A Spatiotemporal Agent for Robust Multimodal Registration

ZIWEI LUO°1, XIN WANG°2, XI WU°1, (Member, IEEE), YOUBING YIN°2, KUNLIN CAO°2, QI SONG°2, AND JING HU°1, (Member, IEEE)

1Department of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China
2CuraCloud Corporation, Seattle, WA 98104, USA

Corresponding author: Jing Hu (jing_hu09@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602065, in part by the Scientific Research Foundation of the Education Department of Sichuan Province under Grant 17ZA0062, in part by the Chengdu University of Information and Technology (CUIT) Foundation for Leaders of Disciplines in Science under Project J201608, and in part by the Scientific Research Foundation of CUIT under Project KYTZ201610.

ABSTRACT Multimodal image registration is a crucial step for a variety of medical applications to provide complementary information from the combination of various data sources. Conventional image registration methods aim at finding a suited similarity metric as well as a descriptive image feature, which is quite challenging due to the high diversity of tissue appearance across modalities. In this paper, we present a novel approach to register images via an asynchronously trained reinforcement learning agent automatically. Within this approach, convolutional gated recurrent units (ConvGRU) is incorporated after stacked convolutional layers to extract both spatial and temporal features of the neighboring frames and implicitly learn the similarity metric. Moreover, we propose a customized reward function driven by fixed points error (FPE) to guide the agent to the correct registration direction. A Monte Carlo rollout strategy is also leveraged to perform a look-ahead inference to the elimination of jitter in the test stage. Evaluation is performed on paired CT and MR images from patients diagnosed as nasopharyngeal carcinoma. The results demonstrate that our method achieves state-of-the-art performance in medical image registration.

INDEX TERMS Medical image, reinforcement learning, multimodal registration, actor-critic, convolutional GRU.

I. INTRODUCTION Multimodal registration is concerned with involving two or more images from different modalities into spatial alignment. Usually, a fixed image remains stationary, while the moving image is transformed to match the fixed image. Fig. 1 shows some examples of image pairs to be registered. Image registration is a vital component in current clinical applications, such as treatment planning, and minimally invasive surgery [23]. For example, the combination of Computed Tomography (CT) and Magnetic Resonance (MR) would be beneficial for radiotherapy treatment planning, and the registration on the brain could be helpful for monitoring the development of brain cancer.

Traditional image registration can be categorized into two classes: intensity-based [2] or feature-based [3]. Intensity-based methods operate directly on image intensities via some similarity metrics such as sum-of-squared-differences, mutual information, and correlation coefficient [24]. Feature-based methods register images by using the correspondence between image features, including gradient, geometric shape, contour, landmarks, and the response of Gabor filter [23]. However, both of these two kinds of image registration methods are designed to extract handcrafted image features with poor adaptability to multimodal image registration. Deep learning (DL) methods, by contrast, are capable of automatically learning features from raw data, and they have changed the landscape of image registration research.

In deep-learning-based methods, image registration is regarded as a data-driven optimizing problem. Registration is achieved by maximizing a similarity metric between a moving image and a fixed image, and the corresponding transformation parameters are directly learned. Convolutional Neural Networks (CNNs) are the most commonly used techniques in recent computer vision tasks such as image
Classification, segmentation, and object detection [34]. In registration, CNNs are employed to extract hierarchical features embedded in the raw visual input and learn the high-level relationship between images [6]. Deep learning techniques used in registration is capable of avoiding the common pitfall of intractable non-convex optimization in traditional registration methods, which always leads to local optima, i.e., the null gradient at saddle points [17]. Moreover, the highly parallel property of CNN makes the registration progress fast and computationally efficient. Work in [18] was the first approach to use CNN to estimate the transformation parameters of the rigid registration directly. A hierarchical CNN regression model was proposed in this work, in which six transformation parameters were divided into three groups to reduce the complexity of regression. The experiments demonstrated that this approach outperformed most traditional intensity-based approaches.

Meanwhile, as a breakthrough in combining image registration with deep learning techniques, deep reinforcement learning (DRL) demonstrated its potential about mastering complicated control policies [4], [5]. The goal of RL methods is to learn a policy by trial-and-error in interaction with the environment. At the same time, deep neural networks provide powerful function approximation and representation learning capabilities. Under this framework, the registration problem could then be reformulated as a sequential decision-making problem mimicking an expert performs image registration via selecting actions continually, as shown in Fig 2. For example, Liao et al. [1] trained the agent in the RL framework to predict the optimal action that best enhances the alignment at each time step in a greedy supervised fashion. Ma et al. [6] adopted a dueling deep Q-network (DDQN), to train the artificial agent to register multimodal medical images with contextual information. However, the greedy strategy in [1] may not be globally optimal, and the DDQN architecture in [6] needs a huge space to store the replay memory. Worse still, both methods ignored the temporal dependencies that may accelerate training. In light of these limitations, our previous work [16] presented an end-to-end RL model with long-term recurrent network architecture (LSTM) [13] and a reward function driven by landmark error (LME), which needed no extra storage and explored transformation parameter spaces freely. However, it still suffers from certain limitations. First, although spatial features are very crucial for matching in the registration task, the fully-connected (FC) formulations in conventional LSTM during both input-to-state and state-to-state transitions do not have the flexibility in capturing spatial contextual pixel connections between states. Second, LME reward computation is rather complicated and hence aggravates the burden of RL training.

