3D Brain Generation using Auto-Encoding Generative Adversarial Networks with Cycle Consistent Embedding

by

Shibo Xing

Submitted to the Graduate Faculty of

the Department of Computer Science in partial fulfillment

of the requirements for the degree of

Master of Sciences

University of Pittsburgh

2021

UNIVERSITY OF PITTSBURGH

DEPARTMENT OF COMPUTER SCIENCE

This thesis was presented

by

Shibo Xing

It was defended on

December 14 2021

and approved by

Thesis Advisor: Seong Jae Hwang, Department of Computer Science

Adriana Kovashka, Department of Computer Science

Davneet Minhas, Department of Radiology

3D Brain Generation using Auto-Encoding Generative Adversarial Networks with Cycle Consistent Embedding

Shibo Xing, M.S.

University of Pittsburgh, 2021

An array of generative adversarial networks (GANs) have been accomplishing the realistic generation of full 3D brain images. This largely follows a common procedure of sampling from a latent space prior $\mathcal{Z}$ (i.e., random vectors) and mapping it to realistic images in $\mathcal{X}$ (e.g., 3D brains), but a naïve implementation also comes with the ubiquitous mode collapse issue. This challenge has recently been addressed by strongly imposing certain characteristics, such as Gaussianness, to the prior by also explicitly mapping $\mathcal{X}$ to $\mathcal{Z}$ via encoder. This Auto-Encoder type GANs, however, fail to accurately map 3D brain images to the desirable prior, which the generator assumes to be sampling the random vectors from. While Variational Auto-Encoding GAN (VAE-GAN) handles this mode collapse issue by explicitly imposing Gaussianness, this also causes blurriness in images. In this thesis, we demonstrate how our cycle consistent embedding GAN (CCE-GAN) is able to solve both the mode collapse and blurriness issues by accurately encoding 3D MRIs to the standard normal prior while maintaining the image generation quality. Using our trained novel model with T1 MRI brain images from Alzheimer's Disease Neuroimaging Initiative (ADNI) and FLAIR tumor MRI brain images from Brain Tumor Segmentation (BraTS) datasets, we will show how an improved prior $\mathcal{Z}$ space can lead to an output distribution free of mode collapse and of high image quality. We also quantitatively and qualitatively assess the embeddings to reaffirm the importance of embedding in GAN for 3D brain generation.

Table of Contents

List of Tables

List of Figures

1.0 Introduction

As the challenges in medical imaging research have been increasingly benefiting from various deep learning solutions, researchers have begun searching for additional opportunities to quantitatively enhance or automate clinical diagnosis and other pathological analyses. Convolution networks have been continuously demonstrating their success in semantic segmentation of the brain image, which is a necessary step preceding various quantitative analyses [7, 12, 2]. Other tasks such as feature representation learning which used to rely heavily on expert handcrafting can also be guided with deep learning [14]. However, as deep learning training often requires high volume dataset to ensure the efficacy, the insufficient supply of MRI datasets, especially of the underrepresented disease populations, has led to limited real-life applications and reduced reproducibility of the correspondingly trained models [11, 1, 15]. An alternative method to suffice the need for MRI brain image datasets is to generate them through *Generative Adversarial Network* (GAN), a particularly effective model that aims to reproduce the target distribution. Given a good convergence of a dataset distribution, GAN model is able to provide infinite number of samples.

There have been some GAN-based solutions in the field for 2D brain image generation, such as the PET image generation network [5], or various-weight MR brain generation network [4]. Generating 3D MRI images using GAN however, still significantly lacks solutions due to a range of technical difficulties facing this type of network. *Mode collapse* for example, an issue that prevents the model output from converging to a full approximation of image distribution, will be heightened in 3D brain generation due to the complexity of brain structural information [9]. In order to address these problems, researchers have turned their attention to Auto-Encoder network, an encoding-decoding model that can be combined with GAN to regularize its adversarial training [8, 6].

