TR-GAN: Multi-Session Future MRI Prediction With Temporal Recurrent Generative Adversarial Network

Chen-Chen Fan, *Graduate Student Member, IEEE*, Liang Peng, Tian Wang, Hongjun Yang, Xiao-Hu Zhou, *Member, IEEE*, Zhen-Liang Ni, Guan'an Wang, Sheng Chen, Yan-Jie Zhou, *Graduate Student Member, IEEE*, and Zeng-Guang Hou, *Fellow, IEEE*

***Abstract*— Magnetic Resonance Imaging (MRI) has been proven to be an efficient way to diagnose Alzheimer's disease (AD). Recent dramatic progress on deep learning greatly promotes the MRI analysis based on data-driven CNN methods using a large-scale longitudinal MRI dataset. However, most of the existing MRI datasets are fragmented due to unexpected quits of volunteers. To tackle this problem, we propose a novel Temporal Recurrent Generative Adversarial Network (TR-GAN) to complete missing sessions of MRI datasets. Unlike existing GAN-based methods, which either fail to generate future sessions or only generate fixed-length sessions, TR-GAN takes all past sessions to recurrently and smoothly generate future ones with variant length. Specifically, TR-GAN adopts recurrent connection to deal with variant input sequence length and flexibly generate future variant sessions. Besides, we also design a multiple scale & location (MSL) module and a SWAP module to encourage the model to better focus on detailed information, which helps to generate high-quality MRI data. Compared with other popular GAN architectures, TR-GAN achieved the best performance in all evaluation metrics of two datasets. After expanding the Whole MRI dataset, the balanced accuracy of AD vs. cognitively normal (CN) vs. mild cognitive impairment (MCI) and stable MCI vs. progressive MCI classification can be increased by 3.61% and 4.00%, respectively.**

***Index Terms*— Alzheimer's disease, magnetic resonance imaging, generative adversarial network.**

Manuscript received 3 December 2021; revised 13 January 2022; accepted 7 February 2022. Date of publication 11 February 2022; date of current version 1 August 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC2001700; in part by the National Natural Science Foundation of China under Grant 61720106012, Grant U1913601, Grant 62073319, Grant 62003343, and Grant U20A20224; in part by the Beijing Natural Science Foundation under Grant L172050; in part by the Beijing Sci&Tech Program under Grant Z211100007921021; in part by the Youth Innovation Promotion Association of Chinese Academy of Sciences under Grant 2020140; in part by the Strategic Priority Research Program of Chinese Academy of Science under Grant XDB32040000; in part by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health) under Grant U01 AG024904, and in part by the Department of Defense (DOD) ADNI under Grant W81XWH-12-2-0012. *(Corresponding author: Zeng-Guang Hou.)*

Chen-Chen Fan, Zhen-Liang Ni, Guan'an Wang, Sheng Chen, and Yan-Jie Zhou are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: fanchenchen2018@ia.ac.cn; nizhenliang2017@ia.ac.cn; wangguanan2015@ia.ac.cn; chensheng2016@ia.ac.cn; zhouyanjie2017@ia.ac.cn).

Liang Peng, Hongjun Yang, and Xiao-Hu Zhou are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: liang.peng@ia.ac.cn; hongjun.yang@ia.ac.cn; xiaohu.zhou@ia.ac.cn).

Tian Wang is with the Neuroscience and Intelligent Media Institute, Communication University of China, Beijing 100024, China (e-mail: tian_wang@cuc.edu.cn).

Zeng-Guang Hou is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, also with the CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing 100190, China, and also with the CASIA-MUST Joint Laboratory of Intelligence Science and Technology, Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China (e-mail: zengguang.hou@ia.ac.cn).

Digital Object Identifier 10.1109/TMI.2022.3151118

I. Introduction

ALZHEIMER'S Disease is an irreversible neuro-degenerative disease, which is the most common form of dementia, affecting millions of people worldwide. While there is no cure for AD currently, studies have shown that AD can be diagnosed in very early stages when interventions could effectively prevent the deterioration of AD [1]. Neuroimaging techniques can detect brain abnormalities caused by AD [2]. Magnetic resonance imaging (MRI), one of the leading diagnostic modalities, offers excellent spatial resolution and soft-tissue contrast and helps to catch early AD development [3], [4].

Longitudinal MRI datasets are helpful to study the progression of AD by collecting multi-session data from a large number of AD, mild cognitive impairment (MCI), and cognitively normal (CN) people. Unfortunately, these datasets are incomplete due to many participants' midway withdrawal (Fig. 2). Ideally, participants will undergo a series of tests that repeat over several years for a multi-session dataset. However, many participants' data collection stopped at different intermediate stages due to various technical and practical reasons.

As a chronic neuro-degenerative disease, AD gradually affects brain structure. Comparison between MRI data obtained at different periods for the same participant provides better observation of the brain structure changes and contributes to more accurate diagnosis algorithms [5]. To complete

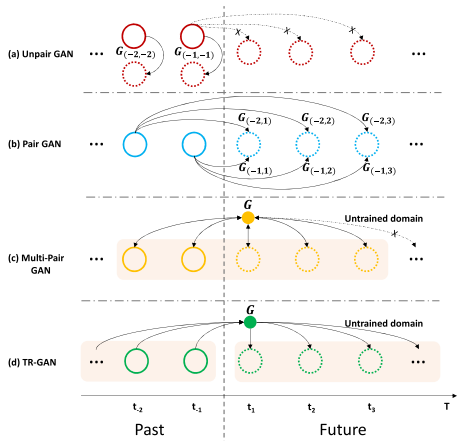


Fig. 1. Comparison of different GAN frameworks in the expansion of longitudinal datasets. *G* represents the generator, the solid arrow indicates the achievable generation direction, and the dashed line indicates that the current model cannot generate the data. (a) Unpair GAN cannot predict future data as no supervision is adopted during training; (b) Pair GAN can perform future one-to-one prediction tasks, but they may require multiple models to handle many-to-many prediction tasks; (c) Multi-Pair GAN can deal with many-to-many prediction tasks by training multiple sessions data but cannot predict untrained domain data; (d) Our proposed TR-GAN can deal with many-to-many predictions with varying inputs and outputs and predict untrained domain data.

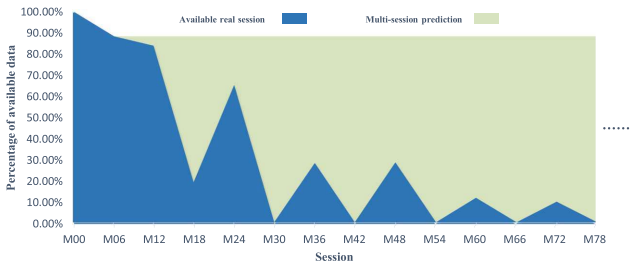


Fig. 2. Available real session and multi-session prediction task in ADNI dataset (Introduced in section IV-A). "M*j*" represents the data that was collected in the *j*-th month after the first collection. The blue area represents the distribution of available sessions, which are fragmented. The multi-session prediction task aims to use the participants' existing session data to generate missing data in the future. The green area is the prediction diagram of [M00, M06] → future session.

the missing dataset, for the first time, we propose the multi-session prediction task, using the existing MRI data of each subject to synthesize its data in multiple future periods (Fig. 2).

Generative Adversarial Networks (GAN) have been used to synthesis 3D MRI images. They can be divided into pair and unpair GAN according to whether paired data is adopted in training. Unpair GANs can generate realistic MRI data [6]–[9]. As no supervision is adopted during training, unpair GAN can only generate data that belong to the same data collection time of its training data, failing to predict future sessions (Fig. 1 (a)). As for pair GANs, they can perform one-to-one future MRI data prediction tasks. Popular conditional GAN architectures like Cycle-GAN [10] and Pix2Pix [11] fail to consider the connection between MRI data acquired at different times and instead treat each data independently (Fig. 1 (b)). LDGAN [12] predicts multiple future data by training multiple models, which training is complex and cannot predict untrained data. StarGAN [13] and CollaGAN [14] try to deal with many-to-many prediction tasks (Fig. 1 (c)) by a single model, but they also cannot predict untrained domain data. Current GAN-based MRI generation methods can only predict future sessions with fixed-length due to the finite domain session training or even fail to generate future data. Synthesizing each subject's data in multiple future periods can complete the missing data and contributes to a more accurate observation of their AD progression. Therefore, we explore the multi-session prediction task (Fig. 1 (d)) to perform many-to-many predictions, even untrained domain data. Specifically, the multi-session prediction task generates missing future session data by effectively using the existing session data of the subjects to complement the fragmented longitudinal dataset.

Several challenges need to be considered when solving this novel task. The first one is labeled data generation. A well-labeled dataset is required for diagnosis classification algorithms based on deep learning. The second one is various input lengths, i.e., inputting different numbers of available 3D MRI data for different participants. The number of 3D MRI data may vary significantly among different participants because of their different data collection times. Neural networks with a fixed number of neurons in the input layer have trouble dealing with varying lengths of 3D MRI sequence. The last one is fine-grained feature extraction. Compared with natural scene images, 3D MRI images have low contrast and high visual consistency, making it hard to distinguish MRI data from a global level. Though 3D MRI scans produce detailed images of the organs and tissues in the body, the fine structure of 3D MRI scans might be intricate for GANs to capture, leading to the loss of essential details.

Based on the above analysis, we propose a Temporal Recurrent Generative Adversarial Network (TR-GAN) (Fig. 3) to solve the three challenges mentioned above. It consists of a SWAP module, a MSL module, a generator, and two discriminators. First, labeled data is generated by conditional GAN, which takes each participant's existing data as prior information and generates future data that share the same label with prior input. Second, recurrent connections are adopted in the generator to deal with variant input lengths. The generator encodes the memory of previous sessions through its hidden state. Third, a SWAP module forces the generator to focus on detailed local information by partitioning the 3D input MRI and shuffling these partitioned local regions in their 3D neighborhood. Meanwhile, a multiple scale & location (MSL) module is proposed to extract multi-scale and multi-location features and send them to the discriminator for pixel-by-pixel discrimination. This forces the generator to focus on multi-scale generation details against the discriminator.