In this paper, we mainly focus on the feasibility of using DRL in 2D rigid registration, and propose an asynchronous agent-based multimodal registration method that handles the aforementioned challenges. The main contributions of our work can be summarized as follows:

- Different from the traditional registration algorithms, we reformulate 2D registration as a 3D sequence learning task. And we introduce a DRL framework derived from asynchronous advantage actor-critic (A3C) [9] that executes multiple agents in parallel to accelerate convergence. Moreover, using different agents to learn a global policy on multiple instances of the environment simultaneously can decorrelated training into a more stable and effective process, since at any given time, the parallel agents were experiencing a variety of different states.
- We propose a spatiotemporal agent to capture both spatial and temporal information for an image sequence by applying a convolutional gated recurrent unit (ConvGRU). Context information plays an important role in selecting actions in reinforcement learning as well as in RL-based registration. The ConvGRU is used for obtaining the historical characteristics, so as to preserve spatial connectivities in the frames.
- We propose a novel reward function driven by fixed-points error (FPE), which is similar to the landmark error proposed in [16] but more computationally efficient. Using the FPE-based reward function facilitates the extension of 3D image registration.
- A Monte Carlo rollout (MC rollout) method is adopted in the inference to improve the registration capability. So far, our network has outperformed several state-of-the-art image registration methods on our dataset.
The remainder of this paper is organized as follows. Section II discusses related work on image registration and reinforcement learning. Section III describes the details of the proposed method. Experiments and discusses are respectively presented in Section IV and Section V. The conclusion is given in Section VI.

II. RELATED WORK
A. IMAGE REGISTRATION

Much work has been done in medical image registration. Traditionally, the intensity-based methods directly operate on the image intensity values to search one transformation that minimizes (or maximizes) a criterion (such as mutual information and correlation coefficient) measuring the intensity resemblance of corresponding pixels in two images [2]. However, this type of method usually needs a large number of iterations to evaluate the similarity measure, leading to a high computational cost. The feature-based methods select and match suitable features (such as geometric shape, contour, and landmarks) between images to be registered to estimate the ground truth transformation model [25]. These traditional methods are simple to implement but heavily rely on handcrafted features, which has poor generalization ability to multimodal image registration.

Since recently, deep learning methods have attracted researchers’ attention and been widely used in medical image registration. Wu et al. [26] applied a stacked auto-encoders (CAE) to extract high-level image features for accurate image registration. However, this method is not end-to-end, and the extracted features still rely on other conventional feature-based registration methods. Learning similarity metrics between multimodal images through CNNs was proposed by Simonovsky et al. [27]. This learned metric outperforms mutual information by a large margin in the task of discriminating image patches from different modalities as aligned or misaligned. Miao et al. [18] formulated rigid registration as a regression task and used CNN to predict the transformation matrix. Reference [19] proposed a self-supervised affine image registration network (AIRNet) that directly predicts the affine transformation parameters. AIRNet uses a densely connected CNN structure, which is composed of two pathways that share the same parameters to capture discriminative features between different images.

There is another line of approaches to handle registration tasks, which is inspired by deep reinforcement learning. For instance, Liao et al. [1] reformulated the rigid registration as a greedy decision-making problem, learning to predict optimal actions in a supervised manner. However, the greedy strategy may be limited to local optima. Ma et al. [6] used a variant of the Q-learning algorithm, dueling deep Q-network (DDQN) [32], to train the artificial agent to register multimodal medical images like human experts. This approach is free to search the parameter space during training but suffers time-consuming. Our previous work [16] introduced a parallel CNN+LSTM architecture and a landmark error (LME) based reward function, which overcame the problem of Ma’s and improved the robustness of multimodal image registration. However, the conventional LSTM does not have the flexibility in capturing spatial contextual information between the consecutive frames, and the computation of LME is rather complicated and hence aggravates the burden of RL training.

B. REINFORCEMENT LEARNING

Reinforcement learning is about an artificial agent interacting with environments, trying to learn a strategy by trial-and-error over a sequence of discrete time steps. At each time step \( t \), with a received state \( s_t \) from environment \( \mathcal{E} \), the agent takes an action \( a_t \in \mathcal{A} \) following a policy \( \pi \), receives an immediate numerical reward \( r_t \in \mathcal{R} \) returned by the environment and steps to the next state \( s_{t+1} \) with respect to the state transition probabilities \( \mathcal{P}(s_{t+1}|s_t, a_t) \). The policy \( \pi \) is often defined as a mapping function from current state to the probabilities of action selection. The return \( R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \) is a sum of discounted future rewards from state \( s_t \) with a discounting factor: \( \gamma \in (0, 1] \). The goal of the agent in RL is to seek an optimal policy \( \pi^* \) to maximize \( R_t \). Typically, RL methods can be divided into two types: value-based and policy-based.

The action value \( Q^\pi(s, a) \) represents the expected return for selecting action \( a \) in state \( s \) following policy \( \pi \). Value-based methods iteratively update the \( Q \) value to predict the expected cumulative reward following the Bellman equation: \( Q^\pi(s, a) = E_{\pi \sim \pi}[r + \gamma \max Q^\pi(s', a')|s, a] \), such as Q-learning and Sarsa [8]. And as a breakthrough in combining with deep learning techniques, DQN demonstrated its potential about mastering difficult control policies for Atari games [7]. Thereafter, variety extensions of value-based algorithms has been proposed to improve the stability and efficiency, such as Double DQN [30], Prioritized experience replay [31] and Dueling DQN [32].

