Tensorizing GAN With High-Order Pooling for Alzheimer's Disease Assessment

Wen Yu, Baiying Lei, *Senior Member, IEEE*, Michael K. Ng, *Senior Member, IEEE*, Albert C. Cheung, *Member, IEEE*, Yanyan Shen, and Shuqiang Wang, *Member, IEEE*

Abstract—**It is of great significance to apply deep learning for the early diagnosis of Alzheimer's disease (AD). In this work, a novel tensorizing GAN with high-order pooling is proposed to assess mild cognitive impairment (MCI) and AD. By tensorizing a three-player cooperative game-based framework, the proposed model can benefit from the structural information of the brain. By incorporating the high-order pooling scheme into the classifier, the proposed model can make full use of the second-order statistics of holistic magnetic resonance imaging (MRI). To the best of our knowledge, the proposed Tensor-train, High-order pooling and Semisupervised learning-based GAN (THS-GAN) is the first work to deal with classification on MR images for AD diagnosis. Extensive experimental results on Alzheimer's disease neuroimaging initiative (ADNI) data set are reported to demonstrate that the proposed THS-GAN achieves superior performance compared with existing methods, and to show that both tensor-train and high-order pooling can enhance classification performance. The visualization of generated samples also shows that the proposed model can generate plausible samples for semisupervised learning purpose.**

Index Terms—**Alzheimer's disease (AD), high-order pooling, magnetic resonance (MR) images, semisupervised generative adversarial network (SS-GAN), tensor decomposition.**

I. Introduction

ALZHEIMER'S disease (AD) is an irreversible and chronic neurodegenerative disease with progressive impairment of memory and other mental functions. It is estimated to be the third leading cause of death after heart disease and cancer [1]. According to the World Alzheimer Report [2], the total estimated prevalence of AD was around 50 million worldwide in 2018, and the number will increase to 152 million × 2050. AD is caused by abnormal deposits of protein in the brain that destroys cells in the regions that control memory and mental functions. To date, AD is incurable but preventable. Early diagnosis of AD is crucial for timely therapy to slow the progression of the disease. Currently, the clinical diagnosis of AD heavily depends on clinical history [3]. The diagnosis procedure is time-consuming and requires extensive clinical training and experience for neurologists. Therefore, accurate AD assessment in its earliest stage by utilizing deep learning is highly desirable.

T1-magnetic resonance imaging (MRI) is significant for AD diagnosis in routine clinical practice. Early work for AD diagnosis using MR images primarily focused on traditional machine learning techniques [4], [5], which heavily relied on specific assumptions about brain structural abnormalities, such as regional cortical thickness, hippocampal volume, and gray matter volume. The performance of these manual feature extraction methods is limited since they require advanced clinical domain knowledge and complicated preprocessing steps. Therefore, they tend to be time-consuming and subjective. Besides, the brain is a huge network with complicated connections. The disease-related structure changes are subtle and scattered throughout the entire brain in different tissues. These kinds of patterns are difficult to learn since not all morphological abnormalities related to AD can be captured accurately, and the extracted regions of interest (ROIs) or voxel features are processed independently. Hence these features are unable to express the internal brain connections sufficiently.

Recent advances in machine learning especially deep learning have explosive popularity in computer vision and various medical applications [6], [7]. Instead of manually extracting features according to domain-specific knowledge, deep learning can discover the discriminant representations of images by incorporating feature extraction into the task learning process. However, most existing methods can only utilize the labeled data in a supervised manner. Annotation of MR images is laborious and costly, which requires clinical confirmation with great effort by experts. As a result, only small amounts of labeled MR images are available for AD assessment, and the unlabeled MR images cannot be used directly.

Generative adversarial network (GAN) has attracted much attention as it is capable of generating data without explicitly modeling the probability density function. It is intelligent for the discriminator to incorporate unlabeled data into the training process by utilizing the adversarial loss [8]. Furthermore, GAN

Manuscript received 28 November 2019; revised 22 November 2020; accepted 26 February 2021. Date of publication 17 March 2021; date of current version 2 September 2022. This work was supported in part by the National Natural Science Foundations of China under Grant 61872351; in part by the International Science and Technology Cooperation Projects of Guangdong under Grant 2019A050510030, Grant HKRGC GRF 12200317, Grant 12300218, Grant 12300519, and Grant 17201020; in part by the Strategic Priority CAS Project under Grant XDB38040200; and in part by the Shenzhen Key Basic Research Project under Grant JCYJ20200109115641762 and Grant RCYX20200714114641211. *(Wen Yu and Baiying Lei contributed equally to this work.) (Corresponding author: Shuqiang Wang.)*

Wen Yu, Yanyan Shen, and Shuqiang Wang are with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518060, China (e-mail: sq.wang@siat.ac.cn).

Baiying Lei is with the School of Biomedical Engineering, Health Science Center, Shenzhen University, Shenzhen 518060, China, and also with the National-Regional Key Technology Engineering Laboratory for Medical Ultrasound, Shenzhen 518060, China (e-mail: leiby@szu.edu.cn).

Michael K. Ng is with the Department of Mathematics, The University of Hong Kong, Hong Kong (e-mail: mng@maths.hku.hk).

Albert C. Cheung is with the Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: albertccheung@yahoo.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3063516.

Digital Object Identifier 10.1109/TNNLS.2021.3063516

has been proven to be feasible in data augmentation, image-to-image translation, and semisupervised learning (SSL). To make full use of both labeled and unlabeled MR images, semisupervised GAN (SS-GAN) [9]–[12] can be adopted. In this article, our primary goal is to leverage GAN to characterize the high-order distribution of MR images for semisupervised classification. In particular, we discovered that the recently introduced triple-GAN could alleviate the instability and incompatible problems of the SS-GAN [12]. Triple-GAN designed a three-player cooperative game instead of the conventional two-player competition game by introducing the auxiliary classifier network based on generator and discriminator. Inspired by this, our model exploits the three-player cooperative game for modeling MR images to assess mild cognitive impairment (MCI) and AD.

Based on these observations, in this article, we propose a novel Tensorizing GAN with High-order pooling to assess MCI and AD. More specifically, in order to stabilize the training of GAN and speed up the convergence, the proposed model utilizes the compatible learning functions of the three-player cooperative game. Our proposed model is called THS-GAN, i.e., Tensor-train decomposition, Higher order pooling, and Semisupervised learning are employed in the proposed GAN model. Instead of vectorizing each layer as conventional GAN, the tensor-train decomposition is applied to all layers in classifier and discriminator, including fully connected layers and convolutional layers. Thus the number of parameters can be reduced significantly. Besides, in such a tensor-train format, our model can benefit from the structural information of the brain. Moreover, compared with the first-order pooling, the high-order pooling module can extract more significant features by making full use of the second-order statistics of the holistic MR image. Thus our model also exploits the Global Second-order Pooling (GSP) block as a high-order pooling module in the classifier. In particular, the GSP block can capture the long-range dependences of features at distant positions by computing all pairwise channel correlations of the 4-D feature-maps extracted by 3D-DenseNet. Thus both GSP and 3D-DenseNet are integrated into the classifier to enhance salient feature channels and suppress less-useful feature channels. As a result, useful features related to anatomical abnormalities are extracted in a self-attention manner to improve the performance of classification. The contributions of this article are summarized as follows.

1) By tensorizing the three-player cooperative game-based framework, the proposed model can benefit from the structural information of the brain.
2) The proposed THS-GAN leverages the high-order pooling to make full use of the second-order statistics of the holistic MR images. The long-range dependences between slices of different directions can be captured effectively. Thus more significant features can be extracted automatically in a self-attention manner to boost the predictive performance.
3) The THS-GAN model is designed to assess MCI and AD in a semisupervised manner to take advantage of both labeled and unlabeled MR images.

The rest of this article is organized as follows. We review the related work in Section II. In Section III, we present the proposed THS-GAN in detail. In Section IV, THS-GAN is tested with various configurations, and experimental results are presented to demonstrate its advantage. Finally, concluding remarks and future work are discussed in Section V.

II. Related Work

The current AD diagnosis model can be categorized into two types: the traditional machine learning-based approach and the deep learning-based approach.

The traditional machine learning techniques can be divided further into three categories: voxel-based approach, ROI-based approach, and patch-based approach. Although the voxel-based approach [13] is intuitive and straightforward in terms of interpretation, the process of classification is computationally expensive since the voxelwise features are of extremely high dimensionality, and the classification performance will deteriorate due to the "curse of dimensionality" [14]. For the ROI-based approach [4], the ROIs are segmented by prior hypothesis, but the abnormal regions related to AD may not fit the predefined ROIs ideally in practice, and the features extracted from ROIs are very coarse in the sense that they cannot sufficiently represent all subtle changes involved in the brain diseases. As a result, the representation power of ROI features is limited. Patch-based approach dissected brain areas into small 3D-patches, followed by extracting features from each selected patch individually, and then the features are combined hierarchically in a classifier level [15]. However, the features extracted by these methods neglect the correlated variations of the whole brain structure affected by AD in other regions. Besides, the extraction of these handcrafted features heavily depends on how well the images are registered and segmented, which often require the domain expert knowledge.

In the application domain of AD diagnosis, the previous deep learning studies focused on two directions: 1) CNN is utilized for supervised classification, primarily by using large-scale annotated data sets and 2) unsupervised GAN is exploited for data synthesis or image-to-image translation [16], [17]. In the first approach, Islam and Zhang [18] presented a method based on 2D-DenseNet. The MR images are sliced in three directions (axial, coronal, and sagittal). Then three parallel 2D-DenseNets are evaluated on MRI slices separately. Finally, the results are fused for AD diagnosis. However, the way of converting a 3D-image into a series of 2D-slices causes CNNs to disregard the spatial information of 3-D space, and different slicing methods lead to loss of features. Thus many studies focus on 3D-CNN instead of 2-D to alleviate this issue. For instance, Wang *et al.* [19] proposed an ensemble of 3-D densely connected convolutional networks (3D-DenseNets) for AD and MCI diagnosis. In the second approach, Pan *et al.* [16] imputed the missing PET images by learning bidirectional mappings between MRI and PET via 3D-cGAN. Then, based on the complete MRI and PET (after imputation), they develop a landmark-based multimodal multi-instance learning method (LM3IL) for AD diagnosis. Karim Armanious and Jiang [17] proposed the Cycle-MedGAN framework based on the traditional Cycle-GAN with new nonadversarial losses for PET to CT translation. Wang *et al.* [20] proposed a 3-D autocontext-based locality adaptive multimodality GAN model

