Nonrigid registration using free-form deformations: application to breast MR images," *IEEE Transactions on Medical Imaging*, 1999.

[22] X. Pennec *et al.*, "Understanding the "demon's algorithm": 3D non-rigid registration by gradient descent," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 1999.

[23] T. Vercauteren *et al.*, "Diffeomorphic demons: Efficient non-parametric image registration," *Neuroimage*, 2009.

[24] J. Glaunes *et al.*, "Large deformation diffeomorphic metric curve mapping," *IJCV*, 2008.

[25] B. B. Avants *et al.*, "Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain," *Medical image analysis*, 2008.

[26] H. Sokooti *et al.*, "Nonrigid image registration using multi-scale 3D convolutional neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2017.

[27] B. D. De Vos *et al.*, "A deep learning framework for unsupervised affine and deformable image registration," *Medical image analysis*, 2019.

[28] T. C. Mok *et al.*, "Robust image registration with absent correspondences in pre-operative and follow-up brain MRI scans of diffuse glioma patients," in *International MICCAI Brainlesion Workshop*, 2022.

[29] X. Song *et al.*, "Dino-Reg: Efficient multimodal image registration with distilled features," *IEEE Transactions on Medical Imaging*, 2025.

[30] K. Chen *et al.*, "A novel few-shot learning framework for supervised diffeomorphic image registration network," *IEEE Transactions on Medical Imaging*, 2025.

[31] X. Jia *et al.*, "Decoder-only image registration," *IEEE Transactions on Medical Imaging*, 2025.

[32] S. Zhao *et al.*, "Unsupervised 3D end-to-end medical image registration with volume tweening network," *IEEE Journal of Biomedical and Health Informatics*, 2019.

[33] H. Xiaojun *et al.*, "Dual-stream pyramid registration network." in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2019.

[34] J. Jiang *et al.*, "One shot PACS: Patient specific anatomic context and shape prior aware recurrent registration-segmentation of longitudinal thoracic cone beam cts," *IEEE transactions on medical imaging*, 2022.

[35] A. Hering *et al.*, "mlVIRNET: Multilevel variational image registration network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2019.

[36] T. Ma *et al.*, "IIRP-Net: iterative inference residual pyramid network for enhanced image registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[37] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *MWSCAS*, 2017.

[38] C. Szegedy *et al.*, "Rethinking the Inception architecture for computer vision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[39] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *ECCV*, 2020.

[40] V. Arsigny *et al.*, "A log-Euclidean framework for statistics on diffeomorphisms," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2006.

[41] D. S. Marcus *et al.*, "Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults," *Journal of Cognitive Neuroscience*, 2007.

[42] B. Landman *et al.*, "MICCAI multi-atlas labeling beyond the cranial vault–workshop and challenge," in *Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*, 2015.

[43] B. Kim *et al.*, "CycleMorph: cycle consistent unsupervised deformable image registration," *Medical image analysis*, 2021.

[44] R. Beare *et al.*, "Image segmentation, registration and characterization in R with simpleitk," *Journal of Statistical Software*, 2018.

[45] S. Klein *et al.*, "Elastix: a toolbox for intensity-based medical image registration," *IEEE Transactions on Medical Imaging*, 2009.