In contrast to value-based methods, policy-based methods directly learn a parameterized policy \( \pi_\theta \) by performing gradient ascent on a scalar performance measure \( J(\theta) \). An example of this kind of method is the REINFORCE algorithm [8] which updates the parameters \( \theta \) in the direction \( (R_t - b_1(s_t))\nabla_\theta \log \pi(a_t|s_t) \), where \( b_1(s_t) \) denotes a baseline used to reduce the variance. One common used baseline is the learned value function of the state: \( V(s) \). Notice that the action value \( Q(s, a) \) is an estimate of \( R_t \), then the approximate advantage of the action \( a \) in state \( s \) could defined as \( A(s, a) = Q(s, a) - V(s) \). Then this approach is actually an actor-critic architecture where the policy \( \pi \) is called the actor and the baseline \( b_1 \) is called the critic [8].

III. METHOD

The goal of our method is to automatically align medical images across modalities. The fixed image and the moving image are represented by \( I_f \) and \( I_m \) respectively. The mapping from \( I_f \) to \( I_m \) is a rigid spatial transformation \( T \) parameterized by \( \{t_x, t_y, s, \theta\} \). \( T \) maps a point \( (x_1, y_1) \) of the moving image
to a point \((x_2, y_2)\) of the fixed image as follows:

\[
\begin{pmatrix}
  x_2 \\
  y_2
\end{pmatrix} =
\begin{pmatrix}
  t_x \\
  t_y
\end{pmatrix} +
\begin{pmatrix}
  \cos \theta - \sin \theta \\
  \sin \theta \cos \theta
\end{pmatrix}
\begin{pmatrix}
  x_1 \\
  y_1
\end{pmatrix}
\]

(1)

where \([t_x, t_y]\) denotes translation, \(s\) denotes scaling, and \(\theta\) denotes rotation. We now reformulate image registration as a decision task of taking a series of sequential actions to align the images, and in this way, the optimal transformation \(T^*\) is found. The registration environment is described in Section III-A. We train the spatiotemporal agent (see Section III-B for the agent architectures) from interactions with the environment to learn a registration policy, which selects the alignment action with the highest probability from a certain state at each time step \(t\) using MC rollout (see Section III-C). An overview of the framework is shown in Fig 3. The following sections elaborate on the proposed registration method.

A. ENVIRONMENT

The environment consists of a fixed image, ground truth moving image, moving image, and ground truth transformation matrix. In every episode, the environment randomly picks a pair of the fixed image and the ground truth moving image and then randomly generates a transformation matrix \(T_0\), which applies to the ground truth moving image to generate a moving image as the current training data. The goal of the agent in this environment is to align the generated moving image to the fixed image sequentially.

1) STATE AND ACTION

The state \(s_t\) is represented by a stacked tensor of \(I_F\) and \(I_m\), where \(I_F\) is the fixed image to be registered over a whole trajectory and \(I_m\) is the transformed moving image by \(T_i \circ I_m\). When received a state, the agent would choose an action \(a_t\) from the action space \(A\), which consists of 8 candidate transformations that include translations in both \(x\) and \(y\) directions with \(\pm 1\) pixel, rotation with \(\pm 1^\circ\) and scaling with \(\pm 0.05\). Fig. 3 illustrated the details of the state and action space. Specifically, after action selection, the transformation matrix \(T_i\) in time \(t\) updates to \(T_{i+1}\) by adjusting the corresponding parameters, and the environment updates to next state w.r.t. the state transition \(T((I_F, I_m), a_t) = (I_F, T_i \circ I_m)\). In this paper, we assume that the state is fully-observable, so that the RL framework is further formulated as a standard Markov Decision Process (MDP).

2) REWARD

Reward function \(R(s_t, a_t)\) reflects the expected return of action \(a_t\) taken in a certain state \(s_t\). It provides an immediate feedback for the RL agent to make decisions. A matrix-based reward function used in work [6] is inversely proportional to the Euclidean distance between \(T_g\) and \(T_t\). However, in this way, the unit discrepancy between scaling and the other two parameters in the transformation matrix might cause erratic training. Although the landmark error (LME) reward function helps convergence [16], searching the key-points using the scale-invariant feature transform (SIFT) [22] is burdensome and make [16] hard to extend to 3D registration.

In rigid registration, the transformation \(T\) is applied to the entire moving image, including the points in the background. We propose a fixed points reward (FPE) that only considers the coordinates of the points without concerning the corresponding pixel values or whether they are in the background area. Specifically, we alternatively pre-choose a group of fixed points for all training image pairs. These fixed points \(P_G\) are uniformly selected across the ground truth of moving image, and then they are warped using the perturbation transformation matrix \(T_0\) and transformed back using \(T_i\). We define the reward function with fixed points

FIGURE 3. An overview of the proposed asynchronous framework. (a) A single global model shares network parameters and is asynchronously updated by all independent threads. (b) The details of the interaction between agent and environment on each thread.
error (FPE) as:

\[ R(s_t, a_t) = -D = -\frac{1}{\#(P_G)} \sum_{i \in G} ||p_i - T_t \circ \tilde{p}_i||_2 \] (2)

where \( p_i \) and \( \tilde{p}_i \) are a pair of fixed points from the ground truth and the transformed moving image respectively, \( \#(P_G) \) calculates the number of fixed points. The default number of fixed points is set to 100 (a grid of 10 rows and 10 columns), and the performance of different numbers of fixed points is discussed in section V-C. Especially, the FPE is presented by \( D \), which is the average Euclidean distance between ground truth and transformed fixed points. We assume that the registration terminal is triggered when \( D \) is smaller than a threshold.