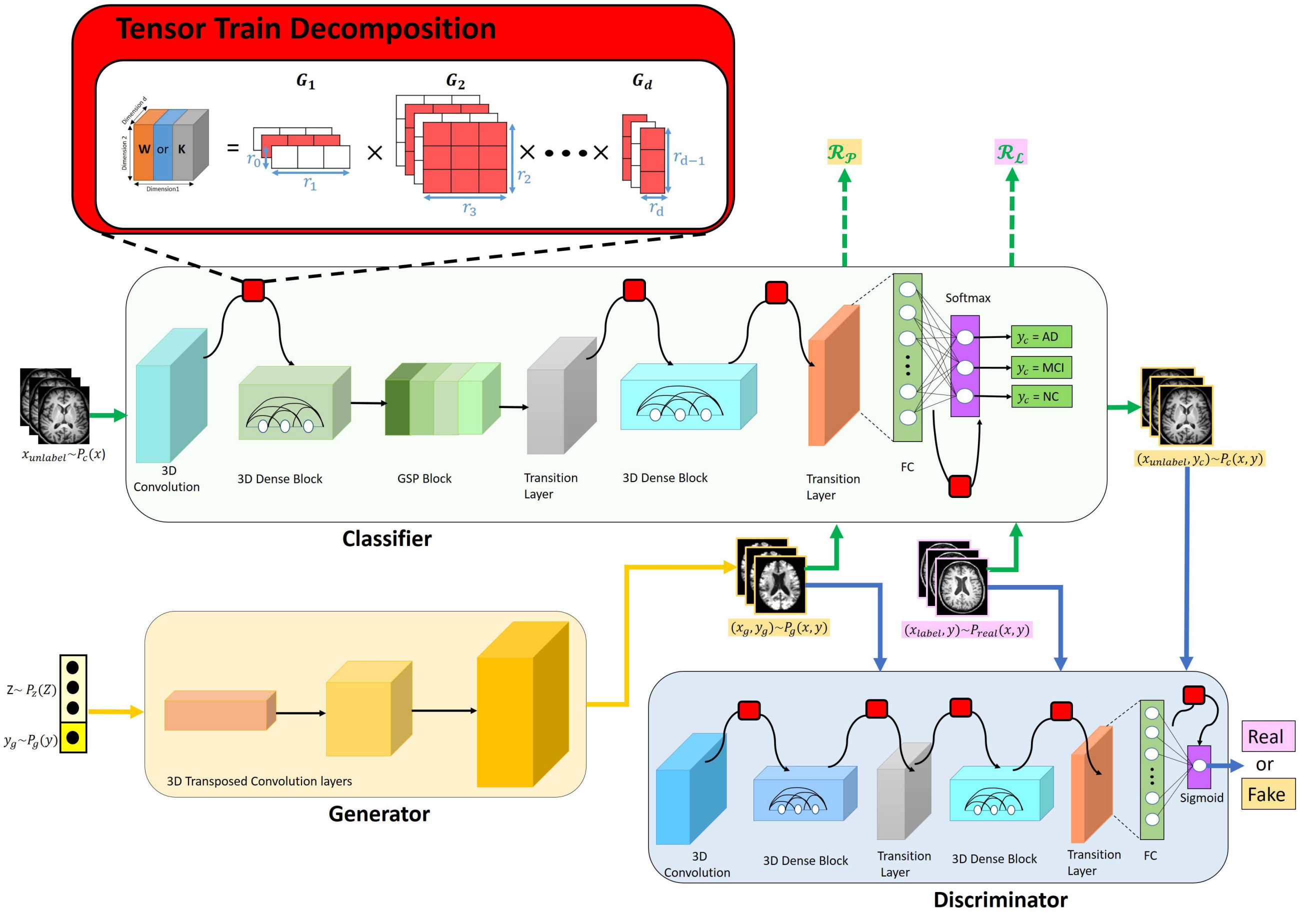


Fig. 1. Illustration of THS-GAN (best viewed in color). Real and Fake are the adversarial losses. $\mathcal{R}_{\mathcal{L}}$ and $\mathcal{R}_{\mathcal{P}}$ denote the cross-entropy loss for supervised learning for real data and generated data, respectively. $\mathcal{R}_{\mathcal{L}}$ and $\mathcal{R}_{\mathcal{P}}$ are unbiased regularizations that ensure the consistency between p_g, p_c, and p_{real}, which are the distributions defined by the generator, classifier and true data, respectively.

(LA-GANs) to synthesize the high-quality FDG-PET image from the low-dose one with the MR images that provide anatomical information.

The previous GAN applications focus on image synthesis and image-to-image translation. However, different from the previous GAN applications, the aim of the proposed THS-GAN is for AD classification in a semisupervised manner with less annotated MR images. We remark that the research of GAN adaptation in MR images is still under development.

III. Proposed THS-GAN Method

A. Overview

Fig. 1 summarizes the architecture of the proposed THS-GAN. After data preprocessing (see Section IV-A), the normalized MR images are fed into THS-GAN. Since the input MR images are high-order with complicated brain structure, we modify the triple-GAN with the following four significant improvements: 1) instead of 2-D transposed convolution, 3-D transposed convolution is utilized in the generator to generate MR images; 2) 3D-DenseNet [21], [22] is adopted in both the classifier and discriminator to extract subtle features related to AD within the limited receptive field at a local level; 3) all layers in classifier and discriminator are compressed by tensor-train decomposition; and 4) the high-order pooling module GSP block is incorporated into the classifier to make full use of the correlation within feature-maps along the channel axis to capture more discriminative features at the global level to represent the holistic brain. The details of the proposed method will be presented in Section III-B.

B. Architecture

The proposed THS-GAN is designed for semisupervised classification. Input data x is partially labeled and y represents the corresponding label. $p_{\text{real}}(x)$ denotes the empirical distribution of input data and $p_{\text{real}}(y)$ is assumed as the distribution of labels on partially annotated data. The goal is to predict the label y for both labeled and unlabeled data x as well as to the new generated samples x conditioned on y. As the label y is incomplete, our density model should characterize the uncertainty of both x and y, thus the joint distribution $p_{\text{real}}(x, y)$ of image-label pairs can be calculated in two ways: $p_{\text{real}}(x, y) = p_{\text{real}}(y)p_{\text{real}}(x|y)$ and $p_{\text{real}}(x, y) = p_{\text{real}}(x)p_{\text{real}}(y|x)$. The conditional distributions $p_{\text{real}}(x|y)$ and $p_{\text{real}}(y|x)$ are learned by the class-conditional generator and auxiliary classifier, respectively. Thus the proposed THS-GAN consists of three networks: 1) a class-conditional generator that approximately characterizes the conditional distribution $p_g(x|y) \approx p_{\text{real}}(x|y)$; 2) a classifier that approximately characterizes the conditional distribution in the opposite direction $p_c(y|x) \approx p_{\text{real}}(y|x)$; and 3) a discriminator that distinguishes whether the image-label pair (x, y) comes from the real data distribution $p_{\text{real}}(x, y)$.

More specifically, in the three-player game as illustrated in Fig. 1, a sample x_{unlabel} is drawn from $p_c(x)$, classifier predict label y_c given x_{unlabel} following the conditional distribution $p_c(y|x)$. Hence, the pseudo image-label pair $(x_{\text{unlabel}}, y_c)$ is from the joint distribution $p_c(x, y) = p_c(x)p_c(y|x)$. Similarly, a pseudo image-label pair (x_g, y_g) is produced by generator given $y_g \sim p_g(y)$ by utilizing $x|y \sim p_g(x|y)$, hence forming the joint distribution $p_g(x, y) = p_g(y)p_g(x|y)$. With respect to $p_g(x|y)$, x_g is transformed by generator given label y_g and the latent variables z. $x_g = G(y_g, z)$, $z \sim p_z(z)$, where $p_z(z)$ is a distribution (e.g., uniform or standard normal). Then the pseudo image-label pairs $(x_{\text{unlabel}}, y_c)$ and (x_g, y_g) are fed into the discriminator for identification. Discriminator will identify the image-label pairs from real data distribution as positive samples, and discriminator D is trained to maximize the probability of assigning the correct label to both real samples and fake samples from generator G and classifier C. To achieve equilibrium that the joint distributions defined by classifier and generator both converge to real data distributions, compatible function of adversarial loss is defined as follows:

$$\begin{aligned}
\min_{C,G}\max_{D}\, & U(C, G, D)\\
&= \mathbb{E}_{(x_{\text{label}}, y)\sim p_{\text{real}}(x,y)}\big[\log D(x_{\text{label}}, y)\big]\\
&\quad + \alpha\mathbb{E}_{(x_{\text{unlabel}}, y_c)\sim p_c(x,y)}\big[\log(1 - D(x_{\text{unlabel}}, y_c))\big]\\
&\quad + (1-\alpha)\mathbb{E}_{(x_g, y_g)\sim p_g(x,y)}\big[\log\big(1 - D\big(x_g, y_g\big)\big)\big]\\
&= \mathbb{E}_{(x_{\text{label}}, y)\sim p_{\text{real}}(x,y)}\big[\log D(x_{\text{label}}, y)\big]\\
&\quad + \alpha\mathbb{E}_{x_{\text{unlabel}}\sim p_c(x)}\big[\log(1 - D(x_{\text{unlabel}}, C(x_{\text{unlabel}})))\big]\\
&\quad + (1-\alpha)\mathbb{E}_{z\sim p_z(z), y_g\sim p_g(y)}\big[\log\big(1 - D\big(G\big(z, y_g\big), y_g\big)\big)\big]
\end{aligned} \tag{1}$$

where C, G, and D are individual networks. C and D are represented by tensor-train layers (TT-layers). $\mathbb{E}_{(x_{\text{label}}, y)\sim p_{\text{real}}(x,y)}$ denotes the expectation over the real labeled data. $\mathbb{E}_{x_{\text{unlabel}}\sim p_c(x)}$ is the expectation over the real unlabeled data produced by the classifier, and $\mathbb{E}_{(x_g, y_g)\sim p_g(x,y)}$ is the expectation over the fake data produced by the generator. $D(x_{\text{label}}, y)$ represents the probability that image-label pair came from the real labeled data. Meanwhile, $D(x_{\text{unlabel}} , y_c)$ and $D(x_g, y_g)$ represent the probability that image-label pair came from fake data produced by classifier and generator, respectively. $\alpha \in (0, 1)$ is a constant that controls the relative importance of generation and classification, and we use the fixed value of 0.5. The game defined in (1) achieves its equilibrium if and only if $p_{\text{real}}(x, y) = \alpha p_c(x, y) + (1 - \alpha)p_g(x, y)$. The equilibrium indicates that if one of classifier and generator tends to the real data distribution, the other will also go toward the data distribution, which addresses the competing problem of the conventional SS-GAN. Note that the conventional SS-GAN only contains two players: generator and discriminator. The discriminator shares incompatible roles of identifying fake samples and predicting real labels simultaneously, and the generator estimates the data without considering the labels. By utilizing the three-player cooperative game, both the classifier and generator will converge to the real data distribution if the model has been trained to achieve the optimum. In this manner, the class-conditional generator can disentangle different modalities and generate MR images to cover all classes (AD, MCI, and NC). On the other hand, the discriminator is trained with dissimilar samples from various classes (AD, MCI, and NC) to provide gradients for the generator. Hence, the mode collapse problem is alleviated.

As aforementioned, layers are tensorized as TT-layer and we treat the elements of the TT-cores as the parameters of the layer. TT-layers of classifier and discriminator are represented as various TT-cores G_k of elements θ_c and θ_d, respectively. The classifier is updated by descending along its stochastic gradient according to $C\_$ loss with respect to all the elements θ_c of TT-cores. The classifier loss function $C\_$ loss is composed of two parts: the supervised loss and the unsupervised loss

$$\underbrace{\frac{\partial C\_\text{loss}}{\partial G_k[i_k, j_k]}}_{\mathbf{r}_{k-1}\times\mathbf{r}_k} = \nabla_{\theta_c}\big[L_{\text{supervised}} + L_{\text{unsupervised}}\big]. \tag{2}$$

The supervised loss function is defined by the cross-entropy loss of real image-label samples and generated image-label samples in a supervised learning setting

$$L_{\text{supervised}} = \mathcal{R}_{\mathcal{L}} + \alpha_{\mathcal{P}}\mathcal{R}_{\mathcal{P}} \tag{3}$$

$$\mathcal{R}_{\mathcal{L}} = \mathbb{E}_{(x_{\text{label}}, y)\sim p_{\text{real}}\,(x,y)}\big[-\log p_c(y|x_{\text{label}})\big] \tag{4}$$

$$\mathcal{R}_{\mathcal{P}} = \mathbb{E}_{(x_g, y_g)\sim p_g(x,y)}\big[-\log p_c\big(y_g|x_g\big)\big]. \tag{5}$$

The cross-entropy loss of real labeled data distribution for classifier is defined as $\mathcal{R}_{\mathcal{L}}$, which is equivalent to model the KL-divergence between $p_c(x, y)$ and $p_{\text{real}}(x, y)$. As the generated data can also be used for boosting classification performance, the cross-entropy loss of synthesis data is defined as $\mathcal{R}_{\mathcal{P}}$, which optimizes classifier on the samples produced by generator in the supervised manner. Minimizing $\mathcal{R}_{\mathcal{P}}$ with respect to classifier is equivalent to minimizing $D_{KL}(p_g(x, y)\|p_c(x, y))$. Note that directly minimizing $D_{KL}(p_g(x, y)\|p_c(x, y))$ is infeasible since the unknown likelihood ratio $p_g(x, y)/p_c(x, y)$ cannot be computed directly. $\alpha_{\mathcal{P}}$ is the weight hyperparameter fixed as 0.05.