More traditionally, Variational Auto-Encoding GAN (VAE-GAN) [6] embeds the latent space $\mathcal{Z}$ by learning two parameter vectors that represent the mean and log variance. This definition of latent space confines it into a Gaussian distribution. Albeit achieving a near-perfect latent space convergence using its Encoder, VAE-GAN outputs severely blurry images

and lack fine details which often render them visually less pleasing.

Recently, Kwon et al. have proposed 3D-α-WGAN [8] which is known to be the first to map Gaussian noise to 3D MRI brain image distribution by utilizing the α-GAN to enhance the latent space embedding. It uses a Code Discriminator network that plays an adversarial game with the Encoder. Essentially, this model consists of two components of adversarial learning, first in the latent embedding which will be then used as a prior for the latter output image space. Unfortunately, through experiments, we have found out that this attempt to foster the Encoder's output distribution to be closer to the standard normal is unsuccessful. The complexity provided by the Code Discriminator network doesn't outperform some of the more simplistic methods of latent space convergence. As a result, 3D-α-WGAN's Generator's ability is necessarily hindered by its latent space divergence.

In response to the aforementioned caveats, our method focuses on the latent space embedding, as we have found that this Generator prior of Auto-Encoder largely regulates the force of the convergence of the entire GAN. A properly defined and approximated latent embedding is critical to both 3D brain image quality and its distribution coverage. In our solution, we replaced 3D-α-WGAN model's Code Discriminator with a more effective Wasserstein loss which regularizes the latent space embedding (WAE-GAN), then added Cycle Consistent loss terms which further enhanced output image quality (CCE-GAN). After that, we used comparable metrics to validate the improvements on latent space embedding as well as on image distribution approximation.

2.0 Method

In this section, we describe our method in details. First, we briefly describe two prior methods, VAE-GAN and 3D-α-WGAN, which are crucial building blocks of our method. We also characterize their weaknesses and demonstrate how our method, CCE-GAN, technically overcome these downfalls.

2.1 Prior Methods

The latent space encapsulates all encoded information of the $\mathcal{X}$ domain, and thus the convergence of it is critical to the Decoder's generation ability. Auto-Encoders controls the output domain by mapping $\mathcal{Z}$ to a specific prior distribution, preferably standard Gaussian. We used two Auto-Encoding models as our baselines. One is the traditional VAE-GAN [6], and the second is 3D-α-WGAN [8]. Although both models can achieve a 3D MRI Brain output distribution, their ability to generate 3D brains are affected by the limitation on latent space.

2.1.1 VAE-GAN

VAE is known to be an effective solution against mode collapse, which is an ubiquitous problem in GAN's training. Mode Collapse is categorized as the Generator producing a distribution of the similar looking images, also known as being in the same mode. It happens when the Discriminator is stuck in a local minimum. VAE essentially prevents the Generator from using just one mode that can "fool" the Discriminator. In the VAE's loss function, the Kullback–Leibler divergence models the latent prior into a Gaussian multivariate distribution. Then the $\mathcal{X}$-space reconstruction loss will ensure that Generator maps each encoded $\mathcal{P}_E(z \mid \mathcal{X}^i)$ to a reconstructed $\mathcal{X}^i$.

However, VAE-GAN's exemption from against mode collapse doesn't equate to high-

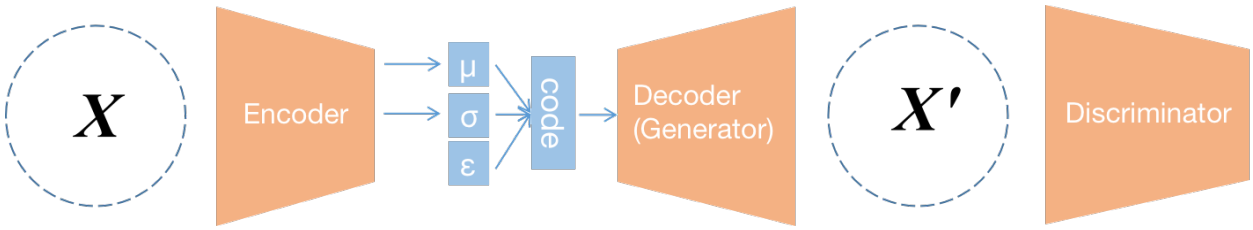


Figure 1: VAE-GAN structure

quality output images. Its output images usually suffer from blurriness. This is primarily due to the limited representation of its latent space which cannot encapsulates all the structural information of $\mathcal{X}$. As shown by Figure 1, VAE encodes the $\mathcal{Z}$ domain into only two parameters, a mean vector and the log variance vector which is then to be translated into standard deviation vector. Using VAE's reparametrization trick that maintains the z vector, or "code", as a deterministic node learnable from back propagation, we can construct it as $\mu + \sigma \cdot \epsilon$ where $\epsilon \in \mathcal{N}(0, I)$.