### B. RL FRAMEWORK

#### 1) A3C ALGORITHM

Our method is derived from A3C [9], which is a multi-thread asynchronous variant of a popular RL model actor-critic. Instead of using experience replay as in [6], we asynchronously apply multiple agents with the same network model to multiple instances of the registration environment. Our A3C model can run on a single machine with a multi-core CPU and perform much faster than most of the improved DQN methods [9]. Notably, our algorithm operates in a forward view via multi-step learning. Both the policy and value function are updated after every \( t_{\text{max}} \) steps or when the agent reached the terminal state. The estimate of the advantage function, \( A(s_t|a_t; \theta_v) \), is given by \( \sum_{i=0}^{k-1} \gamma^i r_{t+i+1} + \gamma^k V(s_{t+k}; \theta_v) - V(s_t; \theta_v) \), where \( k \) is the learning steps upper-bounded by \( t_{\text{max}} \) in each thread. The parameters of the policy function and value function are updated by:

\[ \theta \leftarrow \theta + \eta \nabla_\theta \log \pi(a_t|s_t; \theta) A(s_t|a_t; \theta_v) + \beta \nabla_\theta H(\pi(s_t; \theta)) \] (3)

\[ \theta_v \leftarrow \theta_v - \eta \nabla_{\theta_v} (A(s_t|a_t; \theta_v))^2 \] (4)

where \( \eta \) is the learning rate. Note that \( H(\cdot) \) is the entropy of policy \( \pi \), which is used in the objective function to improve the algorithm’s robustness and stability, and the hyperparameter \( \beta \) controls the strength of the regularization term. Typically, the policy network and value network are designed to share all layers except the output.

#### 2) NETWORK ARCHITECTURE

The main network architecture of the proposed spatiotemporal agent is shown in Fig. 4. The input of our model is a 2-channel image comprised of a CT and an MR. We construct the network with a CNN part to extract features from the received state and a ConvGRU part to encode both spatial and temporal information from neighboring frames [11]. In this work, we mainly focus on GRU since it has shown superior performance to LSTM in many temporal sequential learning tasks and with a lower memory requirement. However, traditional GRU is insufficient to extract spatial dependence. To this end, by replacing traditional affine operation with convolution operation in both the input-to-state and...
state-to-state transitions of GRU, as shown in Fig. 6, we leverage the ConvGRU architecture to model image sequence.

In our CNN-ConvGRU architecture, the CNN part is composed of 10 convolutional layers, each of which is followed by an exponential linear unit (ELU) activation, as shown in Fig. 5. We perform downsampling by directly using a convolutional layer with a stride of 2. Note that layer normalization is employed after every downsampling. Using layer normalization helps to improve both the training efficiency and the generalization performance for the recurrent part. Following the convolutional layers, the resulting feature maps for each of the recent time step are fed into the convolutional GRU part to extract both temporal information and spatial correspondence. The network ends with a global average pooling layer followed by two branches of FC layers: one represents the policy function $\pi(\cdot|s_t; \theta)$ with softmax, and the other represents the state-value function $V(s_t; \theta_s)$. We update the value network and the policy network simultaneously by combining Eq. 3 and Eq.4 into a final loss as follows:

$$L = L_\pi + c_V L_V$$

where $L_\pi$ is the policy loss, $L_V$ is the value loss and $c_V$ is a constant coefficient which controls the balance between the two losses above.

C. MONTE CARLO ROLLOUT

After training, the registration agent applies the learned policy to the unknown test dataset. The registration process will be stopped if the predicted state-value $v_t$ has achieved the threshold $tr_t$, which nears the terminal reward set in the training phase. Nevertheless, in practice, we have observed that the predicted transformation parameters often jitter around specific values, so that the termination is hard to reach. To handle this nonstationarity, we adopt a Monte Carlo method, which is a simple extension of importance sampling, to simulate multiple searching paths so as to improve the accuracy of transformation matrix. Given a starting state $s_t$, whose state value $v_t$ nears $tr_t$, the agent explores $N$ trajectories within a fixed searching depth $D_{mc}$ simultaneously. More specifically, in each trajectory, the action for each node is selected following the policy $\pi(\cdot|s_t; \theta)$, and the corresponding state-value is generated by the value network. Let $T_i$, parameterized by $[x_{ti}, y_{ti}, s_i, \theta_i]$, denotes the final transformation matrix of trajectory $i$ and $V_i = \sum_{j=0}^{D_{mc}} v_{ij}$ denotes the total value for this trajectory, where $v_{ij}$ is the value of the $j$th node at trajectory $i$. The final estimation of transformation parameters are obtained by a weighted averaging:

$$[x_t, y_t, s, \theta] = \frac{1}{\sum_{i=1}^{N} V_i} \sum_{i=1}^{N} V_i \cdot [x_{ti}, y_{ti}, s_i, \theta_i]$$

Note that by performing Monte Carlo sampling, the final predicted matrix is an approximation of the ground truth, which avoids the jittering around a particular value.