The unsupervised loss is the adversarial loss of standard GAN minimax game

$$L_{\text{unsupervised}} = \mathbb{E}_{x_{\text{unlabel}}\sim p_c(x)}\big[\log(1 - D(x_{\text{unlabel}}, C(x_{\text{unlabel}})))\big]. \tag{6}$$

In other words, the unsupervised loss is computed to distinguish real and fake image-label pairs. The supervised loss computes the cross-entropy for real classes. In this work, these classes are AD, MCI, and NC.

The generator loss is defined as

$$G\_loss = \sum_{(x_g, y_g)} \log\big(1 - D\big(x_g, y_g\big)\big) + \lambda\|x_{\text{label}} - x_g\|_{L1}. \tag{7}$$

The L1 reconstruction loss is integrated with the adversarial loss to impose an additional constraint on the generator. Thus, the generator needs to fool the discriminator while minimize the absolute pixelwise intensity distance between synthetic MR images x_g and real MR images x_{label} simultaneously. This encourages the generator to produce MR images as close as possible to ground truth images. λ is set as 0.01 empirically to balance the relative importance between adversarial loss and reconstruction loss.

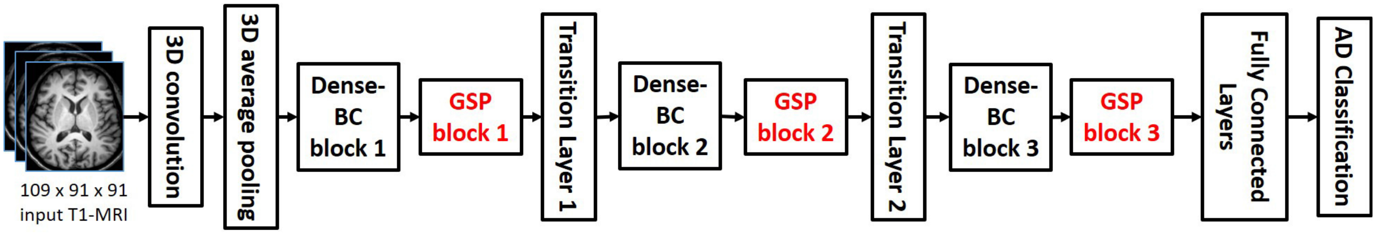


Fig. 2. Classifier framework is composed of 3D-DenseNet and GSP block. Note that only one GSP block is inserted at one of three optional positions in red. There is no GSP block in discriminator.

The discriminator is updated by descending along its stochastic gradient according to $D\_$ loss for all the elements θ_d of TT-cores

$$\underbrace{\frac{\partial D\_loss}{\partial G_k[i_k, j_k]}}_{\mathbf{r}_{k-1}\times\mathbf{r}_k} = \nabla_{\theta_d}\Bigg[\sum_{(x_{\text{label}},y)} \log D(x_{\text{label}}, y) + \alpha \sum_{(x_{\text{unlabel}},y_c)} \log(1 - D(x_{\text{unlabel}}, y_c)) + (1-\alpha)\sum_{(x_g,y_g)} \log\big(1 - D\big(x_g, y_g\big)\big)\Bigg]. \quad (8)$$

Intuitively, a sound generator can produce meaningful labeled data beyond the training set as auxiliary information for the classifier, which will improve the predictive performance, and vice versa, a sound classifier will boost the performance of the generator. As a result, both the classifier and generator can improve mutually. Moreover, the discriminator can utilize the label information of the unlabeled data through the classifier and then assist the generator to generate correct image-label pairs. Therefore, THS-GAN is more likely to reach Nash equilibrium.

Two components of triple-GAN (classifier and discriminator) are converted to the tensor-train format (TT-format) [23]–[25]. We refer to 1-D data as a vector, denoted as v. 2-D array is matrix, denoted as $\boldsymbol{V}$, and higher dimensional array is tensor, denoted as $\mathcal{V}$. To refer one specific element from a tensor, we use $\mathcal{V}(i) = \mathcal{V}(i_1, i_2, \ldots, i_d)$, where d is the dimensionality of the tensor $\mathcal{V}$ and i is the index vector. Our proposed THS-GAN ingests T1-MRI image as 3-D tensor, where each dimension corresponds to height, width, and slice, respectively. A d-dimensional $n_1 \times n_2 \times \cdots \times n_d$ tensor $\mathcal{V}$ can be represented in the TT-format [25], [26] as

$$\mathcal{V}(i_1, i_2, \ldots, i_d) = G_1[i_1]G_2[i_2]\cdots G_d[i_d] \quad (9)$$

where $G_k[i_k]$ is an $r_{k-1} \times r_k$ matrix, which is one slice from the 3-D array G_k. The elements of the collection $\{r_k\}_{k=0}^{d}$ are called TT-ranks. $r_0 = r_d = 1$ is the boundary condition to keep the matrix product (9) of size 1×1.

The collections of matrices $\{\{G_k[j_k]\}_{j_k=1}^{n_k}\}_{k=1}^{d}$ are called TT-cores [24]. The TT-format requires $\sum_{k=1}^{d} n_k r_{k-1} r_k$ parameters to represent a tensor $\mathcal{V} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ which has $\prod_{k=1}^{d} n_k$ elements. The TT-ranks r_k control the trade-off between the number of parameters and the accuracy of the representation. The smaller the TT-ranks, the more memory efficient the TT-format is. But if the TT-ranks are set too small, the accuracy might deteriorate due to information loss caused by overcompressing. Such a representation is memory-efficient to store high-order data. Meanwhile, the significant structural information of data can be preserved. These properties are suitable for representing MR images. In the following, we introduce tensor-train decomposition for fully-connected layers and convolutional layers, respectively.

1) Fully-Connected Layers Tensor-Train Decomposition: The fully-connected layer is applied to an input N-dimensional vector X

$$Y = \boldsymbol{W}X + B \quad (10)$$

where the weight matrix $\boldsymbol{W} \in \mathbb{R}^{M\times N}$ and the bias vector $B \in \mathbb{R}^{M}$ define the linear transformation. A TT-fully-connected-layer transforms a d-dimensional tensor $\mathcal{X}$ (which is constructed from the corresponding vector X) to the d-dimensional tensor $\mathcal{Y}$ (which corresponds to the output vector Y) by factorizing the weight matrix $\boldsymbol{W}$ into the TT-format with the TT-cores $G_k[i_k, j_k]$. Thus the linear transformation [see (10)] of a fully connected layer can be represented in the TT-layer

$$\mathcal{Y}(i_1, \ldots, i_d) = \sum_{j_1,\ldots,j_d} G_1[i_1, j_1]\cdots G_d[i_d, j_d]\mathcal{X}(j_1, \ldots, j_d) + \mathcal{B}(i_1, \ldots, i_d) \quad (11)$$

where $G[i_d, j_d] \in \mathbb{R}^{r_{k-1}\times r_k}$ is a slice of cores as illustrated in the red part of Fig. 1. Since the fully connected layer is a special case of the convolutional layer with kernel size $1 \times 1 \times 1$, such TT-format can also be applied to convolutional layers in a similar manner.

2) Convolutional Layers Tensor-Train Decomposition: 3-D convolution is an extension of 2-D convolution with one more spatial dimension in terms of slice with respect to MRI volume. The traditional 3-D convolutional layer transforms the 4-D input tensor $\mathcal{X} \in \mathbb{R}^{W\times H\times L\times C}$ into the output $\mathcal{Y} \in \mathbb{R}^{W'\times H'\times L'\times S}$ by convolving $\mathcal{X}$ with the kernel $\mathcal{K} \in \mathbb{R}^{\ell\times\ell\times\ell\times C\times S}$

$$\mathcal{Y}(x, y, z, s) = \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\sum_{k=1}^{\ell}\sum_{c=1}^{C}\mathcal{K}(i, j, k, c, s) \times\mathcal{X}(x+i-1, y+j-1, z+k-1, c). \quad (12)$$

When stride is set as 1 and there is no zero padding, $W' = W-l+1$, $H' = H-l+1$ and $L' = L-l+1$. The tensor-train decomposition is applied to the convolutional kernel $\mathcal{K}$ as follows:

$$\mathcal{K}(x, y, z, c, s) = G_0[i, j, k]G_1[c_1, s_1]\cdots G_d[c_d, s_d]. \quad (13)$$

Red part of Fig. 1 also presents an illustration for (13), and the 3-D convolutional layer is converted to TT-layer as follows:

$$\mathcal{X}(x,y,z,c) \overset{\text{reshape}}{\longrightarrow} \tilde{\mathcal{X}}(x,y,z,c_1,c_2,\ldots,c_d) \quad (14)$$

$$\mathcal{Y}(x,y,z,s) \overset{\text{reshape}}{\longrightarrow} \tilde{\mathcal{Y}}(x,y,z,s_1,s_2,\ldots,s_d) \quad (15)$$

and

$$\begin{aligned}\tilde{\mathcal{Y}}&(x,y,z,s_1,\ldots,s_d)\\ &= \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\sum_{k=1}^{\ell}\sum_{c_1,\ldots,c_d} G_0[i,j,k]G_1[c_1,s_1]\cdots G_d[c_d,s_d]\\ &\quad\times\tilde{\mathcal{X}}(i+x-1,j+y-1,k+z-1,c_1,\ldots,c_d) \quad (16)\end{aligned}$$

where $c = \prod_{i=1}^{d} c_i$, $s = \prod_{i=1}^{d} s_i$ and d is the number of TT-cores. By replacing the 4-D convolutional kernel with approximations using lower rank matrices, redundancy in convolutional layers can be removed implicitly. It is worth noting that although applying tensor-train decomposition to neural networks can achieve a large factor of compression, finding optimal TT-ranks remains difficult [23], [27]. The TT-layer is compatible with the existing training algorithms for neural networks because all the derivatives required by the backpropagation algorithm can be computed using the properties of the TT-format.

THS-GAN has a generator network composed of six transposed convolutional layers with 3 × 3 × 3 kernel. Each transposed convolutional layer is followed by batch normalization (BN) and ReLU except the last layer. The tanh is utilized in the last layer. Furthermore, the conditional variable y is concatenated to each transposed convolutional layer except the last layer. DenseNet [21] is utilized in both the classifier and discriminator. We expand it to 3D-DenseNet by adding a spatial dimension to all convolutional and pooling layers in DenseNet for MRI volume. Feature-maps learned by all preceding layers are concatenating along the last dimension for the subsequent layers. Through such dense connectivity, feature-maps are reused and the vanishing-gradient problem is alleviated. Meanwhile, 3D-DenseNet can extract the local features related to AD from the whole volumes efficiently. The details of 3D-denseNet can be referred to [21] and [22]. In this article, the depth is set as 30, the growth rate is set as 12, the number of the Dense-BC block is set as 3, and the reduction is set as 0.5. In particular, since the discriminator of THS-GAN distinguishes between the synthesis distribution and the target real distribution based on the pairs of generated samples x_g and conditional variable y_g, each convolutional layer is also concatenated to y_g along the channel axis in the discriminator.