The main contributions of this work can be concluded as follows:

- To the best of our knowledge, we are the first to explore the multi-session prediction task of MRI by using a single generator model, which can predict the participants' multiple future sessions based on the existing sessions.
- TR-GAN is proposed to deal with variant input sequence length and flexibly generate future variant sessions, which

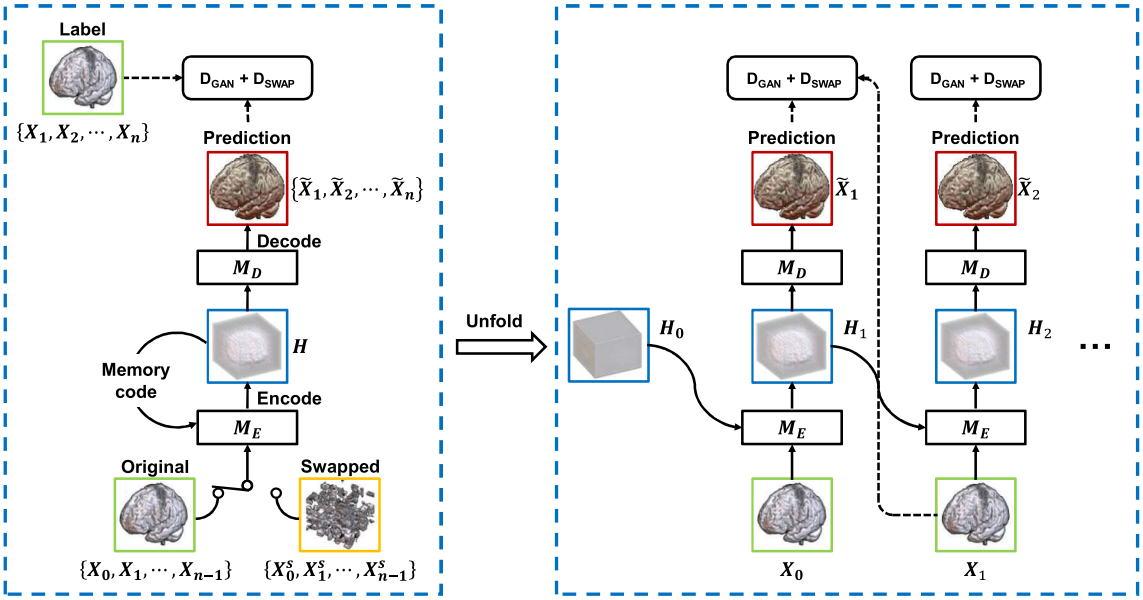


Fig. 3. The framework of TR-GAN. Recurrent connections are adopted in the generator to deal with variant input lengths. Specifically, as shown in the dashed box, $\forall n \in N$, for the input sequence data $\{X_0, X_1, \cdots, X_{n-1}\}$, the information of the varying length data can be encoded into the hidden state H through n times feed forward. In the i-th feed forward, the encoder M_E randomly ($p = 50\%$ probability) takes original X_{i-1} or swapped data X_{i-1}^s and the last hidden state H_{i-1} as input to obtain H_i, which encodes the information of $\{X_0, X_1, \cdots, X_{i-1}\}$. This process of encoding sequence data is called "Memory code". Finally, the decoder M_D obtains the prediction $\widetilde{X}_i$ for the next session by decoding H_i. The discriminators D_{GAN} and D_{SWAP} are used to guide the generation of prediction results.

achieve the best performance in all evaluation metrics of two datasets.

- We design a MSL module and a SWAP module to encourage the model to better focus on detailed information, which helps to generate high-quality MRI data.
- Combining the original and generated images in the Whole MRI dataset, the balanced accuracy of AD vs. CN vs. MCI and sMCI vs. pMCI classification tasks can be increased by 3.61% and 4.00%, respectively.

II. Related Work

A. Longitudinal MRI Analysis

The longitudinal dataset can be considered as a type of time sequence data because it has data at multiple time points. Previous longitudinal studies [15]–[17] focused on analyzing the speed of biomarkers (e.g., hippocampus, enterorhinal cortex) over time.

With the progress in natural language processing (NLP), researchers proposed a series of models in exploring sequence data. Recurrent Neural Network (RNN) [18] was proposed to process variable-length sequence data by recursing in the evolution direction of the sequence. Long short-term memory (LSTM) [19] effectively alleviates the long-term memory decline problem of RNN by introducing a gating mechanism, but the structure is complicated. GRU [20] simplifies LSTM and speeds up training without affecting the effect.

Inspired by these developments, some works use the above sequence data processing model to analyze the MRI longitudinal dataset. Cui *et al.* present a classification framework based on combination of Multi-Layer Perceptron (MLP) neural network and RNN for longitudinal analysis of MRI images for AD diagnosis [21]. Wang *et al.* introduced LSTM to predict the AD progression for a patient's next medical visit through longitudinal data [22]. Ghazi *et al.* proposes a generalized algorithm with LSTM applied to model the progression of AD using six volumetric MRI biomarkers [23]. Jung *et al.* devise a novel computational framework that can predict the phenotypic measurement of MRI biomarkers and cognitive scores at multiple future time points [24]. The above research is dedicated to the prediction and completion of biomarkers and clinical scores. LDGAN [12] attempts to use multiple generators to complete missing 3D MRI data in the longitudinal dataset. The number of generators and training complexity of LDGAN increase linearly with the increase of the sequence data length. Unlike LDGAN, our work employs a single generator to model the laws between the data of different collect times and use the participants' existing data to predict their future time data.

B. MRI Synthesis

GANs have been successfully applied to various image synthesis tasks. Specifically, it can be used for image generation [25], text to image synthesis [26], image to image translation [11], to name a few. GAN in image synthesis, according to whether images are generated from random noise or any conditional information, can be divided into two prominent approaches: unconditional GAN and conditional GAN. Both of them have shown promising results in the field of medical imaging [27].

Unconditional GANs can generate completely new images by learning the data distribution itself. Han *et al.* [6] and Bermudez *et al.* [9] used unconditional GANs to generate

realistic 2D tumor and normal MRI slices from random variables, respectively. Kwon *et al.* [8] generated 3D brain MRI from random vectors. All these works were able to capture the real data distribution and generate diverse samples. However, such generation lack synthesis control [28], since the generation relies on no prior knowledge, different models are required for generating data with different labels.

On the other hand, conditional GANs have been widely used in cross-modality synthesis. Several studies have tried to synthesize Positron Emission Tomography (PET) images from MRI [29], [30]. As for MRI generation, Rusak *et al.* [31] synthesized 3D brain MRI images with more accurate tissue borders from Partial Volume maps. However, they failed to take advantage of the connection between MRI data acquired at different times and instead treated each MRI independently.

C. Detailed Structure Learning

The generation of 3D MRI images with clear and detailed information helps analyze the changes in the brain areas of AD patients over time. Since MRI images are grayscale images and contain less semantic information, different MRI images have a high visual similarity. The detailed structure information of 3D MRI images is the basis for generating high-quality MRI data. In the field of computer vision, the task of fine-grained classification is to distinguish subordinate classes of the same superclass, for example, to distinguish different wild birds. Compared with ordinary images, fine-grained images have more similar appearance and features, making this task more difficult. The related work of this task has certain enlightenment for the extraction of detailed structure information of 3D MRI. Several studies have tried to locate important region details by finding informative regions and then make predictions according to the feature from them [32], [33]. Chen *et al.* proposed a Destruction and Construction Learning method to force the network to learn from discriminative regions and features by carefully destruct and then reconstruct the input [34]. Inspired by the work in this field, we try to make GAN pay attention to the detailed structure information of 3D MRI data.

III. Methods

This section illustrates the proposed TR-GAN method, as shown in Fig. 3. TR-GAN consists of a SWAP module, a generator, and two discriminators. The SWAP module helps TR-GAN focus on the detailed local structure by partitioning and shuffling the input data. The MSL module contributes to stronger discriminators by sending them multi-scale and multi-location information. With the recurrent connection, the generator can process input MRI sequences of any length and predict future data based on previous inputs. As for the two discriminators, one learns to distinguish the synthesized data given by the generator from the real data, the other tries to differentiate the data generated from normal input or shuffled input. The generator competes with the discriminators until it is strong enough to fool them.

The goal of TR-GAN is to predict the patient's future MRI data based on the existing available data. Formally, the MRI data acquired in session i is denoted as $X_i \in R^{C\times D\times H\times W}$ (where C, D, H and W are the channel, depth, height, and width, respectively).

During training, TR-GAN receives multi-session input samples $\{X_0, X_1, \cdots, X_{n-1}\}$ and the corresponding labels $\{X_1, X_2, \cdots, X_n\}$ (where n denotes the number of available sessions in the training set). The generator G_W is based on encoder-decoder architecture. The encoder and decoder network are denoted as M_E and M_D. Given an input sample X_{i-1}, the encoder has a probability of p to receive the swapped input X^s_{i-1} generated by the SWAP module and a probability of $1-p$ to get the original sample X_{i-1}. This process is denoted as $\tau(X_{i-1}, p)$. M_E takes $\tau(X_{i-1}, p)$ as input and encodes it into the hidden state H_i. H_i is then sent to M_D to generate the prediction of the next session $\widetilde{X}_i$. We set $H_0 = 0$. The train phase can be indicated as:

$$H_i = M_E\left(\tau(X_{i-1}, p), H_{i-1}\right) \quad i = 1, 2, \cdots, n \tag{1}$$

$$\widetilde{X}_i = M_D\left(H_i\right) \quad i = 1, 2, \cdots, n \tag{2}$$

In the inference phase, we generate prediction for future sessions $\{X_{n+1}, \cdots, X_{n+m}\}$ (where m denotes the number of sessions needs to be predicted). M_E receives the input data and its corresponding hidden state. The output of M_E is sent to M_D. The inference phase can be indicated as:

$$H_i = \begin{cases} M_E\left(X_{i-1}, H_{i-1}\right); & i = 1, 2, \ldots, n+1 \\ M_E\left(\widetilde{X}_{i-1}, H_{i-1}\right); & i = n+2, \cdots, n+m \end{cases} \tag{3}$$

$$\widetilde{X}_i = M_D\left(H_i\right) \quad i = n+1, n+2, \cdots, n+m \tag{4}$$

A. SWAP Module

The proposed SWAP module "swaps" the local regions in the input, forcing the generator to learn useful local information and generate complete MRI according to these features. Inspired by [34], the SWAP module first uniformly partition the MRI data into sub-regions $N_s \times N_s \times N_s$ denoted by $R_{l,m,n}$, where l, m, n are the depth, height, width coordinate indices respectively and $1 \le l, m, n \le N_s$. SWAP module shuffles partitioned local regions in their 3D neighbourhood. For the m^{th} column of $R_{l,m,n}$, a random vector s_m of size N_s is generated, where the l^{th} element $s_{m,l} = l + r$ and $r \sim U(-K, K)$ is a random variable following a uniform distribution in the range of $[-K, K]$. Here, K is a tunable parameter ($1 \le K < N_s$) defining the neighbourhood range. Then we can get a new permutation σ^{col}_m of regions in m^{th} column by sorting the array s_m, verifying the condition: $\forall m \in \{1, \ldots, N_s\}$, $\left|\sigma^{col}_m(l) - l\right| < 2K$. We use similar permutation in the other two dimensions.