IV. EXPERIMENTS

A. DATASET

We evaluate the effectiveness of our proposed framework on a 2D multimodal registration dataset. This experimental dataset was collected from 99 patients (male/female: 66/33; age range: 21-76 years) diagnosed as nasopharyngeal carcinoma, each with 100 paired 3D MR and CT slices. These patients underwent radiotherapy at West China Hospital, where they received a CT scan on a Siemens SOMATOM Definition AS+ system and an MRI scan on a Philips Achieva 3T scanner. The original resolutions of CT images range from $0.88 \times 0.88 \times 3.0\text{mm}^3$ to $0.97 \times 0.97 \times 3.0\text{mm}^3$ and the original resolution of MR images is $0.61 \times 0.61 \times 0.8\text{mm}^3$. All patients were fully informed and consented to participate in the study. Our study protocol was fully approved by the radiology department of West China Hospital.

After removing invalid backgrounds, all CT and MR images were resampled to the same resolution of $1 \times 1 \times 1\text{mm}^3$. In our experiment, the MR was treated as the fixed image and the CT was the moving image. Particularly, the pre-registration of CT and MR pairs from the same patient was realized by 3D Elastix [14] with default parameters. In light of the strong correlation between adjacent slices, we selected only six slices out the 100 for each modality to represent considerable variations between images. Since this paper mainly focuses on investigating the feasibility of using DRL for multimodal registration on 2D images, the structural smoothness of slices along the third dimension is not considered at the moment. Finally, our experiment obtained 594 aligned pairs of MR-CT, with 474 image pairs selected from 79 patients as the training data, 120 image pairs selected from the rest 20 patients as the testing data.

B. TRAINING AND TESTING

In each episode at the training procedure, the rigid-body perturbation of the moving image is randomly generated within translation $[-25, 25] \text{ pixels}$ for both $x$ and $y$ axes, rotation $[-30^\circ, 30^\circ]$ and scaling $[0.75, 1.25]$. The network is trained with eight parallel threads, each running 24000 episodes,
and finally saw 192,000 pairs of training images. We use a discount factor of 0.99, an entropy coefficient of 0.01, and a bootstrap frequency of 32 steps to update the network weights. Our model is trained by Adam optimizer with a learning rate of 0.00001 and adopts MSRA initialization [15] for all weights. The episode is terminated whenever the agent receives a terminal signal (FPE less than 1) or reaches the maximum number of steps (set to 500). We trained our agent for 2.5 days using 8 CPU cores and a Tesla P100 Pascal GPU.

We evaluate the method on two test data sets (E1 and E2), both consisted of 120 image pairs with different moving image perturbation distribution ranges. More specifically, the moving images of “E1” dataset are perturbed within the same perturbation scope as the training data, and the moving images of “E2” dataset are perturbed with a larger range of translation $[-30, 30]$ for both $x$ and $y$ axes, rotation $[-45^\circ, 45^\circ]$ and scaling $[0.75, 1.25]$. MC rollout is applied to all testing, where the number of simulating branches $N_{mc} = 20$, searching depth $D_{mc} = 10$, and the stopping threshold $tr$ is 10. The details of the proposed model in testing procedure is shown in Fig. 7.

### C. COMPARING WITH OTHER METHODS

We evaluated the efficiency of our proposed model by comparing it with several state-of-the-art methods on dataset E1 and E2, respectively. For most methods, we use the default parameter settings as recommended in their corresponding papers or the readme of some public registration toolkits. The Scale Invariant Feature Transform (SIFT) [33] is one of the most popular techniques for rigid image registration. We set a sift distance ratio of 0.8 to match key-points and use the Random Sampling Consensus (RANSAC) algorithm to improve the results. For Elastix, mattes mutual information with default setting is used. We use Elastix 2D as the registration gold standard since the ground truth moving image was generated by Elastix 3D in the preprocessing stage, as mentioned in Section IV-A. The pure-SL method rephrases image registration as a regression task, inputting MR-CT pairs and outputting transformation parameters: $[t_x, t_y, s, \theta]$. For a fair comparison, the pure-SL model contains the same CNN architecture as in our proposed RL model. Similarly, AIRNet [19] learns to directly predict the affine transformation matrix in a self-supervised manner. It uses a densely connected CNN structure, which is composed of two pathways that share the same parameters to capture discriminative features between different images. Furthermore, we compared the proposed method with our previous work [16] and another RL model based on Deep Recurrent Q-Network (DRQN) [29]. For the DRQN method, since it is designed for the partially observable games, we made some minor changes to their code to fit our registration environment and dataset. And we implement AIRNet and Sun’s work by using default settings and optimizations recommended by their authors. Note that SIFT and Elastix run on the CPU, and all DL-based and DRL-based models work on the same GPU as our proposed method.

In all experiments, we employee target registration error (TRE) [20] as the quantitative measure, which is defined as the Euclidean distance of corresponding points between the predicted image and ground truth:

$$V_{TRE} = \frac{1}{N} \sum_{i=1}^{N} \| p_i - T \circ \tilde{p}_i \|_2$$  \hspace{1cm} (7)$$

where $p_i$ denotes the $i$-th landmark point on the ground truth moving image, $\tilde{p}_i$ denotes the corresponding point in the moving image, $T$ denotes the predicted affine transformation matrix, $N$ is the number of points and $\circ$ represents the align operator.
D. EVALUATION RESULTS

All methods were evaluated on the test datasets E1 and E2, and the quantitative accuracy among all methods is summarized in Table 1. It is obvious that all deep learning based methods are far better than SIFT, and all RL based methods achieved higher registration accuracy than pure-SL and AIRNet. We observe that the proposed method surpasses Elastix 2D registration by a large margin and performed a more robust and reliable 2D registration. By using the convolutional recurrent architecture, the proposed model is significantly improved, compared with DRQN and our earlier RL model [16] which only leverage the fully connected recurrent architecture. Moreover, as reflected in Table 1, our proposed method is the only approach that has a TRE value smaller than one on dataset E1 and even E2, the one with a larger range of perturbation. This indicates that all moving images were perfectly aligned to fixed images. Moreover, our model is also the only method whose performance on E2 is slightly better than E1, which implies the proposed model has the best generalization ability among all methods. A qualitative analysis of registration results is visualized in Fig. 8. We can see that the CT images registered by our method are more robust and accurate.