Furthermore, the high-order pooling module GSP block can make full use of the second-order statistics of the holistic MR images. The long-range dependences between slices of different directions can be effectively captured for extracting more significant features in a self-attention manner. Thus, the GSP block is added after the Dense-BC block in the classifier, as illustrated in Fig. 2, aiming to learn more discriminative representations by recalibrating the 4-D channelwise feature-maps. There is one more GSP block (in red), which can be positioned at: 1) GSP block 1; 2) GSP block 2; or 3) GSP block 3.

Inspired by the study [28], the GSP block is extended to a 4-D tensor as illustrated in Fig. 3. Given a 4-D feature map outputted by a previous Dense-BC block, we first perform GSP to model pairwise channel correlations of the holistic feature map. Then the resulting covariance matrix is processed by convolutions and nonlinear activations, which is finally used for scaling the 4-D feature map along the channel dimension.

More specifically, the GSP block consists of two modules: a squeeze module and an excitation module. The squeeze module aims to model the second-order statistics along the channel dimension of the input feature map for capturing channel dependence. Consider a 4-D feature map of $h' \times w' \times l' \times c'$ as an input, where h' is the spatial height of the feature-map, w' is the width, l' is depth, and c' is the number of channels. It can be seen as c' cubes where each cube is of size $h' \times w' \times l'$. First, $1 \times 1 \times 1$ convolution is utilized to reduce the number of channels from c' to c ($c < c'$) to decrease the computational cost of the following operations. For the $h' \times w' \times l' \times c$ tensor of reduced dimensionality, the pairwise channel correlations are computed to one $c \times c$ covariance matrix. The resulting covariance matrix has a clear physical meaning, its ith row indicates the statistical dependence of channel i with all channels. As the quadratic operations involved change the order of data, row-wise normalization is performed for the covariance matrix with respect to the structural information of the brain. To simplify the block design and to find the appropriate trade-off between computational complexity and classification accuracy, we calculate the size of the covariance matrix as $c = c'/6$ in a self-adaptive manner.

The excitation module aims to scale the channel for feature recalibration. In the excitation module, before channel scaling, we perform two consecutive operations of convolution and nonlinear activation for the covariance matrix. To maintain the structural information, the covariance matrix is processed with row-wise convolution, which is followed by a leaky rectified linear unit (LReLU). Then we perform the second convolution and the sigmoid function as a nonlinear activation to compute the weight vector of [$\mathbf{W}_1$, $\mathbf{W}_2$,..., $\mathbf{W}_{c'}$]. The final output of the GSP block is obtained by operating the dot product between the weight vector [$\mathbf{W}_1$, $\mathbf{W}_2$,..., $\mathbf{W}_{c'}$] and the respective channels [Channel 1, Channel 2,..., Channel c']. Individual channels are thus emphasized or suppressed in this soft manner in terms of the weights. Thus the discriminative features related to AD lesions are enhanced, and redundant features are suppressed. As shown in Fig. 3, the feature map output by the GSP block is close to the benchmark with less redundant features, and all significant features are discovered. On the other hand, the feature map without high-order pooling includes more redundant features compared with the benchmark [29].

Furthermore, the network structure of each component in THS-GAN is further optimized from the following perspectives.

1) For discriminator and generator networks, the condition variable y is concatenated with each convolutional layer and transposed convolutional layer as additional channels, respectively.
2) As suggested by Radford *et al.* [11], BN is utilized to both the discriminator and the generator in

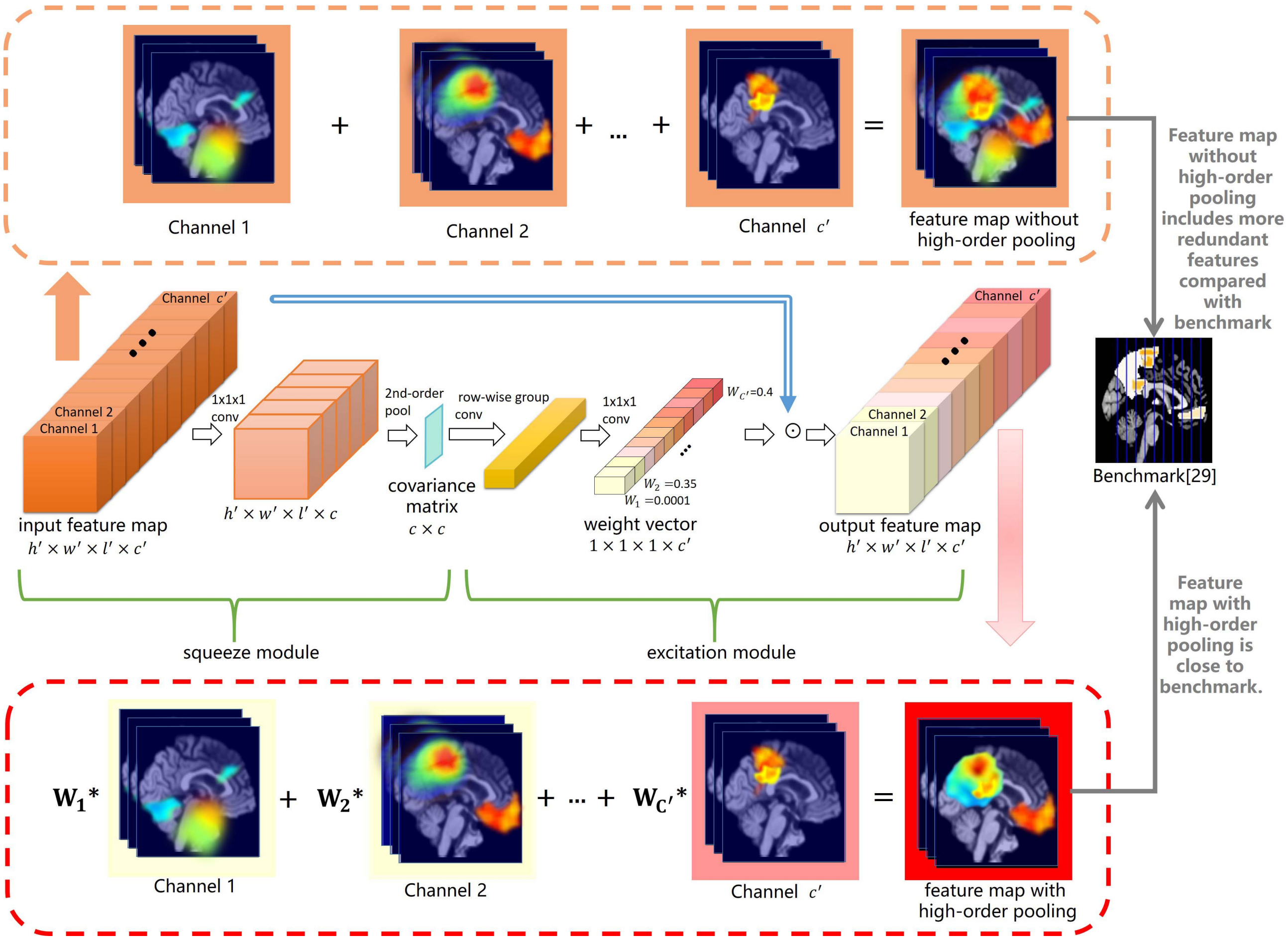

Channel 1
+
Channel 2
+ ... +
Channel c'
=
feature map without high-order pooling
Feature map without high-order pooling includes more redundant features compared with benchmark
Channel c'
Channel 2
Channel 1
input feature map
$h' \times w' \times l' \times c'$
1x1x1 conv
$h' \times w' \times l' \times c$
2nd-order pool
covariance matrix
$c \times c$
row-wise group conv
1x1x1 conv
$W_{c'}$=0.4
W_2 =0.35
W_1 =0.0001
weight vector
$1 \times 1 \times 1 \times c'$
⊙
Channel c'
Channel 2
Channel 1
output feature map
$h' \times w' \times l' \times c'$
Benchmark[29]
squeeze module
excitation module
Feature map with high-order pooling is close to benchmark.
$\mathbf{W_1}$*
Channel 1
+ $\mathbf{W_2}$*
Channel 2
+ ... + $\mathbf{W_{C'}}$*
Channel c'
=
feature map with high-order pooling

Fig. 3. High-order pooling module GSP block. Given an input 4-D feature map, $1 \times 1 \times 1$ convolution is performed to reduce dimension. Then the covariance matrix is computed followed by convolution and nonlinear activation, finally a weight vector is produced to recalibrate the feature map along the channel dimension. The high-order pooling can capture the dependence of features at distant positions by computing all pairwise channel correlations. As a result, significant features will be enhanced. As each channel corresponds to a particular feature, each feature map of all channels is considered as a feature set that can map back to individual voxels of input MR image. The discriminative features related to AD are shown in the benchmark [29].

the THS-GAN model to prevent the generator from collapsing all the samples to a single point. However, adding BN to all layers causes model instability. Hence we also avoid using BN in the generator output layer and the discriminator input layer as they suggest.

3) The first order pooling (average pooling) is still utilized since the GSP block cannot reduce the dimensions of the feature-map resulting in a large number of parameters. Thus the first order pooling is combined with GSP block to abstract the discriminative representations so that the proposed THS-GAN model can take advantage of both first-order and second-order statistics for AD diagnosis.

IV. Experiments and Results

A. Data Set and Preprocessing

T1-weighted MR images from the Alzheimer's Disease Neuroimaging Initiative (ADNI[1]) public data set are used for the evaluation purpose. The ADNI study involves more than 1000 participants including normal control (NC), MCI, and AD subjects. All subjects will have cognitive assessments, and they will have MRI scans at regular intervals (6 or 12 months) throughout the study. A total of 833 MR images are utilized. They are collected from 624 participants including both male and female, and their ages range from 70 to 90. Since a certain participant's brain structure makes a progressive change after a period of time, two scans with the longest interval of one participant will be chosen as different subjects, as long as the interval is more than three years. In this manner, 221 AD subjects, 297 MCI subjects, and 315 NC subjects are collected, respectively. Table I lists the demographic characteristics of the subjects.

All MR images have already been processed with Gradwrap, B1 nonuniformity correction, and N3 correction using standard methodology from ADNI. FSL[2] toolbox is utilized to preprocess MR images following three steps: 1) removal of redundant tissues; 2) brain extraction by FSL-BET; and 3) linear registration to the MNI152 template by FSL-FLIRT [30]. The dimension of each image is 109 × 91 × 91 in the neuroimaging informatics technology initiative (NIfTI) file format. Each image comprises 109 2-D slices of 91 × 91.

To evaluate the effectiveness of our model, we set up three groups of experiments: 1) AD versus NC; 2) MCI versus NC; and 3) AD versus MCI classification. It is worth noting that the second classification is significant to distinguish MCI from

[1] http://adni.loni.usc.edu/

[2] www.fmrib.ox.ac.uk/fsl

TABLE I

DEMOGRAPHIC CHARACTERISTICS OF THE SUBJECTS

Subject	NC	MCI	AD
Number	315	297	221
Gender(F/M))	150/165	157/140	100/121
Age	77.7±5.4	76.0±7.2	76.6±7.5
Education	16.0±2.9	15.6±3.1	14.6±3.2
MMSE	29.1±1.2	25.8±3.6	21.5±4.4
CDR	0±0.19	0.57±0.28	0.93±0.49

NC for early diagnosis so that timely therapeutic interventions can be carried out to slow down the progression of MCI to AD.

The MR image is normalized into the range [−1,1], and the whole volume of 109 × 91 × 91 voxels is fed into the proposed THS-GAN model as a tensor directly without compressing or downsizing to ensure no information loss. No data augmentation was used. For evaluation, 80% of the MR images are allocated for training. The remaining 20% of the MR images are equally partitioned and used as validation and test data sets, respectively. For avoiding prediction bias, the training set, validation set, and test set do not have the MR images from the same subject simultaneously. The validation data set is utilized to tune hyperparameters to obtain the best model out of several epochs during the training process.