The two learned variables μ and σ could create highly similar $\mathcal{P}_E(z \mid \mathcal{X}^i)$ distribution encoding for different i. Thus, it is very difficult for the Generator to approximate $\mathcal{P}_G(\mathcal{X}' \mid \mathcal{P}_E(z \mid \mathcal{X}^i))$ given drastically different $\mathcal{X}^i$. Naturally, the best approach Generator learns to counter the reconstruction loss between $\mathcal{X}'$ and $\mathcal{X}$ is to apply blurriness in the output images. As we are trying to re-create a 3D dataset through GAN, the exponentially higher degree of structural variation of $\mathcal{X}$ would makes the Generator produce even more blurriness compared to that of a 2D dataset.

2.1.2 3D-α-WGAN

A more recent approach called 3D-α-WGAN [8] that adopts the structure of α-WGAN is proven to generate qualitatively sound 3D brain images. Its novelty mainly consists of using Wasserstein gradient penalty for the Discriminator's training and using α-GAN's Code Discriminator for latent space convergence [13].

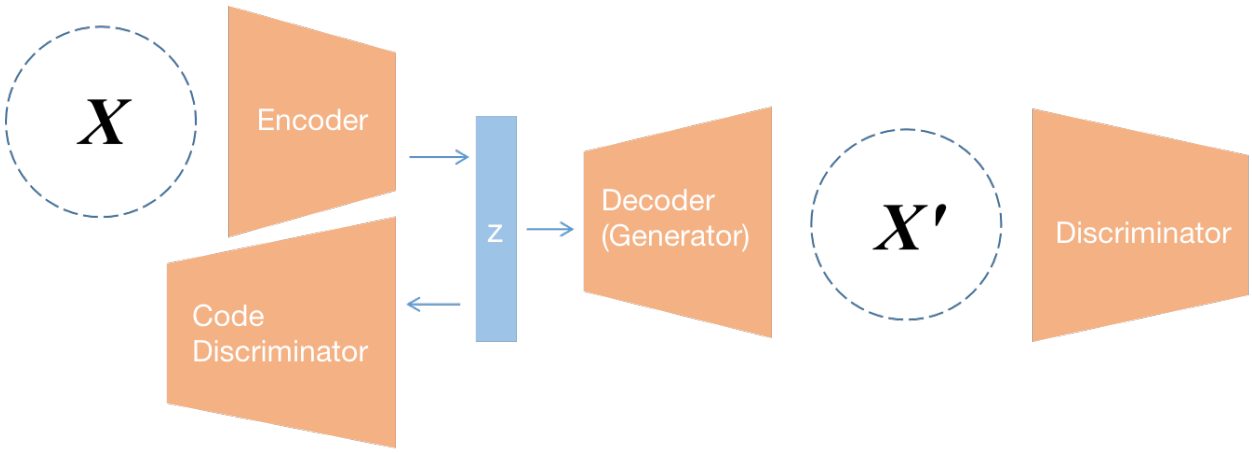


Figure 2: 3D-α-WGAN structure

As shown by Figure 2, 3D-α-WGAN has a much expanded latent vector dimension compared to VAE-GAN. Its Encoder's output domain is a latent space with a certain dimension (e.g., 1000 dimensional vector). By learning the latent distribution directly, the Encoder's output distribution has more variance and can produce a more accurate mapping from $\mathcal{X}$ to $\mathcal{Z}$.