While competing with the discriminator, which distinguishes whether the current MRI is generated based on swapped MRI input, the generator can gradually learn to focus on informative local regions from the swapped input.

B. Generator

The generator G_W is designed based on encoder-decoder architecture. Both encoder M_E and decoder M_D are constructed based on U-net [35]. Each U-net consists of

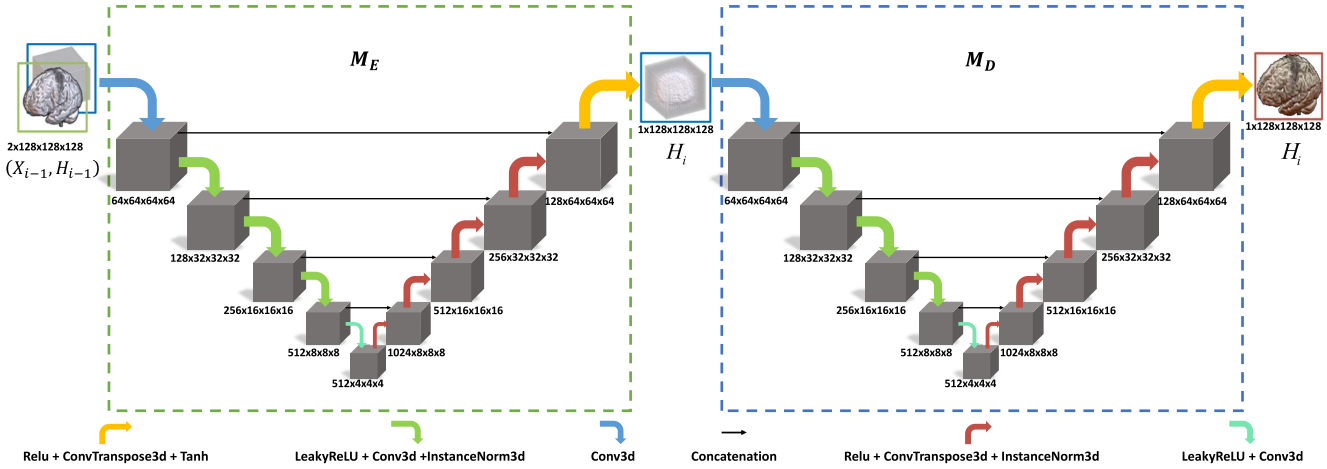


Fig. 4. The architecture of generator G_W. It contains an encoder M_E and a decoder M_D based on the U-Net structure. M_E encodes the current input X_{i-1} and the previous hidden state H_{i-1} into the current hidden state H_i. M_D decodes H_i to generate next session's prediction $\widetilde{X}_i$.

four down-sampling layers followed by four corresponding up-sampling layers, maintaining the same size of input and output data, as shown in Fig. 4.

Because of the inconsistency of the number of existing MRI data for each patient, the generator needs to deal with variant input lengths when predicting the future MRI data. Therefore, we adopt a recurrent connection which is inspired by the architecture of Recurrent Neural Network (RNN) [18], as shown in Fig. 3. Specifically, as shown in the dashed box, $\forall n \in N$, for the input sequence data $\{X_0, X_1, \cdots, X_{n-1}\}$, the information of the varying length data can be encoded into the hidden state H through n times feed forward. In the i-th feed forward, the encoder M_E randomly ($p = 50\%$ probability) takes original X_{i-1} or swapped data X^s_{i-1} and the last hidden state H_{i-1} as input to obtain H_i, which encodes the information of $\{X_0, X_1, \cdots, X_{i-1}\}$. This process of encoding sequence data is called "Memory code". Finally, the decoder M_D obtains the prediction $\widetilde{X}_i$ for the next session by decoding H_i.

C. MSL Module

The MSL module receives the predicted session or the real data, denoted as $\widetilde{X}_i$, $X_i \in R^{C\times D\times H\times W}$. As shown in Fig. 5, it randomly crops the input into N_p patches with different shapes, indicated by $X^r_i \in R^{C\cdot N_p\times\delta_1\times\delta_2\times\delta_3}$ (where δ_1, δ_2, δ_3 represents the random value of the depth, height and width, respectively). These random sized patches are then resized into the same fixed size denoted as $X^{msl}_i \in R^{C\cdot N_p\times\theta\times\theta\times\theta}$. These multi-scale and multi-location patches are then sent to discriminators, leading to stronger discriminators.

D. Discriminator

We propose a multiple scale & location pixel discriminator by combining the MSL module with pixel discriminator [11]. As shown in Fig. 5, the original 2D pixel discriminator is converted into 3D pixel discriminator to deal with the 3D MRI data. The two discriminators have different roles, and both

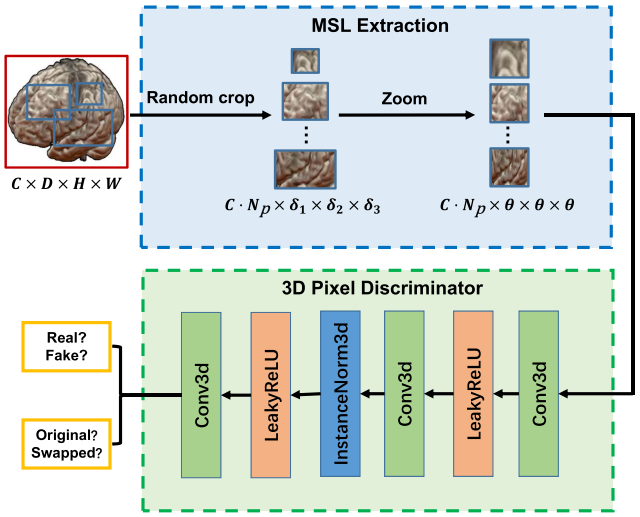


Fig. 5. The architecture of discriminator. Both D_{GAN} and D_{SWAP} use the network structure in this figure. The MSL module randomly crops the input into N_p patches and then resize them to a fixed size. The D_{GAN} learns to classify real *vs.* synthetic input data while the D_{SWAP} learns to classify original *vs.* swapped input data.

of them receive the output of the MSL module. The GAN discriminator D_{GAN} learns to distinguish whether the data is synthetic or real; the SWAP discriminator D_{SWAP} learns to classify whether the data is generated from normal MRI data or the SWAP module.

E. Loss Function

1) Generator Loss: The generator loss $\mathcal{L}_G$ consists of the generation loss $\mathcal{L}_g$, the SWAP loss $\mathcal{L}^g_s$, and the reconstruction loss $\mathcal{L}_r$. Loss term for each $\widetilde{X}_i$ is multiplied with session weight λ_i to control their relative contribution.

$$\mathcal{L}_G = \mathcal{L}_g + \mathcal{L}^g_s + \mathcal{L}_r \tag{5}$$

$\mathcal{L}_g$ is based on binary cross entropy (BCE) loss, measuring the ability for G_W to fool D_{GAN}.

$$\mathcal{L}_g = \sum_{i=1}^{n} \lambda_i \xi_{\text{BCE}}(D_{\text{GAN}}(\kappa(\widetilde{X}_i)), \psi(1)) \tag{6}$$

where $\kappa(\cdot)$ represents the operation of MSL module, $D_{\text{GAN}}(\kappa(\widetilde{X}_i))$ is the output of D_{GAN}, $\psi(a)$ is the respective ground-truth with a tensor of integer a that shares the same size with the output of D_{GAN}, ξ_{BCE} is BCE with logits loss.

$\mathcal{L}_s^g$ measures how well the generator is able to find out useful local information from the swapped MRI input and generate realistic MRI accordingly.

$$\mathcal{L}_s^g = \sum_{i=1}^{n} \lambda_i \xi_{\text{BCE}}(D_{\text{SWAP}}(\kappa(\widetilde{X}_i)), \psi(1 - y_{\text{s}})) \tag{7}$$

where y_{s} is the swap domain label, if the input swapped, $y_{\text{s}} = 1$, otherwise, $y_{\text{s}} = 0$.

L_r adopt smooth L1 loss to evaluate the difference between the output of generator $\widetilde{X}_i$ and the corresponding real MRI data X_i:

$$L_r = \sum_{i=1}^{n} \lambda_i \xi_{\text{sl}}(\widetilde{X}_i, X_i) \tag{8}$$

where ξ_{sl} is smooth L1 loss with hyper-parameter β, *i.e.*:

$$\xi_{\text{sl}} = \begin{cases} \dfrac{|\widetilde{X}_i - X_i)|^2}{2\beta}, & \text{if } |\widetilde{X}_i - X_i| < \beta \\ |\widetilde{X}_i - X_i| - \dfrac{\beta}{2}, & \text{else} \end{cases} \tag{9}$$

2) Discriminator Loss: The discriminator loss $\mathcal{L}_D$ is composed of the discrimination loss $\mathcal{L}_d$ and the SWAP loss $\mathcal{L}_s^d$.

$$\mathcal{L}_D = \mathcal{L}_d + \mathcal{L}_s^d \tag{10}$$

$\mathcal{L}_d$ estimates the ability of D_{GAN} to distinguish between the real and forged data, and $\mathcal{L}_s^d$ evaluates how well D_{SWAP} can discriminate whether the data is generated from swapped MRI input.

$$\begin{aligned} \mathcal{L}_d = \sum_{i=1}^{n} \frac{1}{2} \lambda_i (&\xi_{\text{BCE}}(D_{\text{GAN}}(\kappa(X_i)), \psi(1)) \\ &+ \xi_{\text{BCE}}(D_{\text{GAN}}(\kappa(\widetilde{X}_i)), \psi(0))) \end{aligned} \tag{11}$$

$$\mathcal{L}_s^d = \sum_{i=1}^{n} \lambda_i \xi_{\text{BCE}}(D_{\text{SWAP}}(\kappa(\widetilde{X}_i)), \psi(y_{\text{s}})) \tag{12}$$

IV. Experiments

A. Dataset

Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial MRI, PET, other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early AD.