B. Experimental Setup

The proposed THS-GAN model is trained on the ADNI data set from scratch in an end-to-end manner, and it is implemented by TensorFlow.[3] The experiments are conducted on NVIDIA GeForce GTX 1080 GPU. The initial learning rate is 0.01 and will decrease to 10^{-3} at 75 epochs and 10^{-4} at 110 epochs. Stochastic gradient descent (SGD) optimizer with Nesterov momentum [31] of coefficient 0.9 is utilized in classifier and discriminator. Meanwhile, the Adam optimizer with $\beta_1 = 0$ and $\beta_2 = 0.9$ is utilized in generator. During the training process, discriminator, classifier, and generator are trained iteratively with 1:1:1 balanced updates in turn. The validation accuracy will be evaluated once for each training epoch. Besides, the batch size of both labeled data and unlabeled data is set as 7, and the number of epochs is set as 150. The loss $\mathcal{R}_{\mathcal{P}}$ is not applied until the number of epochs reaches a threshold when the generator can generate meaningful MR images. The threshold is searched in {60,120} based on the validation performance, and $\alpha_{\mathcal{P}}$ is set as 0.05 empirically.

C. Effect of TT-Core Number

As mentioned in Section III-B, the TT-core number and the TT-rank are two parameters that have a great impact on classification results. This section provides a comparative evaluation of the proposed THS-GAN with respect to a range of TT-core numbers. The GSP block is fixed at the position of "GSP block 3." TT-rank of the classifier and discriminator was fixed at 14 and 6, respectively. Fig. 4 shows that as the TT-core number increased from 3 to 6, the classification accuracy decreased for AD/NC classification. Meanwhile, for AD/MCI and MCI/NC classification, there are no specific trends of accuracy as the core number increased from 3 to 6. But similar to AD/NC classification, the best accuracy is achieved at the minimal core number. This observation is consistent with [23]. Thus we set the TT-core number as 3 in the rest of the experiments.

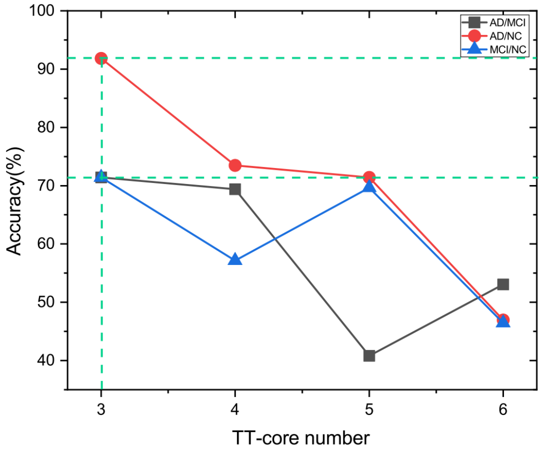


Fig. 4. Comparison of different TT-core numbers.

D. Effect of TT-Rank and GSP Block Position

To investigate the effect of TT-rank and different GSP block position on classification performance, this section provides a comparative evaluation of the proposed THS-GAN with respect to a range of TT-rank values and different GSP block positions for each evaluation group. The TT-core number is fixed as 3. As far as we know, there have been no published studies that adopt tensor-train decomposition in GAN for semisupervised classification. Thus the most suitable TT-rank remains to be explored. Nonetheless, we conducted a variety of preliminary experiments, and have empirically chosen TT-ranks according to the performances in our validation sets. More specifically, we consider the effect of TT-ranks on classification performance when $C\_\text{rank} = \{14, 15, 16, 17, 18, 19, 20\}$ and $D\_\text{rank} = \{6, 7, 8, 9, 10, 11, 12\}$. Note that C_rank and D_rank represent TT-rank of classifier and discriminator, respectively. SS-GAN [12] and triple-GAN [32] are used as two baseline models for comparison purposes. With respect to SS-GAN, the discriminator has three output units corresponding to [CLASS-1, CLASS-2, FAKE]. CLASS-1 and CLASS-2 correspond to one of the classes AD, MCI, NC, respectively, according to the evaluation group. In this case, the discriminator can also act as a classifier. For a fair comparison, the two baselines have the same structure and hyperparameter settings as our model but without tensor-train decomposition and high-order module GSP block.

From Table II, it can be observed that the best AUC can be achieved using $C\_\text{rank} = 20$ and $D\_\text{rank} = 12$ no matter the GSP block is at the position of either GSP block 1, GSP block 2 or GSP block 3 in the context of AD/NC classification. The best AUC of 95.92% is obtained when GSP block 2 is inserted after Dense-BC block 2. On the other hand, in the context of MCI/NC classification, Table III shows that the optimal TT-rank is not consistent with AD/NC classification when GSP block is positioned at different locations. With respect to GSP block 1, a good AUC of 85.71% is obtained when

[3] http://www.tensorflow.org/

TABLE II
COMPARISON OF THS-GAN USING DIFFERENT GSP BLOCK POSITIONS AND TT-RANKS FOR AD/NC CLASSIFICATION

GSP block Position	C_rank	D_rank	#parameters	AUC(%)	Accuracy(%)	Class	precision(%)	recall(%)	f1-score(%)
GSP block 1	14	6	118,210	50.00	56.00	AD	56.00	100	71.79
						NC	0	0	0
	15	7	139,611	50.00	44.00	AD	44.00	100	61.11
						NC	0	0	0
	16	8	163,516	50.00	60.00	AD	60.00	100	75.00
						NC	0	0	0
	17	9	189,925	84.00	84.00	AD	84.00	84.00	84.00
						NC	84.00	84.00	84.00
	18	10	218,838	63.33	77.55	AD	100	26.67	42.11
						NC	75.56	100	86.08
	19	11	250,255	69.64	62.22	AD	100	39.29	56.41
						NC	50.00	100	66.67
	20	*12*	284,176	*91.99*	*92.00*	AD	92.31	92.31	92.31
						NC	91.67	91.67	91.67
GSP block 2	14	6	120,034	83.33	79.59	AD	100	66.67	80.00
						NC	65.52	100	79.17
	15	7	141,435	50.00	48.98	AD	0	0	0
						NC	48.98	100	65.75
	16	8	165,340	93.18	93.88	AD	100	86.36	92.68
						NC	90.00	100	94.74
	17	9	191,749	44.71	55.10	AD	60.47	83.87	70.27
						NC	16.67	5.56	8.34
	18	10	220,662	83.36	83.67	AD	85.71	78.26	81.82
						NC	82.14	88.46	85.18
	19	11	252,079	83.88	83.67	AD	88.89	82.76	85.72
						NC	77.27	85.00	80.95
	20	**12**	**286,000**	**95.92**	**95.92**	AD	**95.83**	**95.83**	**95.83**
						NC	**96.00**	**96.00**	**96.00**
GSP block 3	14	6	121,048	91.81	91.84	AD	91.30	91.30	91.30
						NC	92.31	92.31	92.31
	15	7	142,449	74.48	73.47	AD	83.33	68.97	75.47
						NC	64.00	80.00	71.11
	16	8	166,354	84.32	81.63	AD	95.83	74.19	83.63
						NC	68.00	94.44	79.07
	17	9	192,763	80.77	79.59	AD	69.70	100	82.14
						NC	100	61.54	76.19
	18	10	221,676	73.82	73.47	AD	79.17	70.37	74.51
						NC	68.00	77.27	72.34
	19	11	253,093	69.40	69.39	AD	72.00	69.23	70.59
						NC	66.67	69.57	68.09
	20	*12*	287,014	*92.00*	*91.84*	AD	85.71	100	92.31
						NC	100	84.00	91.30
SS-GAN [12]			251,637	80.02	80.39	AD	82.76	82.76	82.76
						NC	77.27	77.27	77.27
triple-GAN [32]			506,386	86.83	87.76	AD	90.32	90.32	90.32
						NC	83.33	83.33	83.33

C_rank $= 18$ and D_rank $= 10$. Similarly regarding GSP block 2, a good AUC of 88.32% is obtained when C_rank $=$ 15 and D_rank $= 7$. In the same manner, with respect to GSP block 3, a good AUC of 88.72% is obtained when C_rank $= 19$ and D_rank $= 11$. The best AUC of 88.72% is obtained when GSP block 3 is utilized. In the context of AD/MCI classification, Table IV also indicates the same trend that the optimal TT-rank is different when GSP block is positioned at different locations. With respect to GSP block 1, a good AUC of 69.37% is obtained when C_rank $= 14$ and D_rank $= 6$. Similarly regarding GSP block 2, a good AUC of 85.35% is obtained when C_rank $= 17$ and D_rank $= 9$. In the same manner, with respect to GSP block 3, a good AUC of 74% is obtained when C_rank $= 15$ and D_rank $= 7$. The best AUC of 85.35% is obtained when GSP block 2 is utilized.

From Table II to Table IV, the following overall observations can be made.

1) THS-GAN with optimal hyperparameter settings can achieve the best classification performance in terms of AUC and accuracy compared with triple-GAN and SS-GAN. The triple-GAN performs better than the SS-GAN, which confirms that the triple-GAN can alleviate the competing problem of SS-GAN that the discriminator has two incompatible convergence points.
2) Compared with the triple-GAN, THS-GAN can obtain AUC gains of 9.09% (95.92%–86.83%) for AD/NC classification, 15.28% (88.72%–73.44%) for MCI/NC classification, and 13.21% (85.35%–72.14%) for AD/MCI classification, improving the performance by a large margin. This indicates that the performance of the proposed model is significantly improved by introducing tensor-train decomposition and high-order pooling. Furthermore, THS-GAN used far fewer parameters, compared with the triple-GAN which used 506 386 parameters. The compression rates are 506 386/286 000 = 1.77 for AD/NC classification, 506 386/253 093 = 2 for MCI/NC classification, and 506 386/191 749 = 2.64 for AD/MCI classification, respectively.
3) According to our results, the best classification results are obtained by utilizing either GSP block 2 or GSP block 3, but not GSP block 1. This observation indicates that exploiting the second-order statistics in the later layers can improve the predictive power significantly.

TABLE III

COMPARISON OF THS-GAN USING DIFFERENT GSP BLOCK POSITIONS AND TT-RANKS FOR MCI/NC CLASSIFICATION

GSP block Position	C_rank	D_rank	#parameters	AUC(%)	Accuracy(%)	Class	precision(%)	recall(%)	f1-score(%)
GSP block 1	14	6	118,210	70.13	74.07	MCI	69.77	96.77	81.08
						NC	90.91	43.48	58.83
	15	7	139,611	84.89	85.19	MCI	81.25	92.86	86.67
						NC	90.91	76.92	83.33
	16	8	163,516	64.94	59.26	MCI	48.72	90.48	63.34
						NC	86.67	39.39	54.16
	17	9	189,925	79.94	81.48	MCI	84.21	69.57	76.19
						NC	80.00	90.32	84.85
	18	*10*	218,838	*85.71*	*85.19*	MCI	100	71.43	83.33
						NC	76.47	100	86.67
	19	11	250,255	71.56	70.37	MCI	62.16	92.00	74.19
						NC	88.24	51.72	65.22
	20	12	284,176	66.48	61.11	MCI	51.22	95.45	66.67
						NC	92.31	37.50	53.33
GSP block 2	14	6	120,034	65.74	62.50	MCI	54.55	96.00	69.57
						NC	91.67	35.48	51.16
	15	*7*	141,435	*88.32*	*87.50*	MCI	96.15	80.65	87.72
						NC	80.00	96.00	87.27
	16	8	165,340	52.23	53.57	MCI	57.14	14.81	23.52
						NC	53.06	89.66	66.67
	17	9	191,749	70.18	69.64	MCI	63.89	85.19	73.02
						NC	80.00	55.17	65.30
	18	10	220,662	63.87	64.29	MCI	67.74	67.74	67.74
						NC	60.00	60.00	60.00
	19	11	252,079	76.87	76.79	MCI	83.87	76.47	80.00
						NC	68.00	77.27	72.34
	20	12	286,000	81.25	82.14	MCI	81.82	75.00	78.26
						NC	82.35	87.50	84.85
GSP block 3	14	6	121,048	70.75	71.43	MCI	66.67	89.66	76.47
						NC	82.35	51.85	63.63
	15	7	142,449	72.22	73.21	MCI	65.91	100	79.45
						NC	100	44.44	61.53
	16	8	166,354	67.69	67.86	MCI	70.00	70.00	70.00
						NC	65.38	65.38	65.38
	17	9	192,763	77.60	76.79	MCI	68.97	83.33	75.47
						NC	85.19	71.88	77.97
	18	10	221,676	80.14	80.36	MCI	78.13	86.21	81.97
						NC	83.33	74.07	78.43
	19	**11**	**253,093**	**88.72**	**89.29**	MCI	**85.29**	**96.67**	**90.62**
						NC	**95.45**	**80.77**	**87.5**
	20	12	287,014	69.74	71.43	MCI	66.67	93.33	77.78
						NC	85.71	46.15	60.00
SS-GAN [12]			251,637	71.15	69.64	MCI	61.54	92.31	73.85
						NC	88.24	50.00	63.83
triple-GAN [32]			506,386	73.44	71.43	MCI	61.76	87.50	72.41
						NC	86.36	59.38	70.37