However, the convergence of such a latent space relies on α-GAN's Code Discriminator, which we found to be ineffective in 3D brain generation[13]. The Code Discriminator primarily consists of large linear layers, which if combined together with the Encoder forms a GAN for latent space. We realized through experiments that Code Discriminator diverges with gradient explosion, which elevates the Encoder's output scale significantly. The removal of log loss of α-GAN in 3D-α-WGAN and the up-scaling effect of linear layers contribute to the loss divergence of Code Discriminator[8, 13]. Albeit the $\mathcal{X}$ domain is free of mode collapse, the divergence in the latent space is necessarily limited by a poor $\mathcal{Z} \longrightarrow \mathcal{X}$ mapping. In our approach, we will show that a good convergence in $\mathcal{Z}$ could enhance $\mathcal{X}$ domain both quantitatively and qualitatively.

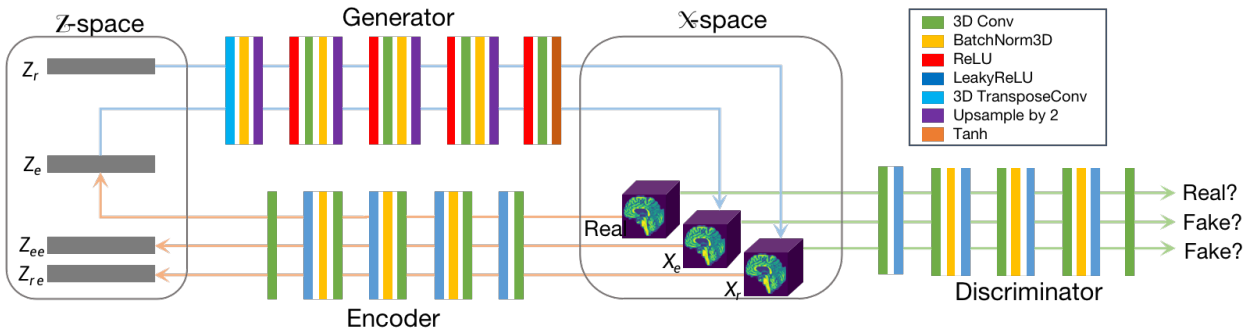


Figure 3: CCE-GAN

2.2 CCE-GAN

We now describe our model shown in Figure 3. Our main goal aligns with the most Auto-Encoder's objective function, that is to learn a mapping from random Gaussian to the image space, $\mathcal{X}$. Our solution involves two structural modifications building on top of 3D-α-WGAN. With the Bayes model of Auto-Encoder signified as $\mathcal{P}_G(\mathcal{X}' \mid z)\mathcal{P}_E(z \mid \mathcal{X})$, we first focus on improving the Encoder prior portion.

2.2.1 Wasserstein Distance

The training of an Auto-Encoder undoubtedly aims to generate diverse samples of $\mathcal{X}$ with high quality. We will show that having an Encoder which excels in the latent space convergence is crucial to the Generator's performance. We inherited 3D-α-WGAN's latent space definition as a 1000 dimensional vector as it is a reasonable domain to compress $\mathcal{X}$ structural information. The key change of our model is facilitating the Encoder's ability to map to Gaussian before it can learn to map $\mathcal{X}$ to various modes in $\mathcal{Z}$.

Wasserstein distance, also known as the optimal transport distance, is a highly effective metric to measure how far apart two probabilities μ and ν are. For discrete probabilities, the expected value of Wasserstein distance signifies the minimum transport effort to move the

probability mass from one distribution to another. This is calculated as summing the product of the mass transferred and distance of every pair in the coupling γ, which minimizes the cost among all of such. In our case, the distance could be an integral over the joint probability density of the two distributions times p-norm of their p-norm distance.

$$\mathcal{W}_p(\mu, \nu) = \left(\inf_{\gamma \in Y(\mu,\nu)} \int ||x - y||^p d\gamma(x, y) \right)^{1/p}. \tag{1}$$

With Wasserstein loss, our Encoder can quickly converge $\mathcal{P}_E(z \mid \mathcal{X})$ to a standard normal. This guarantees that $\mathcal{Z}_{rand}$, Decoder's input vector sampled from $\mathcal{N}(0, 1)$ can more readily represent the modes in the latent space.