ADNI is a longitudinal multi-site observational study of CN, MCI, and AD. The subjects have collected data in multiple periods (usually once every half year). "Mj" represents the data that was collected in the j-th month after the first collection. For example, We use "M00", "M06", "M12", etc. to represent the data collected after the subjects for the first time, after 6, 12 months, etc.

MRI longitudinal time sequence data in ADNI have 1453 subjects. The total record is randomly shuffled split into training (1153) and testing (300) set at subject-level. The testing set contains data for AD, CN, and MCI, each group with 100 subjects. As shown in Fig. 2, the amount of available data gradually decreases over time. More than 88% of subjects have M06 data, but only 28% have M36 data. Therefore, the dataset size of different prediction tasks might be inconsistent.

According to the different preprocessing methods, two datasets are constructed, the grey matter (GM) tissue map dataset and the Whole MRI dataset.

1) GM Tissue Map Dataset: Original dataset passes through the t1-volume pipeline of Clinica [37], [38] to process 3D MRI directly. The Unified Segmentation procedure [39] is used to simultaneously perform tissue segmentation, bias correction, and spatial normalization. After that, t1-weighted volumetric images are segmented into grey matter (GM), white matter (WM), and cerebrospinal fluid (CSF). The detailed preprocessing process and toolbox help can be obtained on this web page.[1] Only a GM tissue map is used in the following experiments as it is more related to the AD diagnosis. Finally, all 3D GM tissue maps are resized to $128 \times 128 \times 128$.

2) Whole MRI Dataset: Whole MRI dataset have been obtained using the t1-linear pipeline of Clinica [5], [37] to process 3D MRI directly. More precisely, bias field correction was applied using the N4ITK method [40]. Next, an affine registration was performed using the SyN algorithm [41] from ANTs [42] to align each image to the MNI space with the ICBM 2009c nonlinear symmetric template [43], [44]. The detailed preprocessing process and toolbox help can be obtained on this web page.[2] Finally, all 3D MRI images are resized to $128 \times 128 \times 128$.

B. Implementation Details

The models are implemented using Python 3.6.12, PyTorch on a workstation with Nvidia Tesla V100-SXM2-32GB-LS. We select the model with the minimum MSE loss during 100 epochs training for evaluation. Adam optimizer is used for both generator and discriminator. The learning rate for M12 and M18 prediction are 9e-4 and 7e-4, respectively. And the hyper-parameter λ for M12 and M18 prediction are $[0.8, 1.0]$ and $[0.3, 0.5, 1.0]$, respectively. For the SWAP module, it is implemented only in the first 50 epochs. The value of division number N_s is set to 8 and the neighborhood range K is 2. In the MSL module, the number of randomly cropped patches

[1] https://aramislab.paris.inria.fr/clinica/docs/public/latest/Pipelines/T1_Linear/

[2] https://aramislab.paris.inria.fr/clinica/docs/public/latest/Pipelines/T1_Volume/

TABLE I
SINGLE SESSION PREDICTION RESULTS FOR TWO DATASET

Methods	Dataset	M-domain	M-input	[M00, M06]→M12 (train:703 sub, test:223 sub)			[M00, M06, M12]→M18 (train:117 sub, test:17 sub)		
				MMSE ($\times 10^{-2}$) ↓	MS-SSIM ↑	PSNR ↑	MMSE ($\times 10^{-2}$) ↓	MS-SSIM ↑	PSNR ↑
AdaIN-GAN [36]	GM tissue map			9.371 ± 0.616	0.9336 ± 0.01385	10.291 ± 0.284	9.380 ± 0.589	0.9323 ± 0.00396	10.287 ± 0.278
StarGAN [13]		✓		2.980 ± 1.773	0.9817 ± 0.02312	15.605 ± 1.445	7.250 ± 0.434	0.9533 ± 0.01320	11.404 ± 0.258
CollaGAN [14]		✓	✓	2.180 ± 1.752	0.9263 ± 0.02442	17.183 ± 1.793	4.208 ± 1.020	0.9409 ± 0.21110	13.850 ± 0.818
LDGAN [12]		✓*		2.582 ± 2.748	0.9749 ± 0.05513	16.729 ± 2.162	1.973 ± 1.327	0.9824 ± 0.01231	17.508 ± 1.655
Pix2Pix [11]				1.916 ± 1.780	0.9867 ± 0.02047	17.929 ± 2.106	1.609 ± 0.426	0.9912 ± 0.00200	18.074 ± 1.075
Pix2Pix(M) [11]			✓	1.724 ± 2.034	0.9861 ± 0.03329	18.699 ± 2.413	1.340 ± 0.943	0.9892 ± 0.01732	19.334 ± 2.012
Cycle-GAN [10]				1.667 ± 1.903	0.9883 ± 0.02125	18.866 ± 2.536	1.423 ± 0.437	0.9915 ± 0.00396	18.664 ± 1.300
TR-GAN(ours)		✓	✓	$\mathbf{1.314 \pm 1.705}$	$\mathbf{0.9898 \pm 0.01907}$	$\mathbf{20.099 \pm 2.680}$	$\mathbf{0.931 \pm 0.522}$	$\mathbf{0.9943 \pm 0.00456}$	$\mathbf{20.818 \pm 1.982}$
AdaIN-GAN [36]	Whole MRI			3.120 ± 1.189	0.8338 ± 0.05058	15.261 ± 1.236	2.615 ± 0.565	0.8464 ± 0.03865	15.919 ± 0.892
StarGAN [13]		✓		2.274 ± 0.949	0.8899 ± 0.05005	16.667 ± 1.326	3.028 ± 0.647	0.8212 ± 0.03735	15.281 ± 0.882
CollaGAN [14]		✓	✓	1.494 ± 0.769	0.9209 ± 0.03906	18.646 ± 1.747	2.027 ± 0.696	0.8779 ± 0.04490	17.104 ± 1.119
LDGAN [12]		✓*		2.489 ± 1.235	0.8356 ± 0.07547	16.321 ± 1.404	2.648 ± 1.035	0.8449 ± 0.05023	16.032 ± 1.430
Pix2Pix [11]				1.497 ± 1.019	0.9302 ± 0.04657	18.728 ± 1.835	1.328 ± 0.427	0.9350 ± 0.02165	18.937 ± 1.151
Pix2Pix(M) [11]			✓	5.443 ± 1.096	0.6698 ± 0.06041	12.725 ± 0.846	5.668 ± 0.659	0.6827 ± 0.03427	12.495 ± 0.501
Cycle-GAN [10]				1.529 ± 1.101	0.9287 ± 0.04917	18.746 ± 2.089	1.280 ± 0.394	0.9324 ± 0.02071	19.122 ± 1.295
TR-GAN (ours)		✓	✓	$\mathbf{1.201 \pm 0.696}$	$\mathbf{0.9404 \pm 0.04456}$	$\mathbf{19.707 \pm 1.972}$	$\mathbf{1.167 \pm 0.440}$	$\mathbf{0.9359 \pm 0.03288}$	$\mathbf{19.586 \pm 1.432}$

Note: "Mj" represents the data that was collected in the j-th month after the first collection. "sub" denotes subjects. "M-domain" means that the model performs multi-domain generation tasks, and "M-input" denotes that the generator has input data for more than one session.
* LDGAN uses multiple generators to achieve multi-domain generation.

N_p is 128 and parameters controlling the output size, θ, is set to 32, β is set to 0.06.

Three image metrics: MSE, multi-scale structural similarity metric (MS-SSIM) [45], and peak signal to noise ratio (PSNR) are adopted to evaluate the authenticity of the generated MRI results. MSE is used to evaluate the reconstruction error between the predicted image and the real image. MS-SSIM is a commonly used image evaluation metric, which considers the similarity between the generated image and the real image in terms of brightness, contrast, structure and resolution. PSNR is the most common and widely used objective measurement method for evaluating image quality.

C. Results

To the best of our knowledge, TR-GAN is the first work to perform a many-to-many 3D MRI generation task by using a single generator model. It predicts the participants' subsequent multiple session data based on the existing multi-session data.

For comparison, we adopt three popular conditional GAN frameworks, AdaIN-GAN, Pix2Pix and Cycle-GAN, to perform the one-to-one predict task (M00→M∗). To study whether the increase of input data can improve the quality of the generated image, we changed the input of Pix2Pix from a single session to the concatenate of multi-session data, denoted as Pix2Pix(M). This direct concatenating multi-session input method is also used to compare with the recurrent connection method we used. In addition, we also adopt two popular many-to-many models (Star-GAN and CollaGAN) and an LDGAN model that try to use multiple generators to complement the longitudinal dataset for comparison.

1) Single Session Prediction: In this experiment, we evaluate TR-GAN's performance on single future session prediction by M12 and M18 generation tasks in two datasets. The loss of training history are shown in Fig. 6 and Fig. 7. Quantitative comparison results are illustrated in Table I, we have annotated the methods in the table according to whether to perform multi-domain generation or accept more than one session of data. TR-GAN achieved the best performance on all evaluation metrics on the two datasets.

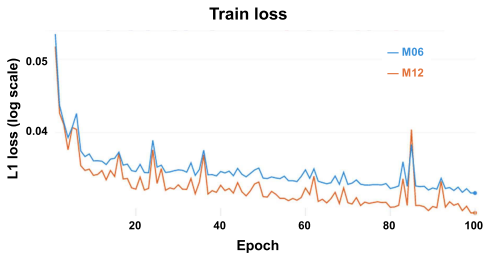


Fig. 6. L1 loss training history of [M00, M06]→M12 task.

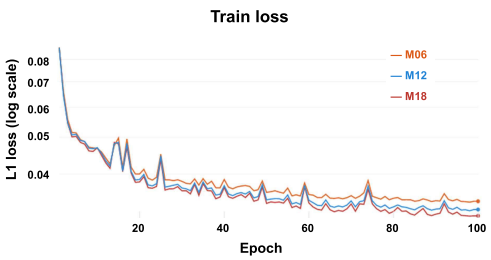


Fig. 7. L1 loss training history of [M00, M06, M12]→M18 task.

On the GM tissue map dataset, the performance of the Multi-domain model (performs multi-domain generation) is not as good as the single-domain model (performs single-domain generation tasks) such as Pix2Pix. This may be because these multi-domain models need to be trained for different generation domains during the training phase, which leads to inferior performance in specific generation tasks. By comparing the predictions of M12 and M18, it can be found that the Multi-domain model performs worse as the number of generated domains increases. LDGAN is not affected because each domain generation has its own generator, and the effect is improved due to the increase of supervision information. It is worth noting that TR-GAN uses a single generator to perform multi-domain generation tasks but still achieves the best results, which shows that the recurrent connection method we adopted is more suitable for multi-domain generation tasks.