V. DISCUSSION

A. EFFECTIVENESS OF SPATIO-TEMPORAL NETWORK

Our previous work in [16] incorporated a fully-connected long short-term memory (FC-LSTM) to extract the correlated temporal features among consecutive frames. The FC-LSTM is a common structure in sequence learning for modeling long-term dependencies. While GRU is a newer generation of RNN which has a similar performance to LSTM but using a simpler and lighter structure. In the context of image registration, spatial information is as important as temporal information and should be taken into account. Traditional fully-connected recurrent architectures were hardly to encode spatial features and powerless to solve spatial dependencies. In regard to this, we introduce the convolutional GRU (ConvGRU) in our model to take spatial information into consideration, which improves the accuracy of the registration results. Although a convolutional operation is used, has a smaller computational cost than most fully-connected recurrent architectures [35]. Table 2 compares the proposed CNN-ConvGRU with CNN-GRU as well as pure CNN architecture. Note that for a fair comparison of these networks, we set the same ten layer CNN structures for all models and all of them are trained with the LME reward in the RL training.

B. MODEL ARCHITECTURE ANALYSIS

A CNN-ConvGRU architecture is proposed in this work, where a deeper CNN structure is used to extract image features that contain high-level contextual information, and the convolutional GRU network is leveraged to extract the hidden information among consecutive frames in our RL algorithm. It has been proven in many experiments that deeper networks lead to better results since a large receptive field and more semantic information are obtained. Experiments of whether a deeper architecture is beneficial to our RL-based image registration are shown in Table 3. It is obvious that the registration...
TABLE 2. The effectiveness of various network architectures. The CNN refers to a separate CNN model, the CNN-GRU means incorporating a GRU unit into the CNN model, and the CNN-ConvGRU means incorporating a ConvGRU unit into the CNN model.

| Methods          | E1          | E2          |
|------------------|-------------|-------------|
|                  | mean±std    | 50th        | 90th        | mean±std    | 50th        | 90th        |
| CNN              | 1.03±0.77   | 0.83        | 1.71        | 1.05±0.86   | 0.90        | 1.65        |
| CNN-GRU          | 1.09±0.65   | 0.98        | 1.71        | 1.07±0.80   | 0.93        | 1.68        |
| CNN-ConvGRU      | 0.93±0.71   | 0.78        | 1.47        | 0.96±0.65   | 0.82        | 1.49        |

TABLE 3. A comparison of the accuracy of kinds of network architectures. The $N$-CNN-RNN means using $N$ layers convolutional network and a type of RNN unit.

| Methods          | E1          | E2          |
|------------------|-------------|-------------|
|                  | mean±std    | 50th        | 90th        | mean±std    | 50th        | 90th        |
| 3-CNN-LSTM       | 1.39±1.45   | 0.98        | 2.59        | 1.66±2.30   | 1.08        | 3.17        |
| 10-CNN-LSTM      | 1.22±2.79   | 0.87        | 1.55        | 1.06±0.57   | 0.97        | 1.62        |
| 10-CNN-ConvLSTM  | 0.93±0.67   | 0.78        | 1.54        | 1.02±0.67   | 0.83        | 1.64        |
| 3-CNN-GRU        | 1.21±0.93   | 0.91        | 2.28        | 1.37±1.13   | 0.98        | 2.85        |
| 10-CNN-GRU       | 1.09±0.63   | 0.98        | 1.71        | 1.07±0.80   | 0.93        | 1.68        |
| 10-CNN-ConvGRU   | 0.93±0.71   | 0.78        | 1.47        | 0.96±0.65   | 0.82        | 1.49        |

FIGURE 9. Learning curves of training based on different reward functions.

result of a deeper CNN structure (10 layers) is significantly improved compared to the shallow structure (3 layers) in all recurrent models. For a better illustration, we also compared the performance between two models with LSTM structure and GRU structure respectively. The results shown in Table 3 indicate that the model with the GRU structure is slightly better than the model with the LSTM structure in our registration scenario.

C. REWARD ANALYSIS

The reward function is an essential component in RL and it directly affects the choice of actions. As mentioned in Section III-A.2, the commonly used reward functions intuitively rely on the prior knowledge, such as the ground truth transformation matrix. This section discusses how reward functions effect registration training and testing, and compares the proposed fixed-points reward (FPE) with LME reward [16] as well as the matrix-based reward. The comparisons of the learning speed between three kinds of rewards are illustrated in Fig 9. We notice that RL using matrix-based rewards converges slightly slower and is in fact very unstable. The training using LME reward is a little faster than FPE reward. However, FPE reward based training uses fixed coordinate points, eliminating the need to calculate SIFT points for all CT images. FPE reward is more elegant and more accessible to extend to 3D registration considering the computational efficiency. Furthermore, the registration results, as shown in Table 4 indicate that the RL model using FPE reward has the highest registration accuracy.