We replaced the Code Discriminator in 3D-α-WGAN with a Wasserstein Optimal Transport distance, while keeping WGAN structure intact. This would give us the objective function:

$$\underset{G,E}{\operatorname{argmin}} - \mathbb{E}_{z_e}[D(X_e)] - \mathbb{E}_{z_r}[D(X_r)] + \lambda_2 ||X_r - X_e||_1 + \lambda_3 \mathcal{W}_l(z_r, z_e). \tag{2}$$

2.2.1.1 Encoder's separation trick

Traditionally, Auto-Encoder is optimized with a single loss function. The loss on the output domain $\mathcal{X}$ is to be back-propagated through both the Decoder and the Encoder. What we found in our experiments made us believe that it would be beneficial to have the Encoder learn the mapping from $\mathcal{X}$ to Gaussian in advance of learning to encode $\mathcal{X}$ into latent vectors of $\mathcal{Z}$. By reaching this intermediate goal, the learning of the Auto-Encoder in general can be based on a standard normal latent space at a very early stage. Moreover, due to the intricacy of the Auto-Encoder GAN loss function, a Wasserstein loss term may not contribute to the optimization as much as expected when combined with other loss terms. Therefore, we isolated the Wasserstein metric term and back-propagated it to Encoder in a separate optimization step.

2.2.2 Cycle Consistent Loss

Similar to VAE-GAN, our model also uses reconstruction loss to enhance the image translation ability. We add an L1 loss between $\mathcal{X}$ and $\mathcal{P}_G(\mathcal{P}_E(\mathcal{X}))$ to the Auto-Encoder, and two additional L2 losses derived from two samples of the Encoder output, $\mathcal{P}_E(\mathcal{P}_G(\mathcal{Z}))$ and $\mathcal{P}_E(\mathcal{P}_G(\mathcal{P}_E(\mathcal{X})))$ The L1 and L2 loss terms combined will form the forward and backward cycle consistency as described in CycleGAN. However, our L2 losses are calculated between latent vectors and Gaussian vectors since we are more inclined to improve the latent space using the cycle consistency. This modification of the network imposes a regularization for both the Generator and the Discriminator networks. The resulting enhancements of translation ability on the Auto-Encoder benefit the output image quality $\mathcal{X}'$, as shown by our quantitative results. Combining all of the above formulations, we have the following objective functions for the model, where $x \in \mathcal{X}$, $z_r \in \mathcal{Z}$:

$$\underset{D}{\operatorname{argmin}}\ \mathbb{E}_x[D(G(E(x)))] + \mathbb{E}_{z_r}[D(G(z_r))] - 2\mathbb{E}_x[D(x)] + \lambda_1 L_{gp}(D) \tag{3}$$

$$\begin{aligned} \underset{G,E}{\operatorname{argmin}} - \mathbb{E}_z[D(G(E(x)))] - \mathbb{E}_x[D(G(z_r))] + \lambda_2||G(z_r) - G(E(x))||_1 \\ + \lambda_3||E(G(z_r)) - z_r||_2 + \lambda_4||E(G(E(X))) - z_r||_2 \end{aligned} \tag{4}$$

$$\underset{E}{\operatorname{argmin}}\ \lambda_5 \mathcal{W}_l(z_r, E(x)) \tag{5}$$

The Wasserstein distance is a prerequisite for cycle consistent loss to be in effect because of its regularization power. Without a controlled latent space, L2 loss alone will likely incur an unpredictable scale of gradients which then would be back-propagated to our Auto-Encoder. Our complete training procedure is described as follows:

Algorithm 1 CCE-GAN Algorithm

Require:

D: the initial Discriminator's weight

G: the initial Generator's weight

E: the initial Encoder's weight

$\mathcal{X}$: the real image dataset

$\mathcal{Z}$: the 1000-dimension Gaussian distribution

Training:

for $i = 1, \ldots, 100000$ **do**

  Sample $x \sim \mathcal{X}$, batch from image dataset

  Sample $z_r \sim \mathcal{Z}$, batch from standard Gaussian

  // **Auto-Encoder Update**

  for 1 updates **do**

    Update G, E using Equation (4)

  end for

  // **Encoder Update**

  for 1 updates **do**

    Update E using Equation (5)

  end for

  // **Discriminator Update**

  for 3 updates **do**

    Update D using Equation (3)

  end for

end for

Generation:

Sample $z_r \sim \mathcal{Z}$

Generated Image $\leftarrow G(z_r)$

3.0 Dataset and Experiments

3.1 Dataset

We have trained our models on healthy and tumored brain datasets. For healthy datasets, we used 991 T1-weighted Control Normal brain images from Alzheimer's Disease Neuroimaging Initiative dataset(ADNI, adni.loni.usc.edu). For the tumored datasets, we used 210 Fluid-attenuated inversion recovery (FLAIR) tumored images from Brain Tumor Segmentation (BraTS) Challenge (surfer.nmr.mgh.harvard.edu).

3.1.0.1 Data Processing

The ADNI dataset is processed by the 'recon-all' procedure of Freesurfer pre-download, which involves skull stripping, intensity normalization, spatial normalization and other essential brain preprocessing steps. Then we removed all 2-dimensional planes with only zero-intensity pixels on all three axes (Axial, Coronal, Sagittal). During the training, the data-loader further performs a data augmentation procedure which involves flipping images and adding random noise pixels to each image. Eventually, each image is resized into 64×64×64 dimension to fit inside our GPU memory.

3.2 Experimental Setup

Our final model CCE-GAN has three benchmark models which are our intermediate model WAE-GAN described in section 2.2.1, VAE-GAN [10] and 3D-α-WGAN [8]. We trained the first three models for 40000 iterations. And as to 3D-α-WGAN, we trained it for 100000 iterations for the sake of epoch fairness. Each model is trained on a NVIDIA GeForce RTX 2080 Ti GPU with 11GB memory. For CCE-GAN and WAE-GAN, we used a mini-batch of size 4, ADAM optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, G_lr=0.0002, D_lr=0.0002,

E_lr=0.00010), with hyper-parameters $\lambda_1 = 1$, $\lambda_2 = 100$, $\lambda_3 = 50$, $\lambda_4 = 50$, $\lambda_5 = 1000$. We set the Discriminator's optimization step to 3 per generator's optimization step. Both our learning rate and the optimization step settings of the Discriminator are an attempt to push the Discriminator's convergence ahead of Generator and therefore reduce the likelihood of mode collapse. We also isolated the Wasserstein distance back-propagation to maximize its effect on Encoder.

3.3 Evaluation

3.3.1 Quantitative Results

We first quantitatively assess the generation results as shown in Table 1 and Table 2. For measuring the output image quality, we used linear kernel and Radial-Basis Function (RBF) kernel in a Maximum Mean Discrepancy (MMD) two-sample test [3]. For both ADNI and BraTS, we iterated the the entire dataset, pairing every sample with an output in $\mathcal{P}_G(\mathcal{X}|\mathcal{Z})$ and calculated their average of 10 MMD scores. We performed the same testing procedure in the $\mathcal{Z}$ space as well, pairing each $\mathcal{P}_E(\mathcal{Z}|\mathcal{X})$ with a standard normal vector. Another quantitative metric we used is the Multiscale Structural Similarity Index (MS-SSIM), which measures luminance, contrast and structural similarity between a pair of samples. We samples 1000 pair from each model's output and compared the score to that of ADNI dataset (0.8426) and BraTS dataset (0.7441).

VAE-GAN's images' blurriness effect is not captured by the linear MMD kernel and resulted the best score. Other than that, CCE-GAN performed the best with the MMD metric in the ADNI $\mathcal{X}$ space. CCE-GAN also achieves the best score in $\mathcal{Z}$ space with linear-kernel MMD and is second to only VAE-GAN with a slight difference with RBF kernel. Same holds true for BraTS dataset's $\mathcal{Z}$ space. In the $\mathcal{X}$-space of BraTS dataset, 3D-α-GAN performs slightly better than our model for both MMD kernel. In the ADNI $\mathcal{X}$ space SSIM score, WAE-GAN achieves the best result as it is the closest to the SSIM score of the ADNI dataset, for BraTS dataset CCE-GAN achieves the best SSIM score.