TABLE II
UNTRAINED MULTI-SESSION PREDICTION RESULTS

Input (test:9 sub)	Predict	GM tissue map			Whole MRI		
		MMSE ($\times 10^{-2}$) ↓	MS-SSIM ↑	PSNR ↑	MMSE ($\times 10^{-2}$) ↓	MS-SSIM ↑	PSNR ↑
[M00, M06]	M12	0.776 ± 0.250	0.9956 ± 0.00138	21.307 ± 1.297	1.006 ± 0.471	0.9473 ± 0.02848	20.377 ± 1.795
	M18*	0.890 ± 0.297	0.9953 ± 0.00127	20.737 ± 1.412	1.000 ± 0.208	0.9483 ± 0.00980	20.088 ± 0.849
	M24*	2.507 ± 3.832	0.9829 ± 0.02993	18.439 ± 3.718	1.749 ± 1.134	0.9023 ± 0.07596	18.250 ± 2.279
[M00, M06, M12]	M18	0.746 ± 0.245	0.9962 ± 0.00110	21.497 ± 1.382	1.035 ± 0.382	0.9478 ± 0.01705	20.061 ± 1.241
	M24*	2.335 ± 3.732	0.9836 ± 0.02999	18.950 ± 3.838	1.768 ± 1.132	0.9043 ± 0.07317	18.228 ± 2.337

* TR-GAN did not use the data of these sessions in the training phase. Therefore, these session prediction tasks can test the prediction ability of TR-GAN on untrained sessions.

In addition, we also found that CollaGAN performs better than StarGAN, while Pix2Pix(M) is better than Pix2Pix. This shows that inputting multiple sessions data can effectively improve performance. The AdaIN-GAN method achieved the worst results. The AdaIN-GAN method achieved the worst results, which may be because the adaptive instance normalization method is more suitable for RGB images of natural scenes, which is quite different from the grayscale images of MRI.

On the Whole MRI dataset, CollaGAN, which receives multi-session data input, performs better than StarGAN. It is worth noting that Pix2Pix(M) achieved the worst performance, which shows that simply connecting multiple input session data in the channel dimension to the model has poor generalization. TR-GAN also achieved the best performance. When predicting M12 and M18, MSE was 19.6% and 8.8% lower than the second-placed CollaGAN and Cycle-GAN, respectively.

Additionally, Fig. 8 presents the comparison of error maps on GM tissue map and whole MRI datasets, showing that data generated by TR-GAN is more realistic than others. Fig. 9 presents the center-cut slices of real and generated samples. To better distinguish the difference between real and generated data, we zoomed in on the local detail, indicated by the DETAIL part. The result shows that TR-GAN can better capture the fine structure than other methods. Meanwhile, the histogram (Fig. 10) provides a clear indication of the data distribution for each sample. It can be observed that TR-GAN shares the most similar distribution to the real data.

2) Untrained Multi-Session Prediction: We use [M00, M06, M12] and [M00, M06, M12, M18] to train two models to predict forward till to M24. M18 and M24 predicted by the first model and M24 predicted by the second model are untrained session data. TR-GAN predicts untrained future session data based on existing data through multiple feed-forwards. As shown in Table II, for untrained session data, the predicted performance deteriorates over time. We use TR-GAN's worst-performing prediction result on M24 to make a simple comparison with other methods in Table I that use the same training data, and find that the result is acceptable. Specifically, on the GM tissue map dataset, TR-GAN trained with [M00, M06, M12] performed better than AdaIN-GAN, StarGAN, and LDGAN, and TR-GAN trained with [M00, M06, M12, M18] performed better than AdaIN-GAN, StarGAN, and CollaGAN. On the Whole MRI dataset, TR-GAN trained with [M00, M06, M12] outperforms AdaIN-GAN, StarGAN, LDGAN, and Pix2Pix(M), and TR-GAN trained with [M00, M06, M12, M18] GAN outperforms AdaIN-GAN, StarGAN, CollaGAN, LDGAN, and Pix2Pix(M). There is no doubt that TR-GAN has accumulated errors in its predictions on untrained future session data. In order to find a reasonable number of predictions, we use AD diagnostic experiments in the section (IV-C.3) to study the validity of the predicted data.

TABLE III
THE NUMBER OF REAL AND FAKE 3D IMAGES IN TRAINING SET FOR VARYING SETTINGS

Task	Setting	TR-GAN-12	TR-GAN-6	TR-GAN-3	Others*
AD vs. CN vs. MCI	All Real	3397, 0	3397, 0	3397, 0	3397, 0
	All Real + Fake	3397, 6233	3397, 2635	3397, 850	3397, 73
	Part Real	73, 0	73, 0	73, 0	73, 0
	Only Fake	0, 6233	0, 2635	0, 850	0, 73
	Part Real + Fake	73, 6233	73, 2635	73, 850	73, 73
sMCI vs. pMCI	All Real	1083, 0	1083, 0	1083, 0	1083, 0
	All Real + Fake	1083, 2759	1083, 1137	1083, 344	1083, 34
	Part Real	34, 0	34, 0	34, 0	34, 0
	Only Fake	0, 2759	0, 1137	0, 344	0, 34
	Part Real + Fake	34, 2759	34, 1137	34, 344	34, 34

Note: We use "," to separate the number of real and fake 3D MRI data in the table.
* Others represent TR-GAN-1, AdaIN-GAN, StarGAN, CollaGAN, LDGAN, Pix2Pix, Pix2Pix(M), and Cycle-GAN.

3) AD Diagnosis Classification: To demonstrate TR-GAN is able to generate realistic data with reasonable diagnosis labels, we test the performance of AD vs. CN vs. MCI and stable MCI (sMCI: MCI patients who did not progress to AD in 36 months) vs. progressive MCI (pMCI: MCI patients who progress to AD in 36 months) classification. We divide the dataset into training, validation (each category contains about 30 images), and testing (each category contains 100 images) sets. For the training set, use [M00, M06] as input to infer M12, M18, etc. The fake data of [M12, …, M$(n+1)*6$] is obtained by infer n times, denoted as TR-GAN-n. Fake M12 data generated by other methods are used to study the impact of fake data generated by different GAN methods on MRI classification tasks. As shown in Table III, we designed various combinations of real and fake 3D MRI data to study the classification performance of fake data. The 3D-ResNet101 is adopted as the backbone for classification and the balanced accuracy [46] as the evaluation metric. For each setting, we train for 50 epochs. The model with the best validation balanced accuracy is used to evaluate the testing set, and the test balanced accuracy is reported in Fig. 11.

Overall, under two classification tasks and two datasets, the balanced accuracy of TR-GAN-1 and other methods in "Only Fake" is slightly lower (or approximately) than that of "Part Real" using the same amount of real data.

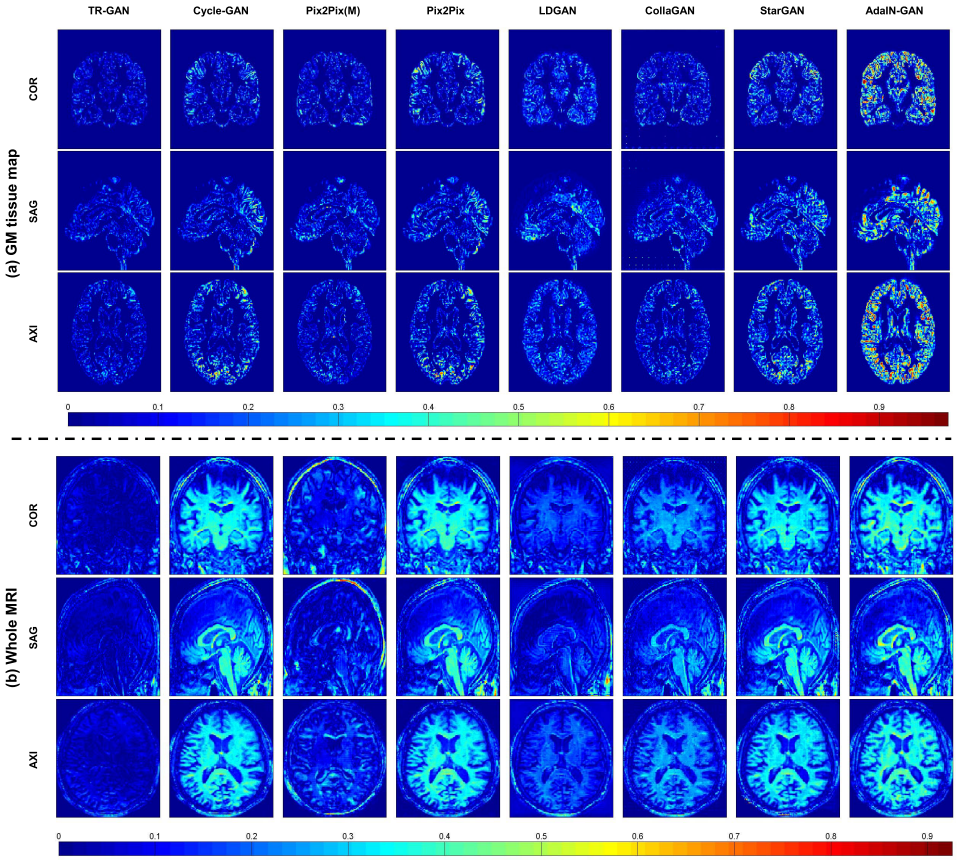


Fig. 8. Comparison of error maps along sagittal (SAG), coronal (COR) and axial (AXI) planes on GM tissue map and whole MRI datasets. For visualization, we use $|\widetilde{X}_i - X_i|/2$ to construct the error map. The pixel value in the figure represents the error between the real image and the fake image generated by different GAN models. Therefore, the closer the pixel color is to blue (0 in the colormap), the smaller the error. Through comparison, we can find that TR-GAN's errors are smaller than those of other methods in the two datasets.

The accuracy of some methods approximates random guessing. The comparison of "Part Real" and "Only Fake" shows that the classification performance of the generated fake data is weaker than the real data. In the "Only Fake" configuration, the balanced accuracy of TR-GAN-3 and TR-GAN-6 gradually increased, while the balanced accuracy of TR-GAN-12 decreased. This shows that the increase in fake data generated by TR-GAN within a specific range can improve classification balanced accuracy. The more forward inferences, the greater the accumulation of errors and the lower the quality of the fake data obtained. Therefore, the classification performance of TR-GAN-12 has declined.