Moreover, we also analyzed the performances of the different number of fixed points in FPE. The experiment shown in Table 5 compares three RL models using FPE reward but with numbers of 10, 100 and 400 fixed points, respectively. The result indicates that the model using 100 fixed points is the best choice for both E1 and E2. The reason may be that 10 fixed points are not sufficient enough to represent the textural information of images to be registered, while 400 fixed points are quite dense for our images, which have a small resolution of $168 \times 168 \text{mm}^2$. Moreover, an RL model using a sparse reward is also compared. More specifically, the sparse reward is set to positive 1 when the moving image is successfully registered (the episode is terminated) and zero otherwise. However, as reflected in Fig. 9, the sparse reward based model fails to converge in our experiment.

D. MONTE CARLO ROLLOUT IN REGISTRATION INFEERENCE

After training, the agent applies the learned strategy to register multimodal images. We consider that the registration is completed when the predicted state value $v_t$ achieves a threshold $r$, which is the same as the terminal reward in training. However, in the testing phase, the transformation parameters may jitter around specific values, and it is hard to reach the termination. To this end, we propose to use the MC Rollout algorithm to handle this problem. In the process of MC, the agent is required to explore $N$ trajectories within a fixed searching depth $D_{mc}$ simultaneously from a certain
TABLE 4. Registration results of our model trained with different reward functions.

| rewards       | E1 mean±std | 50th | 90th | E2 mean±std | 50th | 90th |
|---------------|-------------|------|------|-------------|------|------|
| matrix-based  | 1.14±0.91   | 0.90 | 1.77 | 1.25±1.16   | 0.87 | 2.40 |
| LME           | 0.93±0.71   | 0.78 | 1.47 | 0.96±0.65   | 0.82 | 1.49 |
| FPE           | 0.86±0.63   | 0.68 | 1.48 | 0.85±0.70   | 0.72 | 1.28 |

TABLE 5. The effectiveness of different number of fixed points. FPE-N represents the model trained in FPE reward with a number of N fixed points.

| Methods | E1 mean±std | 50th | 90th | E2 mean±std | 50th | 90th |
|---------|-------------|------|------|-------------|------|------|
| FPE-10  | 1.01±0.84   | 0.81 | 1.50 | 0.93±0.86   | 0.81 | 1.44 |
| FPE-100 | 0.86±0.63   | 0.68 | 1.48 | 0.85±0.70   | 0.72 | 1.28 |
| FPE-400 | 0.89±0.61   | 0.75 | 1.56 | 0.88±0.61   | 0.73 | 1.51 |

TABLE 6. Registration results of the proposed model uses or does not use Monte Carlo Rollout in the test.

| Methods            | E1 mean±std | 50th | 90th | time(s) | E2 mean±std | 50th | 90th | time(s) |
|--------------------|-------------|------|------|---------|-------------|------|------|---------|
| ConvGRU w/o MC     | 0.93±0.71   | 0.78 | 1.47 | 1.96    | 0.96±0.65   | 0.82 | 1.49 | 2.06    |
| ConvGRU w/ MC      | 1.22±0.74   | 1.06 | 1.81 | 2.53    | 1.21±0.75   | 1.09 | 1.95 | 3.00    |

TABLE 7. A comparison of the effectiveness of our model uses or does not use layer normalization in the CNN part.

| Methods            | E1 mean±std | 50th | 90th | E2 mean±std | 50th | 90th |
|--------------------|-------------|------|------|-------------|------|------|
| ConvGRU w/o LN     | 0.93±0.71   | 0.78 | 1.47 | 0.96±0.65   | 0.82 | 1.49 |
| ConvGRU w/ LN      | 1.17±0.86   | 0.96 | 1.65 | 1.40±3.21   | 0.92 | 1.76 |

state, as elaborated in Section III-C. Note that the MC is only activated when the predicted state value nears $r_t$. The comparison between RL models with and without using MC is shown in Table 6. The performance of the model using MC is significantly improved. It is worth noting that MC sampling is a computational burden but leads to a more robust and accurate registration performance. Interestingly, we found in our experiment that the average registration time for an RL model using MC is shorter than the one without MC. This is because the agent might not be able to access the terminal threshold $r_t$ on some testing images, and if no MC is used, the agent has to explore 500 steps for the registration of those images, which slows down the registration process.

E. THE IMPORTANCE OF LAYER NORMALIZATION

A proper normalization in deep learning can be a crucial factor in the effective training of the deep neural networks. Batch Normalization (BN) is adapted in most neural networks in computer vision problems to normalize inputs to accelerate convergence. However, the proposed RL model does not accept mini-batch inputs but has a single tensor composed of a stacked fixed and moving images. In this regard, we use layer normalization (LN) [21] instead to normalize all features in a layer for a single input. This work redefines image registration as a sequence learning task, which is also very suitable for using LN. The comparison of our RL model with or without using LN is shown in Table 7. It is clear that LN plays an important role and provides a significant improvement in registration accuracy.

VI. CONCLUSION

In this paper, 2D registration is redefined as a 3D sequence learning task. We have presented a multimodal registration approach based on a spatiotemporal agent. The agent learns the registration policy using the A3C algorithm with an FPE reward. Both spatial and temporal information is taken into account in our network by leveraging both CNN and ConvGRU layers. This leads to enhanced performance in the registration task. Experimental results demonstrate that our approach outperforms several state-of-the-art methods.

In addition, we analyzed the influence of the network structure, reward function, Monte Carlo and layer normalization on registration.