The conjectured reason for this is that the features extracted in the earlier layers are simple and common, but in the later layers representative features will be abstracted, and by inserting the high-order pooling module GSP block in the later layers, more discriminative features can be enhanced and redundant features will be suppressed; thus the predictive performance is improved. Although inserting GSP block at the later layers will increase the number of parameters, the best trade-off between accuracy and number of parameters should be chosen at GSP block 2. GSP block 2 arrangement leads to the best accuracy with the optimal TT-ranks.

4) TT-rank has a significant effect on testing accuracy, and the optimal value of TT-rank depends on network architecture and data. It is difficult to specify an optimal value for TT-rank in advance. Again, this observation is consistent with [23] that finding optimal TT-rank remains a challenge.

According to the experimental results, the optimal value of TT-rank lies in the range [14, 20] for classifier and [6, 12] for discriminator. It is not time-consuming to find it in practical applications. Under optimal TT-ranks, THS-GAN can achieve better performance than triple-GAN and our model uses fewer parameters, which indicates that TT-decomposition can utilize parameters more efficiently, and is less likely to converge to local minima. Note that the optimal hyperparameter settings for each evaluation group will be utilized in the rest of the experiments.

E. Effect of the Amount of Labeled Data

In this subsection, the effect of the number of labeled data for semisupervised classification is investigated. For the proposed THS-GAN, the architecture and hyperparameters are set as the optimal settings found in Section IV-D. The 3D-DenseNet architecture is the same as the classifier of THS-GAN but without tensor-train decomposition and GSP block. Similarly, the structure of SS-GAN is also the same as THS-GAN but without tensor-train decomposition and GSP block. It can be seen from Fig. 5 that as the number of labeled data increased, our THS-GAN outperforms SS-GAN by a large margin and performs better than 3D-DenseNet when there are less labeled data for AD/MCI classification. Fig. 6 shows that as the number of labeled data increased, THS-GAN always

TABLE IV

COMPARISON OF THS-GAN USING DIFFERENT GSP BLOCK POSITIONS AND TT-RANKS FOR AD/MCI CLASSIFICATION

GSP block Position	C_rank	D_rank	#parameters	AUC(%)	Accuracy(%)	Class	precision(%)	recall(%)	f1-score(%)
GSP block 1	14	6	118,210	69.37	68.89	AD	62.50	90.91	74.07
						MCI	84.62	47.83	61.12
	15	7	139,611	50.00	46.94	AD	0	0	0
						MCI	46.94	100	63.89
	16	8	163,516	59.08	59.18	AD	59.09	54.17	56.52
						MCI	59.26	64.00	61.54
	17	9	189,925	63.83	64.44	AD	80.00	36.36	50.00
						MCI	60.00	91.30	72.41
	18	10	218,838	59.45	61.22	AD	70.00	60.43	64.86
						MCI	58.97	88.46	70.77
	19	11	250,255	55.18	55.10	AD	52.00	56.52	54.17
						MCI	58.33	53.85	56.00
	20	*12*	284,176	*69.00*	*71.11*	AD	76.92	50.00	60.61
						MCI	68.75	88.00	77.19
GSP block 2	14	6	120,034	63.68	67.35	AD	60.00	47.37	52.94
						MCI	70.59	80.00	75.00
	15	7	141,435	48.71	53.06	AD	38.46	25.00	30.30
						MCI	58.33	72.41	64.61
	16	8	165,340	70.65	69.39	AD	86.67	50.00	63.42
						MCI	61.76	91.30	73.68
	17	**9**	**191,749**	**85.35**	**85.71**	AD	**85.71**	**88.89**	**87.27**
						MCI	**85.71**	**81.82**	**83.72**
	18	10	220,662	61.45	61.22	AD	56.00	63.64	59.58
						MCI	66.67	59.26	62.75
	19	11	252,079	57.74	63.27	AD	80.00	19.05	30.77
						MCI	61.36	96.43	75.00
	20	12	286,000	45.26	48.98	AD	33.33	25.00	28.57
						MCI	55.88	65.52	60.32
GSP block 3	14	6	121,048	63.44	71.43	AD	70.73	93.55	80.56
						MCI	75.00	33.33	46.15
	15	*7*	142,449	*74.00*	*73.47*	AD	80.95	65.38	72.34
						MCI	67.86	82.61	74.51
	16	8	166,354	72.07	71.43	AD	80.00	61.54	69.57
						MCI	65.52	82.61	73.08
	17	9	192,763	59.47	55.10	AD	75.00	40.00	52.17
						MCI	45.45	78.95	57.69
	18	10	221,676	49.56	57.14	AD	37.50	15.79	22.22
						MCI	60.98	83.33	70.42
	19	11	253,093	69.05	71.43	AD	71.00	86.00	77.78
						MCI	73.00	52.00	60.74
	20	12	287,014	70.37	67.35	AD	100	41.00	58.16
						MCI	58.00	100	73.42
SS-GAN [12]			251,637	50.00	48.98	AD	48.98	100	65.75
						MCI	0	0	0
triple-GAN [32]			506,386	72.14	73.47	AD	76.47	59.09	66.67
						MCI	71.88	85.19	77.97

outperforms both 3D-DenseNet and SS-GAN for MCI/NC classification. The same trend can be found in Fig. 7. We can also observe that the THS-GAN requires fewer labeled samples to achieve comparable results. In Fig. 5, when the number of labeled data is small such as 300, THS-GAN can still achieve better performance than SS-GAN and 3D-DenseNet which use more labeled data such as 330, 360, 390, and 420, respectively, in the context of AD/MCI classification. Similar trends can also be found for MCI/NC and AD/NC in Figs. 6 and 7, respectively. This improvement is beneficial from real unlabeled MR images and the synthetic MR images produced by the generator.

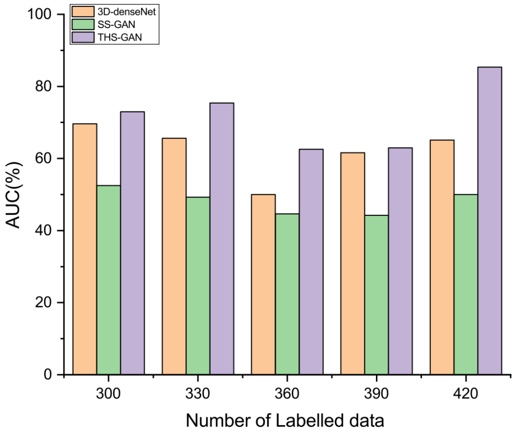


Fig. 5. Comparison of different number of labeled data for AD/MCI classification.

F. Effect of Number of Parameters

In this section, we investigate the properties of THS-GAN and compare it with triple-GAN uncompressed for AD/MCI classification. In order to compare the performance for the same range of parameters, various TT-ranks are utilized for THS-GAN. The result in Fig. 8 illustrates that THS-GAN can obtain the best AUC with optimal TT-ranks when the number of parameters is compressed in the range $[10^5, 2\times10^5]$ (in red dashed circle). Furthermore, THS-GAN can achieve comparable AUC when TT-ranks are set to large numbers, and the number of parameters is in the range of $[2 \times 10^5, 3\times10^5]$ or $[3\times10^5, 4\times10^5]$. Overall speaking, THS-GAN can achieve much better performance when TT-rank is not large,

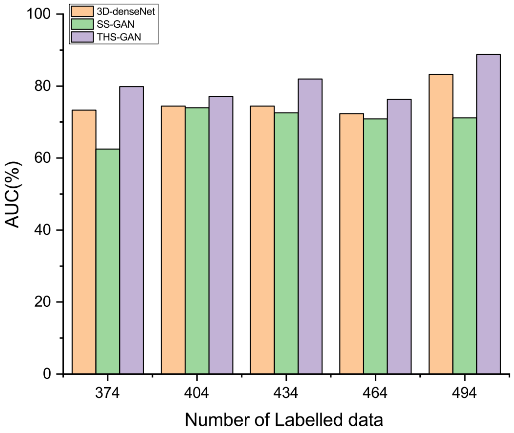


Fig. 6. Comparison of different numbers of labeled data for MCI/NC classification.

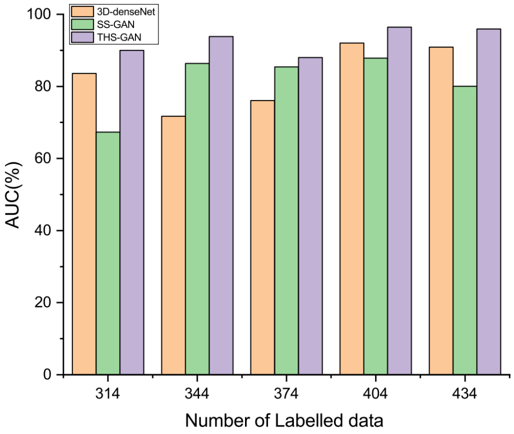


Fig. 7. Comparison of different numbers of labeled data for AD/NC classification.

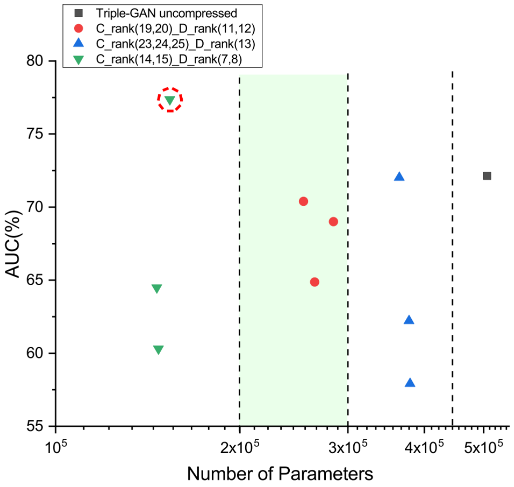


Fig. 8. Comparison of different numbers of parameters for AD/MCI classification.

and the number of the parameter is compressed between 10^5 and 2×10^5.

G. Convergence Comparison

Figs. 9 and 10 show the convergence curves of the proposed THS-GAN and SS-GAN for evaluation group MCI/NC and AD/MCI, respectively. THS-GAN converges faster than conventional SS-GAN. In the case of AD/MCI classification, SS-GAN cannot converge since the differences between AD and MCI are so subtle that the MR images of AD, MCI, and fake are hard to be distinguished by the discriminator. These results are also consistent with Tables III and IV in Section IV-D. More specifically, for MCI/NC classification, AUC of THS-GAN (88.72%) is much higher than SS-GAN (71.15%) since THS-GAN converges faster than SS-GAN. For AD/MCI classification, the AUC of SS-GAN is only 50%, and SS-GAN cannot converge during the training process.