In the "All Real+Fake" configuration, TR-GAN-1 has achieved performance that exceeds (or approximates) other comparison methods. As the data generated by TR-GAN increases, the classification balanced accuracy first increases until TR-GAN-6 reaches the highest point, and there is a decline in TR-GAN-12. TR-GAN-1 improves the corresponding classification balanced accuracy in both classification tasks of the two datasets. TR-GAN-6 is 2.56%, 2.50%, 3.61%, and 4.00% higher than using only real data under the four tasks in Figure 11, respectively. These results show that under a reasonable number of predictions, the data predicted by TR-GAN can effectively improve the accuracy of the diagnosis algorithm. In the "Part Real+Fake" configuration, the performance of each model is higher (or approximately) than its performance in the "Only Fake" configuration.

TABLE IV
ABLATION STUDY FOR SWAP AND MSL MODULE

Methods	MSE ($\times 10^{-2}$) ↓	MS-SSIM ↑	PSNR ↑
baseline	0.9891 ± 0.6398	0.9929 ± 0.008825	20.676 ± 2.151
SWAP	0.9634 ± 0.5535	0.9938 ± 0.005781	20.693 ± 2.014
MSL	0.9461 ± 0.5268	0.9940 ± 0.004847	20.741 ± 1.963
SWAP + MSL	$\mathbf{0.9313 \pm 0.5223}$	$\mathbf{0.9943 \pm 0.004560}$	$\mathbf{20.818 \pm 1.982}$

4) Ablation Study: We conduct ablation studies to evaluate the effectiveness of different components in our proposed TR-GAN. The baseline is referred to as the TR-GAN architecture without the SWAP and MSL module. Performance for different models is evaluated on the task of M18 prediction. According to the results in Table IV, compared with the baseline model, when the SWAP module and the MSL module are adopted, MSE drops by 2.60% and 4.35%, respectively. The best performance is achieved when this two modules are used simultaneously, decreasing the MSE by 5.84% compared with the baseline model. These performance improvements prove the potency of the two modules. The multi-scale information extracted by the MSL module can cultivate better discriminators, forcing the generator to focus on multi-scale

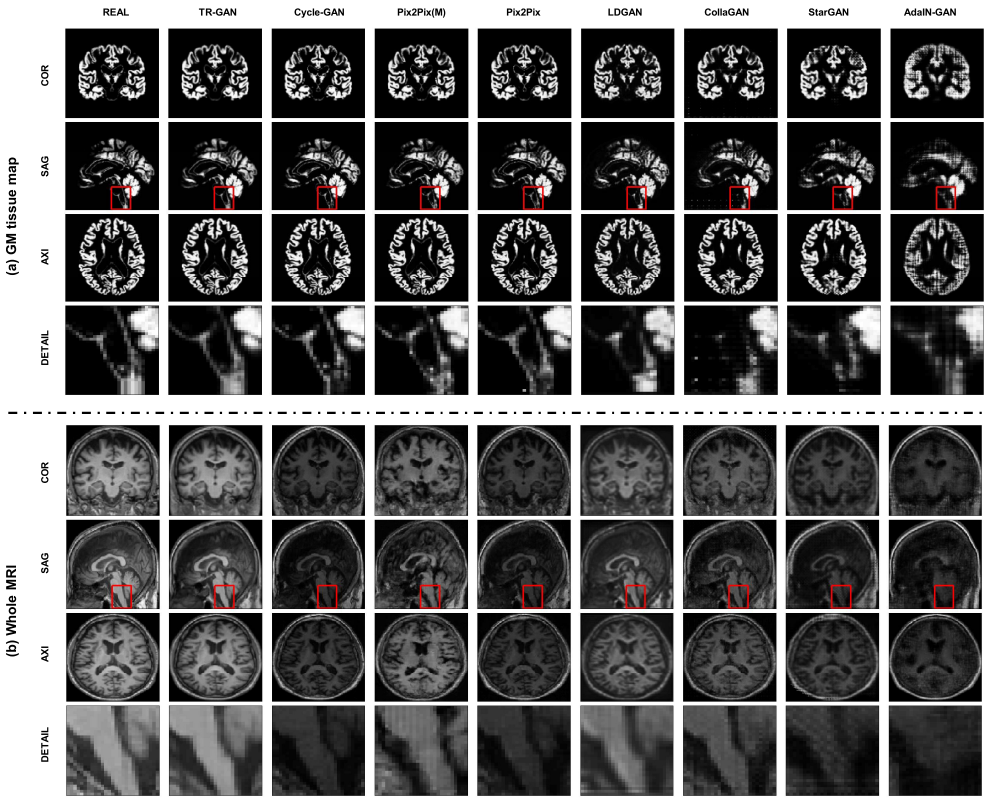


Fig. 9. Comparison of fake images generated by different methods on GM tissue map and whole MRI datasets. "REAL" represents the ground truth. To better distinguish the difference between real and generated images, we zoomed in on the local detail, indicated by the "DETAIL" part. The result shows that TR-GAN can better capture the fine structure than other methods.

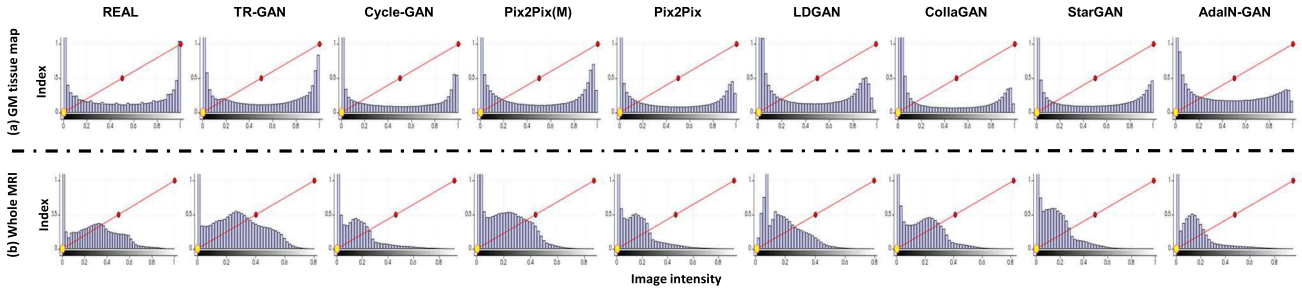


Fig. 10. Comparison of data distribution histograms in fake images generated by different methods on GM tissue map and whole MRI datasets. "REAL" represents the ground truth. It can be observed that TR-GAN shares the most similar distribution to the ground truth on the two datasets.

features. The SWAP module makes the generator stronger by helping it concentrate on detailed valuable local information.

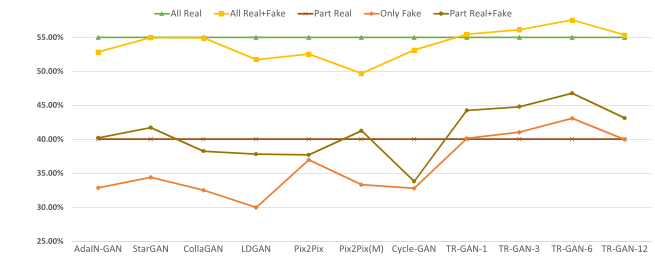
V. Discussion

A. Number of Sub-Regions in the SWAP Module

The number of partitions N_s for the SWAP module is an important parameter. Table V presents the GM tissue map synthesis quality of the TR-GAN trained with different feasible N_s. We can observe that the synthesis quality decreases while N_s goes bigger. The best performance is achieved at $N_s = 8$. In general, if we set N_s as a large number, information contained in each sub-region might be limited because of the destructed brain structure. On the other hand, it might be easier for us to keep the proper local information and take advantage of it with a smaller N_s.

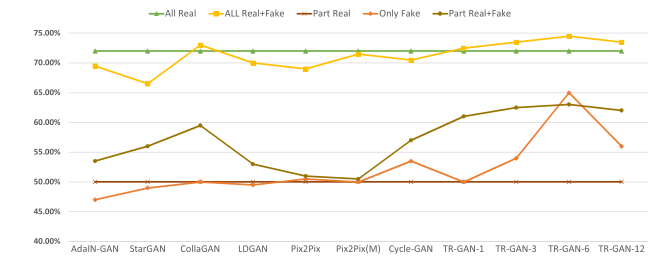
B. Number of Patches in MSL Module

Table VI presents the GM tissue map synthesis quality of the TR-GAN trained with different feasible N_{p}. It can be observed that the performance first increases and then decreases while N_{p} increases. When N_{p} is set to 128, the best performance is obtained. We guess that the advantage of our MSL module would likely be restricted with a small N_{p}, as a lot of information might be lost when only a limited number of



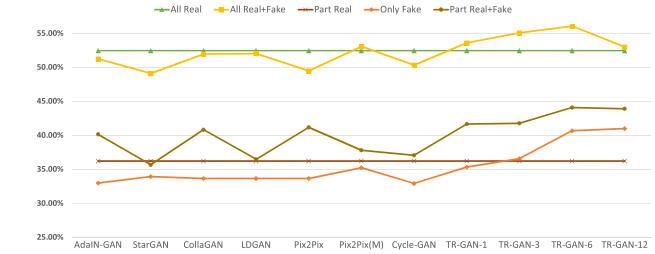
	AdaIN-GAN	StarGAN	CollaGAN	LDGAN	Pix2Pix	Pix2Pix(M)	Cycle-GAN	TR-GAN-1	TR-GAN-3	TR-GAN-6	TR-GAN-12
All Real	55.01%	55.01%	55.01%	55.01%	55.01%	55.01%	55.01%	55.01%	55.01%	55.01%	55.01%
All Real+Fake	52.86%	54.97%	54.93%	51.76%	52.54%	49.71%	53.15%	55.46%	56.12%	**57.57%**	55.36%
Part Real	40.06%	40.06%	40.06%	40.06%	40.06%	40.06%	40.06%	40.06%	40.06%	40.06%	40.06%
Only Fake	32.90%	34.43%	32.54%	30.00%	37.00%	33.36%	32.83%	40.15%	41.06%	**43.11%**	40.00%
Part Real+Fake	40.22%	41.75%	38.28%	37.85%	37.75%	41.29%	33.83%	44.25%	44.81%	**46.79%**	43.15%

(a) Balanced accuracy of AD vs. CN vs. MCI (GM tissue map).