One of our future work is to extend the current framework to 3D rigid registration and the other work is RL-based 2D deformable registration. For 3D rigid registration, the curse of dimensionality may cause over-fitting. To address this problem, we plan to incorporate residual blocks and attention mechanisms into our network to improve the learning process. For deformable registration, a large number of parameters in the deformation field is a big challenge, which makes the optimization process extremely difficult. We would like to use pre-trained networks to assist RL learning and apply dimension reduction to the action space.
REFERENCES

[1] R. Liao, S. Miao, P. de Tournemire, S. Gribc, A. Kamen, T. Mansi, and D. Comaniciu, “An artificial agent for robust image registration,” in Proc. 21st AAAI Conf. Artif. Intell., Feb. 2017.

[2] G. P. Penney, J. Weese, J. A. Little, P. Desmedt, and D. L. Hill, “A comparison of similarity measures for use in 2-d-3-d medical image registration,” IEEE Trans. Med. Imag., vol. 17, no. 4, pp. 580–595, Aug. 1998.

[3] Y. Kita, L. L. Wilson, and J. A. Noble, “Real-time registration of 3D cerebral vessels to X-ray angiograms,” in Medical Image Computing and Computer-Assisted Intervention–MICCAI. Springer, 1998, pp. 1125–1133.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of go with deep neural networks and tree search,” Nature, vol. 529, no. 7587, pp. 484–494, Jan. 2016.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[6] K. Ma, J. Wang, V. Singh, B. Tampersoy, Y.-J. Chang, A. Wimmer, and D. Silver, “Asynchronous methods for deep reinforcement learning,” in Proc. 21st AAAI Conf. Artif. Intell., Feb. 2017.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” 2013, arXiv:1312.5602. [Online]. Available: http://arxiv.org/abs/1312.5602

[8] R. S. Sutton and A. G. Barto, Introduction to reinforcement learning, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[9] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in Proc. Int. Conf. Mach. Learn. Comput. Assist. Intervent. Springer, 2017, pp. 240–248.

[10] T. Lindeberg, “Scale invariant feature transform,” Scholarpedia, vol. 7, no. 5, 2012, Art. no. 010491.

[11] M. Siam, S. Valipour, M. Jagersand, and N. Ray, “Convolutional gated recurrent networks for video segmentation,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2017, pp. 3090–3094.

ZIWEI LUO received the B.Eng. degree in transportation from the Hebei University of Technology, China, in 2015. He is currently pursuing the M.S. degree with the School of Computer Science, Chengdu University of Information Technology. From 2015 to 2017, he was a Software Engineer with Tianjin Mini Technology Company, Ltd. His current research interests include artificial intelligence, machine learning, and computer vision.

XIN WANG received the Ph.D. degree in computer science from the State University of New York, University at Albany, in 2015. He is currently a Senior Machine Learning Scientist of CuraCloud Corporation. His current research interests include artificial intelligence, machine learning, and computer vision.

XI WU (Member, IEEE) received the B.S. degree in communication engineering from Sichuan University, Chengdu, China, in 2003, the M.S. degree in communication engineering from the University of Electronic Science and Technology of China, Chengdu, in 2006, and the Ph.D. degree in computer science from Sichuan University, in 2012. He is currently a Professor with the Department of Computer Science, Chengdu University of Information Technology, Chengdu. His current research interests include image processing and computer vision.
YOUBING YIN received the Ph.D. degree from The University of Iowa. He was a Senior Research Scientist of VIDA Diagnostics. He is currently the Vice President of research and development with CuraCloud Corporation. His research interests include the development and commercialization of machine learning and imaging-based solutions in healthcare. He is familiar with strict developing procedures in this highly regulated industry (FDA, CE, and CFDA). He was a Key Developer of an FDA approved quantitative lung imaging software, which received the Best Product Award from the European Respiratory Society.

KUNLIN CAO received the Ph.D. degree in electrical and computer engineering from The University of Iowa, USA. She was a Lead Scientist of the Biomedical Image Analysis Laboratory, GE Global Research Center. She has over ten years of research experience in medical image analysis, and extensively involved in image segmentation, image registration, feature extraction, functional information quantification, motion detection, and object localization/tracking for real-time image-guided applications. She was with CuraCloud Corporation, involved in a variety of research projects and product development in the field of a quantitative assessment of anatomical and functional changes toward precise disease diagnosis/prognosis/treatment planning using artificial intelligence technologies. She published over 60 articles in international journals and conferences in the medical image analysis area, which were cited more than 1000 times. Her current research interests include multidimensional image and signal processing, medical image analysis (CT, ultrasound, MR, and PET), artificial intelligence, and computer vision.

QI SONG received the bachelor’s degree from the University of Electronic Science and Technology, in 2003, the master’s degree from Tsinghua University, in 2006, and the Ph.D. degree from The University of Iowa, USA, in 2011. He was a Research Assistant with The University of Iowa, in 2011. He became a Scientist of the Global Research and Development Center, General Electric Company, New York, in 2014. He was a Senior Scientist of HeartFlow Corporation, USA, from 2014 to 2015, and a Founder and the CEO of CuraCloud Corporation, Seattle, USA, from 2016 to 2017. Since 2017, he has been the Founder and the General Manager of Shenzhen LE Medical Technology Company Ltd.

JING HU (Member, IEEE) received the bachelor’s degree from the University of Electronic Science and Technology of China (UESTC), in 2009, and the Ph.D. degree from Tsinghua University, in 2015. She is currently an Associate Professor with the Chengdu University of Information Technology. Her current research interests include image processing, deep learning, and reinforcement learning and their applications.