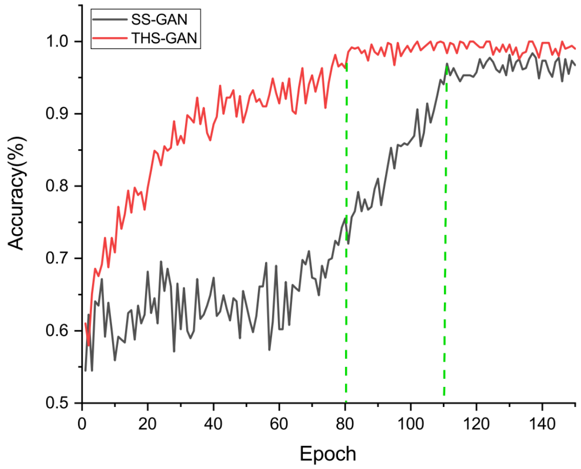


Fig. 9. Convergence curves for MCI/NC classification.

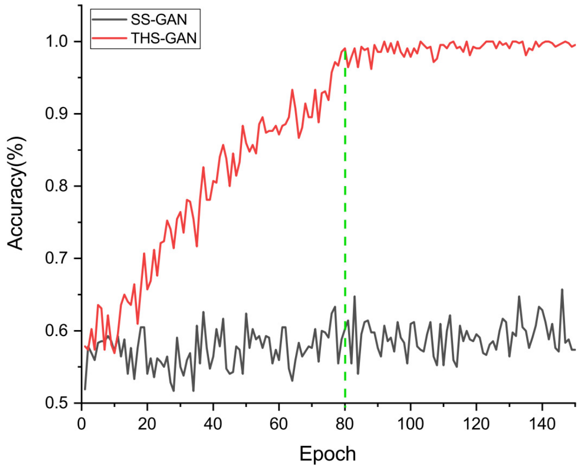
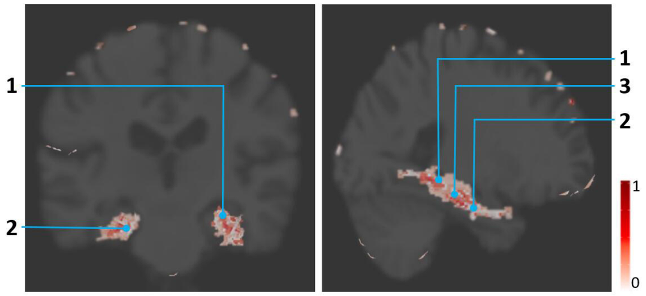


Fig. 10. Convergence curves for AD/MCI classification.



(a) (b)

Fig. 11. Visualization of AD-related regions recognized by THS-GAN. The number in the figure denotes specific brain region (1 Hippocampus, 2 Entorhinal cortex, and 3 Parahippocampal cortex). (a) Coronal view. (b) Sagittal view.

TABLE V
COMPARISON WITH EXISTING METHODS

Model	Classification Method	MCI vs NC(%)			AD vs MCI(%)			AD vs NC(%)		
		ACC	Recall	AUC	ACC	Recall	AUC	ACC	Recall	AUC
Plocharski et al. [37]	Feature Selection+SVM	84.40	82.30	84.00	81.50	81.70	83.00	92.30	91.30	98.00
Peng et al. [38]	Feature Selection+SVM	71.60	83.90	-	65.40	41.20	-	88.40	84.10	-
Xu et al. [39]	Feature Selection+SVM	70.89	61.39	79.02	-	-	-	90.40	92.36	95.36
Neffati et al. [40]	Feature Selection+SVM	-	-	-	-	-	-	91.11	85.00	-
Li et al. [41]	3D-DenseNet+RNN	75.00	81.90	75.80	-	-	-	89.10	84.60	91.00
Cui et al. [42]	3D-CNN+RNN	-	-	-	-	-	-	91.33	86.87	93.22
Ren et al. [43]	2D-CNN	88.50	82.16	82.00	85.32	78.79	80.00	93.75	94.23	93.00
Liu et al. [44]	CNN	77.84	76.81	82.72	-	-	-	84.97	82.65	90.63
Cheng et al. [45]	3D-CNN	82.33	80.65	81.66	79.38	82.24	78.55	87.13	86.31	91.24
Our method	THS-GAN	**89.29**	**96.67**	**88.72**	**85.71**	**88.89**	**85.35**	**95.92**	**95.83**	95.92

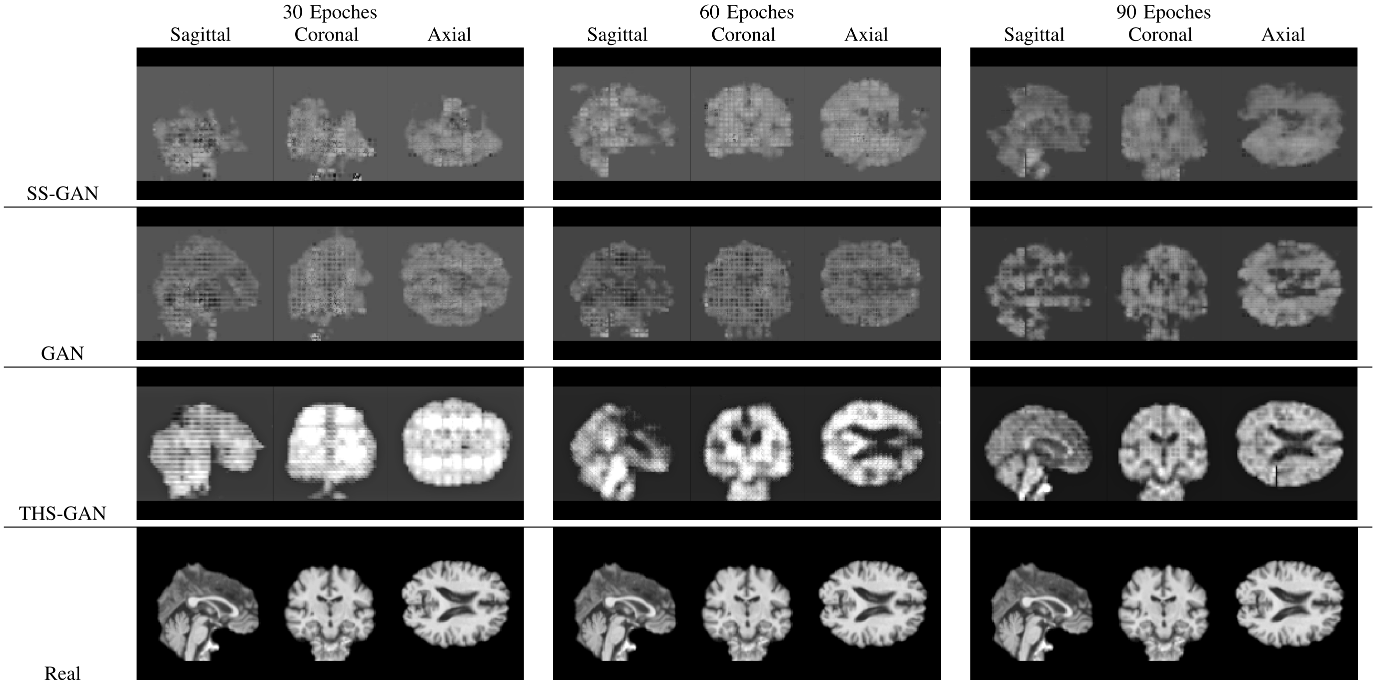


Fig. 12. Comparison of brain MRI slices generated by SS-GAN, GAN, and THS-GAN.

On the other hand, THS-GAN converges faster and an AUC of 85.35% can be achieved.

H. Visualization of AD-Related Regions Recognized by THS-GAN

It is significant to identify the relevant biomarkers for AD diagnosis. The biomarker can quantitatively measure the neurodegeneration and brain atrophy in MR images. Various biomarkers in MR images have been discovered, such as cortical thickness, volumetric decline, shape changes, and annualized rates of atrophy in specific brain regions. In particular, the annual atrophy rate of the hippocampal for MCI and AD patients is much higher than that for healthy elderly subjects. The annual atrophy of the hippocampus for healthy elderly subjects is 1.6% to 1.7% per year, while the annual atrophy of the hippocampus for MCI patients is 2.8% per year, and for AD patients is 3.5% to 4% per year.

In this work, the AD-related regions recognized by the proposed THS-GAN are shown in Fig. 11. It can be observed that the lesions recognized by THS-GAN focus on the hippocampus, entorhinal cortex, and parahippocampal cortex. It has already been validated by the previous studies [33] that these recognized regions are discriminative for AD diagnosis, meanwhile, the hippocampus and entorhinal cortex are significant for identifying biomarkers in clinical practice. Although the volume loss and shape changes of these recognized regions cannot be quantitatively measured in this work, they are beneficial for identifying the biomarkers in future work. Based on these recognized regions, the existing biomarkers such as brain boundary shift integral (BBSI) [34], scoring by nonlocal image patch estimator (SNIPE) [35], and other grading biomarkers [36] can be computed. Furthermore, new potential biomarkers might be discovered based on these recognized regions in future work.

I. Visualization of Generated Images

In this section, we visualize the center-cut slices of the generated MR images from random latent vectors during the training process as shown in Fig. 12. In the beginning, generated samples are blurry, and the detailed features of the brain disappear. In the latter stage, compared with SS-GAN and GAN, the generated samples from the proposed THS-GAN can reflect more detailed attributes of the brain (e.g., sulci, gyri).

J. Comparison With Existing Methods

Several machine learning methods have been proposed for AD diagnosis using MR images. Table V shows the comparison results with the existing methods. Except classification method of [40] used MR images from the OASIS data set, the other eight methods all used the ADNI data set. Although 3D-CNN in [45] did not release the source code, the network structure of the model is described in detail in their article. Thus we reimplemented the model by TensorFlow according to their paper. The other eight papers neither released the source code nor provided a detailed description of the model. Therefore the experimental results reported in these papers are referred directly for comparison in Table V. It can be seen that the proposed THS-GAN model achieves the best classification performance with ACC of 89.29% for MCI versus NC, 85.71% for AD versus MCI, and 95.92% for AD versus NC. Meanwhile, the best recall is also obtained by the proposed THS-GAN. More specifically, compared with the machine learning methods based on feature selection and support vector machine (SVM) [37]–[40], the proposed THS-GAN not only achieves better classification performance by a large margin but also requires less image-preprocessing steps for model training. No segmentation and rigid registration are required for feature extraction in the proposed THS-GAN. Moreover, THS-GAN also outperforms the existing deep learning models such as 2D-CNN [43], 3D-CNN [44], [45] and the hybrid network combining CNN and recurrent neural networks (RNNs) [41], [42]. It demonstrates the benefit of tensor-train decomposition and the high-order pooling module leveraged in THS-GAN. Furthermore, THS-GAN achieves superior classification performance, indicating its potential capability of assessing MCI and AD.

V. Conclusion

In this article, we developed a novel THS-GAN for assessing MCI and AD. The three-player cooperative game-based framework is tensorized so that THS-GAN can benefit from the structural information of the brain. By introducing high-order pooling in THS-GAN, more significant features can be extracted by making full use of the second-order statistics of the holistic MR images. To the best of our knowledge, THS-GAN is the first work to consider tensor-train decomposition in GAN and leverage GAN for semisupervised classification on MR images for AD diagnosis. The experimental results demonstrate that the proposed THS-GAN model can obtain promising results. We will focus on identifying the relevant biomarkers in future work.

References

[1] B. D. James, S. E. Leurgans, L. E. Hebert, P. A. Scherr, K. Yaffe, and D. A. Bennett, "Contribution of Alzheimer disease to mortality in the United States," *Neurology*, vol. 82, no. 12, pp. 1045–1050, Mar. 2014.