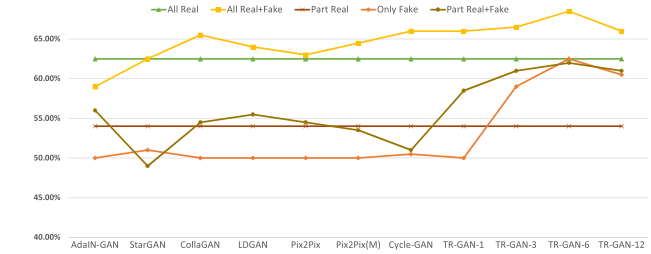
	AdaIN-GAN	StarGAN	CollaGAN	LDGAN	Pix2Pix	Pix2Pix(M)	Cycle-GAN	TR-GAN-1	TR-GAN-3	TR-GAN-6	TR-GAN-12
All Real	72.00%	72.00%	72.00%	72.00%	72.00%	72.00%	72.00%	72.00%	72.00%	72.00%	72.00%
ALL Real+Fake	69.50%	66.50%	73.00%	70.00%	69.00%	71.50%	70.50%	72.50%	73.50%	**74.50%**	73.50%
Part Real	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%
Only Fake	47.00%	49.00%	50.00%	49.50%	50.50%	50.00%	53.50%	50.00%	54.00%	**65.00%**	56.00%
Part Real+Fake	53.50%	56.00%	59.50%	53.00%	51.00%	50.50%	57.00%	61.00%	62.50%	**63.00%**	62.00%

(b) Balanced accuracy of sMCI vs. pMCI (GM tissue map).



	AdaIN-GAN	StarGAN	CollaGAN	LDGAN	Pix2Pix	Pix2Pix(M)	Cycle-GAN	TR-GAN-1	TR-GAN-3	TR-GAN-6	TR-GAN-12
All Real	52.46%	52.46%	52.46%	52.46%	52.46%	52.46%	52.46%	52.46%	52.46%	52.46%	52.46%
All Real+Fake	51.26%	49.10%	51.96%	52.04%	49.51%	53.06%	50.35%	53.61%	55.07%	**56.07%**	52.99%
Part Real	36.22%	36.22%	36.22%	36.22%	36.22%	36.22%	36.22%	36.22%	36.22%	36.22%	36.22%
Only Fake	33.00%	33.94%	33.67%	33.67%	33.67%	35.24%	32.94%	35.35%	36.58%	**40.67%**	41.00%
Part Real+Fake	40.14%	35.68%	40.82%	36.46%	41.19%	37.79%	37.08%	41.68%	41.76%	**44.10%**	43.93%

(c) Balanced accuracy of AD vs. CN vs. MCI (Whole MRI).



	AdaIN-GAN	StarGAN	CollaGAN	LDGAN	Pix2Pix	Pix2Pix(M)	Cycle-GAN	TR-GAN-1	TR-GAN-3	TR-GAN-6	TR-GAN-12
All Real	62.50%	62.50%	62.50%	62.50%	62.50%	62.50%	62.50%	62.50%	62.50%	62.50%	62.50%
All Real+Fake	59.00%	62.50%	65.50%	64.00%	63.00%	64.50%	66.00%	66.00%	66.50%	**68.50%**	66.00%
Part Real	54.00%	54.00%	54.00%	54.00%	54.00%	54.00%	54.00%	54.00%	54.00%	54.00%	54.00%
Only Fake	50.00%	51.00%	50.00%	50.00%	50.00%	50.00%	50.50%	50.00%	59.00%	**62.50%**	60.50%
Part Real+Fake	56.00%	49.00%	54.50%	55.50%	54.50%	53.50%	51.00%	58.50%	61.00%	**62.00%**	61.00%

(d) Balanced accuracy of sMCI vs. pMCI (Whole MRI).

Fig. 11. The test balanced accuracy comparison of the ADNI dataset expanded by different GAN models. We designed various combinations of real data and fake data to study the classification performance of fake data (see Table III for details). The experiment uses 3D-ResNet101 as the classification backbone and performs AD vs. CN vs. MCI and sMCI vs. pMCI classifications under the GM tissue map and Whole MRI datasets, respectively. In the "All Real+Fake" configuration, TR-GAN-6 is 2.56%, 2.50%, 3.61%, and 4.00% higher than using only real data under those four tasks, respectively.

TABLE V
THE GM TISSUE MAP SYNTHESIS QUALITY OF THE MODEL TRAINED WITH A DIFFERENT VALUE OF N_s

N_s	[M00, M06, M12]→M18 (train:117, test:17)		
	MSE ($\times 10^{-2}$) ↓	MS-SSIM ↑	PSNR ↑
8	$\mathbf{0.9313 \pm 0.5223}$	$\mathbf{0.9943 \pm 0.00456}$	$\mathbf{20.818 \pm 1.982}$
16	1.0100 ± 0.7156	0.9925 ± 0.01060	20.681 ± 2.292
32	1.6602 ± 2.0910	0.9832 ± 0.03621	19.512 ± 3.188

TABLE VI
THE GM TISSUE MAP SYNTHESIS QUALITY OF THE MODEL TRAINED WITH A DIFFERENT VALUE OF N_p

N_p	[M00, M06, M12]→M18 (train:117, test:17)		
	MSE ($\times 10^{-2}$) ↓	MS-SSIM ↑	PSNR ↑
64	0.939 ± 0.5665	0.9941 ± 0.00574	20.805 ± 2.083
128	$\mathbf{0.9313 \pm 0.5223}$	$\mathbf{0.9943 \pm 0.00456}$	$\mathbf{20.818 \pm 1.982}$
256	1.593 ± 2.8620	0.9756 ± 0.07743	20.264 ± 3.357

patches are selected. However, if the N_p is too big, redundant information might be adopted, which contributes negatively to the performance.

C. Discussion on State Encode

To better understand how the previous session data is encoded by the generator's encoder, we visualize the left hippocampus in the hidden state during the training process. As shown in Fig. 12, the horizontal and vertical axis represents different training times and different sessions, respectively. Having trained for one epoch, the hidden state can roughly show the shape of the left hippocampus, but is accompanied by grid-like noise. As the training progresses, the noise gradually decreases, and the left hippocampus becomes more explicit and is filled with more detailed brain structure information. These results indicate that the recurrent connection in TR-GAN can capture the information of previous sessions, which is constantly updated and enriched during the training.

D. Limitation

Some limitations should be noted in our work. First, the development of AD disease is affected by many factors, such as genes, lifestyle habits, academic qualifications, etc. From a clinical perspective, the future session MRI data generated by TR-GAN may not be directly used to predict the disease progression of AD patients. Fortunately, the data generated by our model can indeed be combined with real data to expand existing datasets and improve the accuracy of existing data-driven diagnostic algorithms. Second, the comparison of computing resources in the [M00, M06, M12]→M18 task is shown in Table VII. The memory occupation of TR-GAN in the training phase is 7073M, which is large than Pix2Pix and Pix2Pix(M). Cycle-GAN and LDGAN with multiple generators occupy the most memory. Compared with StarGAN and CollaGAN, which also perform multi-domain generation

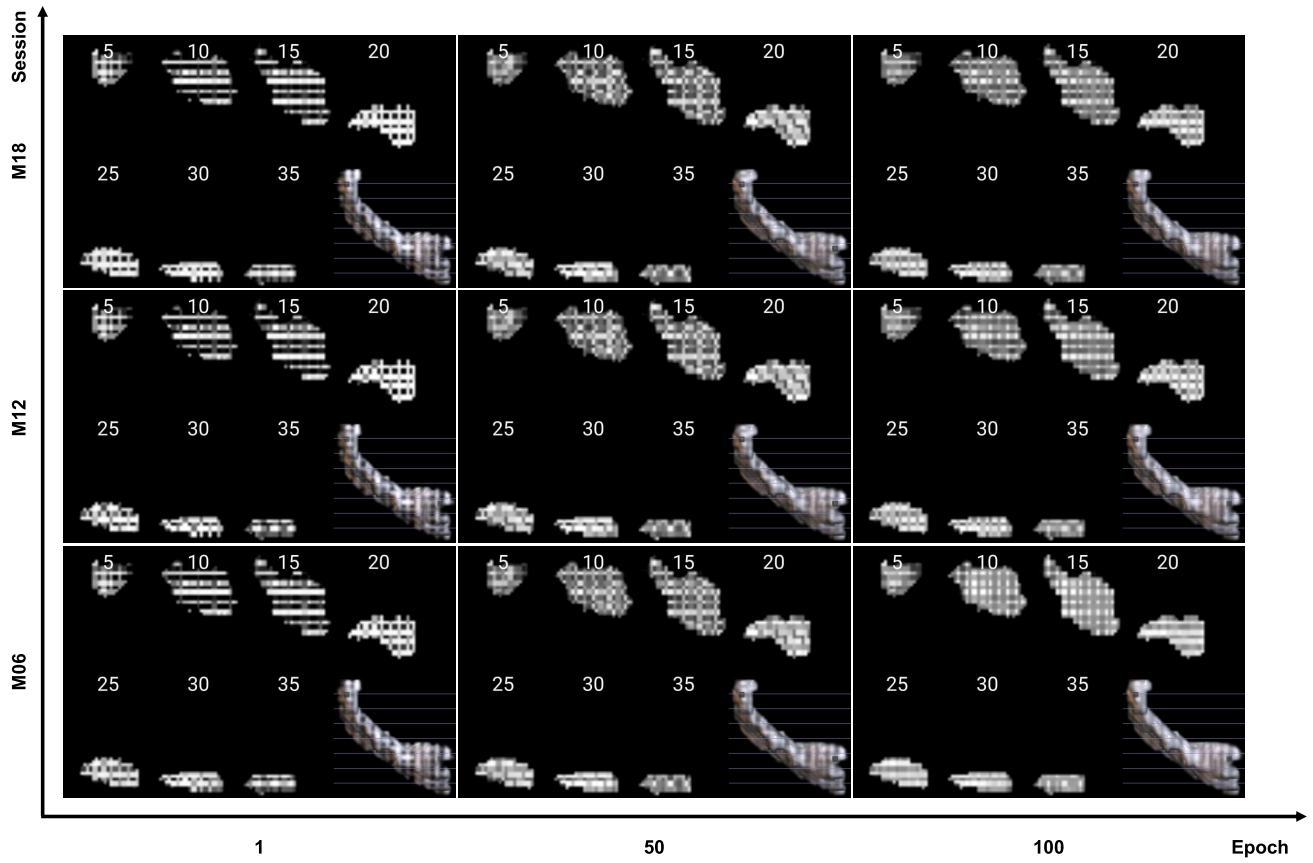


Fig. 12. Evolution of the left hippocampus in the hidden state with different training processes on Whole MRI dataset. The hippocampus on the left was extracted from the AAL template [47]. As the hidden state encodes the memory of previous sessions, the hidden state contains the general information of the MRI data and gradually becomes clear during the training.

TABLE VII
COMPARISON OF REQUIRED TRAINING TIME AND COMPUTING RESOURCES ([M00, M06, M12]→M18)

Methods	G Para.(M)	D Para.(M)	Infer time(ms)	Train time(h)	Memory(M)
AdaIN-GAN [36]	21.02	/	346.72	7.93	13269
StarGAN [13]	26.62	178.91	1189.43	59.90	21531
CollaGAN [14]	167.29	178.91	928.68	196.29	13419
LDGAN [12]	0.41 * 4	0.28 * 4	163.81	67.25	22027
Pix2Pix [11]	167.27	11.05	924.12	11.01	4849
Pix2Pix(M) [11]	167.28	11.06	925.35	11.04	4901
Cycle-GAN [10]	12.88 * 2	11.05 * 2	279.80	59.14	27699
TR-GAN (ours)	133.2	0.02 * 6	1767.1 * 3	59.86	7073

Note: G Para. and D Para. represent the parameters of the generator and discriminator, respectively.

tasks, TR-GAN's recurrent connection method saves memory. In terms of model parameters, TR-GAN's generator parameters are 133.2M, which is lower than Pix2Pix and Pix2Pix(M). TR-GAN's discriminator has the smallest amount of parameters. In general, TR-GAN has a moderate amount of memory and parameters, but it has achieved the best prediction performance. We will explore the generation of MRI data based on the combination of lower resolution MRI input and super-resolution techniques. Third, due to TR-GAN adopting a recurrent connection method, multiple forward propagations are required in the inference stage, which causes the inference time of TR-GAN to be longer than other methods. In terms of training time, the training time of the Multi-domain model (performs multi-domain generation) is longer than the single-domain model (performs single-domain generation tasks). The training time of TR-GAN is 59.86 hours, which is less than (or similar to) other models that perform multi-domain generation tasks. There is no real-time requirement for multi-session MRI prediction tasks, so it is acceptable to sacrifice inference time to obtain higher accuracy prediction results. Like other related works mentioned in the paper, TR-GAN cannot carry out mini-batch training. We are working on a better network architecture to achieve batch-level gradient updates.

VI. CONCLUSION

In this paper, we explore the challenging problem of multi-session MRI prediction. TR-GAN is proposed to learn from existing data and predict future sessions for each patient by using a single generator model. Compared with other popular GAN architectures, TR-GAN achieved the best performance in all evaluation metrics of two datasets. In two datasets expanded by TR-GAN, the balanced accuracy of AD vs. CN vs. MCI classification tasks can be increased by 2.56% and 3.61%, respectively. The balanced accuracy of sMCI vs. pMCI task can be increased by 2.50% and 4.00%, respectively.

REFERENCES

[1] C. R. Jack *et al.*, "The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods," *J. Magn. Reson. Imag.*, vol. 27, no. 4, pp. 685–691, 2008.

[2] C. Davatzikos, Y. Fan, X. Wu, D. Shen, and S. M. Resnick, "Detection of prodromal Alzheimer's disease via pattern classification of magnetic resonance imaging," *Neurobiol. Aging*, vol. 29, no. 4, pp. 514–523, Apr. 2008.

[3] B. Murugesan, S. V. Raghavan, K. Sarveswaran, K. Ram, and M. Sivaprakasam, "Recon-glgan: A global-local context based generative adversarial network for MRI reconstruction," in *Proc. Int. Workshop Mach. Learn. Med. Image Reconstruction*. Cham, Switzerland: Springer, 2019, pp. 3–15.

[4] L. de Toledo-Morrell *et al.*, "MRI-derived entorhinal volume is a good predictor of conversion from MCI to AD," *Neurobiol. Aging*, vol. 25, no. 9, pp. 1197–1203, Oct. 2004.

[5] J. Wen *et al.*, "Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation," *Med. Image Anal.*, vol. 63, Jul. 2020, Art. no. 101694.

[6] C. Han *et al.*, "GAN-based synthetic brain MR image generation," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 734–738.

[7] L. Zhang, A. Gooya, and A. F. Frangi, "Semi-supervised assessment of incomplete LV coverage in cardiac MRI using generative adversarial nets," in *Proc. Int. Workshop Simul. Synth. Med. Imag.* Cham, Switzerland: Springer, 2017, pp. 61–68.

[8] G. Kwon, C. Han, and D.-S. Kim, "Generation of 3D brain MRI using auto-encoding generative adversarial networks," in *Proc. Int. Conf. Med. Image Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2019, pp. 118–126.

[9] C. Bermudez, A. J. Plassard, L. T. Davis, A. T. Newton, S. M. Resnick, and A. B. Landman, "Learning implicit brain MRI manifolds with deep learning," *Proc. SPIE*, vol. 10574, Mar. 2018, Art. no. 105741L.

[10] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2223–2232.

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1125–1134.

[12] Z. Ning, Y. Zhang, Y. Pan, T. Zhong, M. Liu, and D. Shen, "LDGAN: Longitudinal-diagnostic generative adversarial network for disease progression prediction with missing structural MRI," in *Mach. Learn. Med. Imag.*, M. Liu, P. Yan, C. Lian, and X. Cao, Eds. Cham, Switzerland: Springer, 2020, pp. 170–179.

[13] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.

[14] D. Lee, J. Kim, W.-J. Moon, and J. C. Ye, "CollaGAN: Collaborative GAN for missing image data imputation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2487–2496.

[15] A. Chincarini *et al.*, "Integrating longitudinal information in hippocampal volume measurements for the early detection of Alzheimer's disease," *NeuroImage*, vol. 125, pp. 834–847, Jan. 2016.

[16] C. R. Jack *et al.*, "Comparison of different MRI brain atrophy rate measures with clinical disease progression in AD," *Neurology*, vol. 62, no. 4, pp. 591–600, Feb. 2004.

[17] C. Aguilar *et al.*, "Application of a MRI based index to longitudinal atrophy change in Alzheimer disease, mild cognitive impairment and healthy older individuals in the AddNeuroMed cohort," *Frontiers Aging Neurosci.*, vol. 6, p. 145, Jul. 2014.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[21] R. Cui, M. Liu, and G. Li, "Longitudinal analysis for Alzheimer's disease diagnosis using RNN," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 1398–1401.

[22] T. Wang, R. G. Qiu, and M. Yu, "Predictive modeling of the progression of Alzheimer's disease with recurrent neural networks," *Sci. Rep.*, vol. 8, no. 1, pp. 1–12, Dec. 2018.

[23] M. M. Ghazi *et al.*, "Training recurrent neural networks robust to incomplete data: Application to Alzheimer's disease progression modeling," *Med. Image Anal.*, vol. 53, pp. 39–46, Apr. 2019.

[24] W. Jung, E. Jun, and H.-I. Suk, "Deep recurrent model for individualized prediction of Alzheimer's disease progression," *NeuroImage*, vol. 237, Aug. 2021, Art. no. 118143.

[25] Y. Guo, Q. Chen, J. Chen, Q. Wu, Q. Shi, and M. Tan, "Auto-embedding generative adversarial networks for high resolution image synthesis," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2726–2737, Nov. 2019.

[26] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2337–2346.

[27] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101552.

[28] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[29] H.-C. Shin *et al.*, "Gandalf: Generative adversarial networks with discriminator-adaptive loss fine-tuning for Alzheimer's disease diagnosis from MRI," in *Proc. Int. Conf. Med. Image Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2020, pp. 688–697.

[30] C. Han *et al.*, "Synthesizing diverse lung nodules wherever massively: 3D multi-conditional GAN-based CT image augmentation for object detection," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 729–737.

[31] F. Rusak *et al.*, "3D brain MRI GAN-based synthesis conditioned on partial volume maps," in *Proc. Int. Workshop Simul. Synth. Med. Imag.* Cham, Switzerland: Springer, 2020, pp. 11–20.

[32] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang, "Learning to navigate for fine-grained classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 420–435.

[33] Y. Peng, X. He, and J. Zhao, "Object-part attention model for fine-grained image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1487–1500, Mar. 2017.

[34] Y. Chen, Y. Bai, W. Zhang, and T. Mei, "Destruction and construction learning for fine-grained image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5157–5166.

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

[36] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1501–1510.

[37] A. Routier *et al.*, "Clinica: An open-source software platform for reproducible clinical neuroscience studies," *Frontiers Neuroinform.*, vol. 15, Aug. 2021, Art. no. 689675, doi: 10.3389/fninf.2021.689675.

[38] J. Samper-González *et al.*, "Reproducible evaluation of classification methods in Alzheimer's disease: Framework and application to MRI and PET data," *NeuroImage*, vol. 183, pp. 504–521, Dec. 2018.

[39] J. Ashburner and K. J. Friston, "Unified Segmentation," *NeuroImage*, vol. 26, no. 3, pp. 839–851, 2005.

[40] N. J. Tustison *et al.*, "N4ITK: Improved N3 bias correction," *IEEE Trans. Med. Imag.*, vol. 29, no. 6, pp. 1310–1320, Jun. 2010.

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

[42] B. B. Avants, N. J. Tustison, M. Stauffer, G. Song, B. Wu, and J. C. Gee, "The insight toolkit image registration framework," *Frontiers Neuroinform.*, vol. 8, p. 44, Apr. 2014.

[43] V. Fonov, A. C. Evans, K. Botteron, C. R. Almli, R. C. McKinstry, and D. L. Collins, "Unbiased average age-appropriate atlases for pediatric studies," *NeuroImage*, vol. 54, no. 1, pp. 313–327, Jan. 2011.

[44] V. Fonov, A. Evans, R. McKinstry, C. Almli, and D. Collins, "Unbiased nonlinear average age-appropriate brain templates from birth to adulthood," *NeuroImage*, vol. 47, p. S102, Jul. 2009.

[45] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, Nov. 2003, pp. 1398–1402.

[46] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 3121–3124.

[47] N. Tzourio-Mazoyer *et al.*, "Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain," *NeuroImage*, vol. 15, no. 1, pp. 273–289, 2002.