[2] C. Patterson, *World Alzheimer Report 2018: The State of the Art of Dementia Research: New Frontiers*. London, U.K.: Alzheimer's Disease International (ADI), 2018.

[3] J. Zhang, Y. Gao, Y. Gao, B. C. Munsell, and D. Shen, "Detecting anatomical landmarks for fast Alzheimer's disease diagnosis," *IEEE Trans. Med. Imag.*, vol. 35, no. 12, pp. 2524–2533, Dec. 2016.

[4] S. Liu *et al.*, "Multimodal neuroimaging feature learning for multiclass diagnosis of Alzheimer's disease," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 4, pp. 1132–1140, Apr. 2015.

[5] B. Lei, P. Yang, T. Wang, S. Chen, and D. Ni, "Relational-regularized discriminative sparse learning for Alzheimer's disease diagnosis," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1102–1113, Apr. 2017.

[6] H. Dai, S. Jiang, and Y. Li, "Atrial activity extraction from single lead ECG recordings: Evaluation of two novel methods," *Comput. Biol. Med.*, vol. 43, no. 3, pp. 176–183, Mar. 2013.

[7] S. Wang *et al.*, "Skeletal maturity recognition using a fully automated system with convolutional neural networks," *IEEE Access*, vol. 6, pp. 29979–29993, 2018.

[8] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *Med Image Anal*, vol. 58, pp. 101–119, Dec. 2019.

[9] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[10] T. Salimans *et al.*, "Improved techniques for training GANs," in *Proc. NIPS*, vol. 2016, pp. 2234–2242.

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[12] A. Odena, "Semi-supervised learning with generative adversarial networks," 2016, *arXiv:1606.01583*. [Online]. Available: http://arxiv.org/abs/1606.01583

[13] J. C. Baron *et al.*, "*In vivo* mapping of gray matter loss with voxel-based morphometry in mild Alzheimer's disease," *NeuroImage*, vol. 14, no. 2, pp. 298–309, Aug. 2001.

[14] Y. Fan, D. Shen, R. C. Gur, R. E. Gur, and C. Davatzikos, "COMPARE: Classification of morphological patterns using adaptive regional elements," *IEEE Trans. Med. Imag.*, vol. 26, no. 1, pp. 93–105, Jan. 2007.

[15] M. Liu, D. Zhang, and D. Shen, "Hierarchical fusion of features and classifier decisions for Alzheimer's disease diagnosis," *Hum. Brain Mapping*, vol. 35, no. 4, pp. 1305–1319, Apr. 2014.

[16] Y. Pan, M. Liu, C. Lian, T. Zhou, Y. Xia, and D. Shen, "Synthesizing missing pet from MRI with cycle-consistent generative adversarial networks for Alzheimer's disease diagnosis," in *Proc. MICCAI*, 2018, pp. 455–463.

[17] K. Armanious, C. Jiang, S. Abdulatif, T. Kustner, S. Gatidis, and B. Yang, "Unsupervised medical image translation using cycle-MedGAN," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2019, pp. 1–5.

[18] J. Islam and Y. Zhang, "Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks," *Brain Informat.*, vol. 5, no. 2, pp. 1–14, Dec. 2018.

[19] S. Wang, H. Wang, Y. Shen, and X. Wang, "Automatic recognition of mild cognitive impairment and Alzheimer's disease using ensemble based 3D densely connected convolutional networks," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 517–523.

[20] Y. Wang *et al.*, "3D auto-context-based locality adaptive multi-modality GANs for PET synthesis," *IEEE Trans. Med. Imag.*, vol. 38, no. 6, pp. 1328–1339, Jun. 2019.

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[22] D. Gu, "3D densely connected convolutional network for the recognition of human shopping actions," M.S. thesis, Univ. Ottawa, Ottawa, ON, Canada, 2017. [Online]. Available: https://ruor.uottawa.ca/handle/10393/36739, doi: 10.20381/ruor-21013.

[23] A. Novikov, D. Podoprikhin, A. Osokin, and D. Vetrov, "Tensorizing neural networks," in *Proc. NIPS*, 2015, pp. 442–450.

[24] T. Garipov, D. Podoprikhin, A. Novikov, and D. Vetrov, "Ultimate tensorization: Compressing convolutional and FC layers alike," 2016, *arXiv:1611.03214*. [Online]. Available: http://arxiv.org/abs/1611.03214

[25] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, Jan. 2011.

[26] H. Huang and H. Yu, "LTNN: A layerwise tensorized compression of multilayer neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1497–1511, May 2018.

[27] X. Cao, X. Zhao, and Q. Zhao, "Tensorizing generative adversarial nets," in *Proc. IEEE Int. Conf. Consum. Electron. Asia (ICCE-Asia)*, Jun. 2018, pp. 206–212.

[28] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3024–3033.

[29] J. M. Rondina *et al.*, “Selecting the most relevant brain regions to discriminate Alzheimer’s disease patients from healthy controls using multiple kernel learning: A comparison across functional and structural imaging modalities and atlases,” *NeuroImage, Clin.*, vol. 17, pp. 628–641, Jan. 2018.

[30] M. Jenkinson and S. Smith, “A global optimisation method for robust affine registration of brain images,” *Med. Image Anal.*, vol. 5, no. 2, pp. 143–156, Jun. 2001.

[31] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in *Proc. PMLR*, May 2013, pp. 1139–1147.

[32] C. Li, T. Xu, J. Zhu, and B. Zhang, “Triple generative adversarial nets,” in *Proc. NIPS*, 2017, pp. 4088–4098.

[33] F. Mrquez and M. A. Yassa, “Neuroimaging biomarkers for Alzheimer’s disease,” *Mol. Neurodegeneration*, vol. 14, no. 1, p. 21, Jun. 2019.

[34] N. C. Fox and P. A. Freeborough, “Brain atrophy progression measured from registered serial MRI: Validation and application to Alzheimer’s disease,” *J. Magn. Reson. Imag.*, vol. 7, no. 6, pp. 1069–1075, Nov. 1997.

[35] P. Coup, S. Eskildsen, J. Manjon, V. Fonov, and L. Collins, “Simultaneous segmentation and grading of anatomical structures for patient’s classification: Application to Alzheimer’s disease,” *NeuroImage*, vol. 59, pp. 3736–3747, Nov. 2011.

[36] T. Tong *et al.*, “A novel grading biomarker for the prediction of conversion from mild cognitive impairment to Alzheimer’s disease,” *IEEE Trans. Biomed. Eng.*, vol. 64, no. 1, pp. 155–165, Jan. 2017.

[37] M. Plocharski and L. R. Østergaard, “Sulcal and cortical features for classification of Alzheimer’s disease and mild cognitive impairment,” in *Image Analysis*. Cham, Switzerland: Springer, 2019, pp. 427–438.

[38] J. Peng, X. Zhu, Y. Wang, L. An, and D. Shen, “Structured sparsity regularized multiple kernel learning for Alzheimer’s disease diagnosis,” *Pattern Recognit.*, vol. 88, pp. 370–382, Apr. 2019.

[39] L. Xu, Z. Yao, J. Li, C. Lv, H. Zhang, and B. Hu, “Sparse feature learning with label information for Alzheimer’s disease classification based on magnetic resonance imaging,” *IEEE Access*, vol. 7, pp. 26157–26167, 2019.

[40] S. Neffati, K. B. Abdellafou, I. Jaffel, O. Taouali, and K. Bouzrara, “An improved machine learning technique based on downsized KPCA for Alzheimer’s disease classification,” *Int. J. Imag. Syst. Technol.*, vol. 29, no. 2, pp. 121–131, Jun. 2019.

[41] F. Li and M. Liu, “A hybrid convolutional and recurrent neural network for hippocampus analysis in Alzheimer’s disease,” *J. Neurosci. Methods*, vol. 323, pp. 108–118, Jul. 2019.

[42] R. Cui and M. Liu, “RNN-based longitudinal analysis for diagnosis of Alzheimer’s disease,” *Computerized Med. Imag. Graph.*, vol. 73, pp. 1–10, Apr. 2019.

[43] F. Ren *et al.*, “Exploiting discriminative regions of brain slices based on 2D CNNs for Alzheimer’s disease classification,” *IEEE Access*, vol. 7, pp. 181423–181433, 2019.

[44] M. Liu, D. Cheng, K. Wang, and Y. Wang, “Multi-modality cascaded convolutional neural networks for Alzheimer’s disease diagnosis,” *Neuroinformatics*, vol. 16, nos. 3–4, pp. 295–308, Oct. 2018.

[45] D. Cheng, M. Liu, J. Fu, and Y. Wang, “Classification of MR brain images by combination of multi-CNNs for AD diagnosis,” *Proc. SPIE*, vol. 10420, Jul. 2017, Art. no. 1042042.

Wen Yu received the Ph.D. degree from the Department of Computer Science, University of Liverpool, Liverpool, U.K., in 2015.

She worked as a Senior Software Engineer with HSBC from 2016 to 2018. She held a Post-Doctoral Fellowship at the Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Science, Shenzhen, China, from 2018 to 2021. Her current research interests include deep learning and computer vision.

Baiying Lei (Senior Member, IEEE) received the M.Eng. degree in electronics science and technology from Zhejiang University, Hangzhou, China, in 2007, and the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2013.

She is currently with the Health Science Center, School of Biomedical Engineering, Shenzhen University, Shenzhen, China. She has coauthored more than 180 scientific articles, e.g., *Medical Image Analysis*, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON MEDICAL IMAGING, and the IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING. Her current research interests include medical image analysis, machine learning, and pattern recognition.

Dr. Lei serves as an Associate Editor for the IEEE TRANSACTIONS ON MEDICAL IMAGING and an Editorial Board Member for *Medical Image Analysis*, *Neural Computing and Application*, *Frontiers in Neuroinformatics*, *Frontiers in Aging Neuroscience*? *Scientific Reports*, and *PLOS One*.

Michael K. Ng (Senior Member, IEEE) received the B.Sc. and M.Phil. degrees from The University of Hong Kong, Hong Kong, in 1990 and 1992, respectively, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 1995.

He is a Chair Professor with the Department of Mathematics, The University of Hong Kong. His research interests include data science, imaging science, and scientific computing.

Dr. Ng serves on the editorial boards of international journals.

Albert C. Cheung (Member, IEEE) received the B.S. degree in physics from Stanford University, Stanford, CA, USA, in 1965, and the M.A. and Ph.D. degrees in physics from UC Berkeley, Berkeley, CA, USA, in 1966 and 1976, respectively.

He did his post-graduate work at UC Berkeley and was an Associate Professor in astrophysics at UC Davis, Davis, CA, USA. He is currently with the School of Engineering, The Hong Kong University of Science and Technology, Hong Kong, and the Mathematics Department, Hong Kong Baptist University, Hong Kong. He was the Director and the Head of the Laboratory Center, City University of Hong Kong, Hong Kong. His research interests are microwave astrophysics, signal and multidimensional series analysis, big data applications in finance and engineering, and medical imaging.

Yanyan Shen received the Ph.D. degree from the Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Hong Kong, in 2012.

From 2013 to 2014, she was a Post-Doctoral Research Fellow at the School of Information and Communication Engineering, Inha University, Incheon, South Korea. She is currently an Associate Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. Her current research interests include optimization methods and machine learning in wireless networks.

Shuqiang Wang (Member, IEEE) received the Ph.D. degree in system engineering and engineering management from the City University of Hong Kong, Hong Kong, in 2012.

He was a Research Scientist with Huawei Technologies Noah’s Ark Lab. He held a Post-Doctoral Fellowship at The University of Hong Kong, Hong Kong, from 2013 to 2014. He is currently a Professor with the Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Science, Shenzhen, India. His current research interests include machine learning, medical image computing, and optimization theory.