	$\mathcal{X}$-space			$\mathcal{Z}$-space	
Model	**Linear**	**RBF**	**SSIM**	**Linear**	**RBF**
3D-α-WGAN	735.1	0.751	0.8321	2741973.8	3.026
VAE-GAN	**359.6**	1.148	0.9706	249.0	**0.422**
WAE-GAN	719.5	0.770	**0.8471**	286.1	0.475
CCE-GAN	**585.5**	**0.747**	0.8732	**237.3**	0.424

Table 1: ADNI **Linear** and **RBF** MMD, and **SSIM**.

	$\mathcal{X}$-space			$\mathcal{Z}$-space	
Model	**Linear**	**RBF**	**SSIM**	**Linear**	**RBF**
3D-α-WGAN	**2280.5**	**1.018**	0.7674	725448.0	3.181
VAE-GAN	**1507.3**	1.905	0.7579	272.10	**0.465**
WAE-GAN	2567.9	1.032	0.9392	369.8	0.625
CCE-GAN	2339.9	1.026	**0.7305**	**237.6**	0.483

Table 2: BraTS **Linear** and **RBF** MMD, and **SSIM**.

3.3.2 Qualitative Results

We use a combination of T-Distributed Stochastic Neighbor Embedding (TSNE) and Principal Component Analysis (PCA) to qualitatively measure the degree of mode collapse of each model. For each of our trained model and the target distribution, we would collect 512 samples from both the $\mathcal{X}$ space and $\mathcal{Z}$ space, then perform a T-SNE to reduce their dimensionality to 50 then PCA to 2. In the resulting 2-dimensional plot we can evaluate how close each model's convergence is to the target distribution. Both types of samples of all four models' outputs are shown in Figure 8 and 9.

Figure 4: ADNI TSNE-PCA Plots. $\mathcal{Z}$-space ($N = 150$)

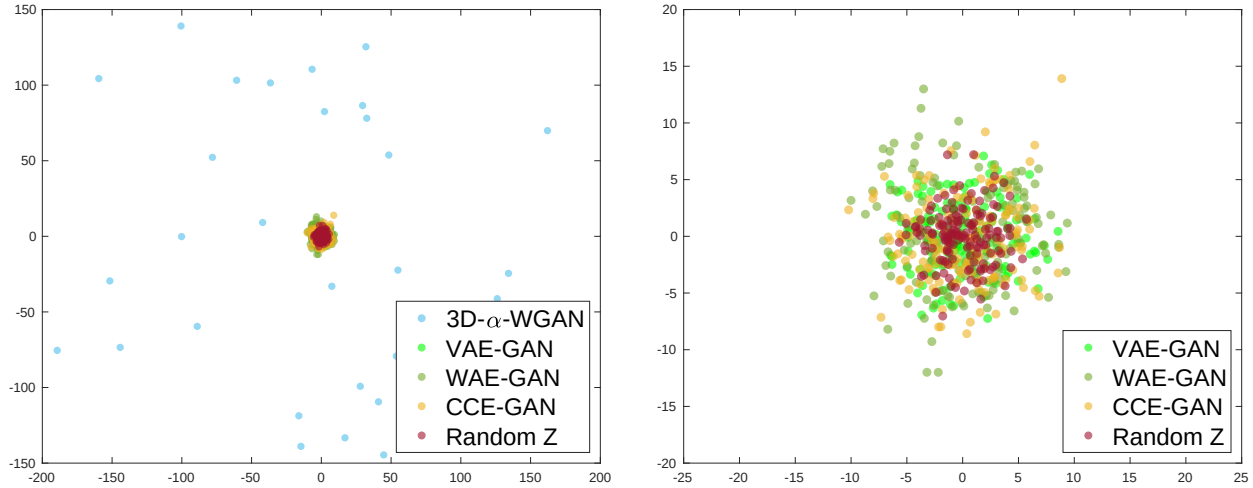
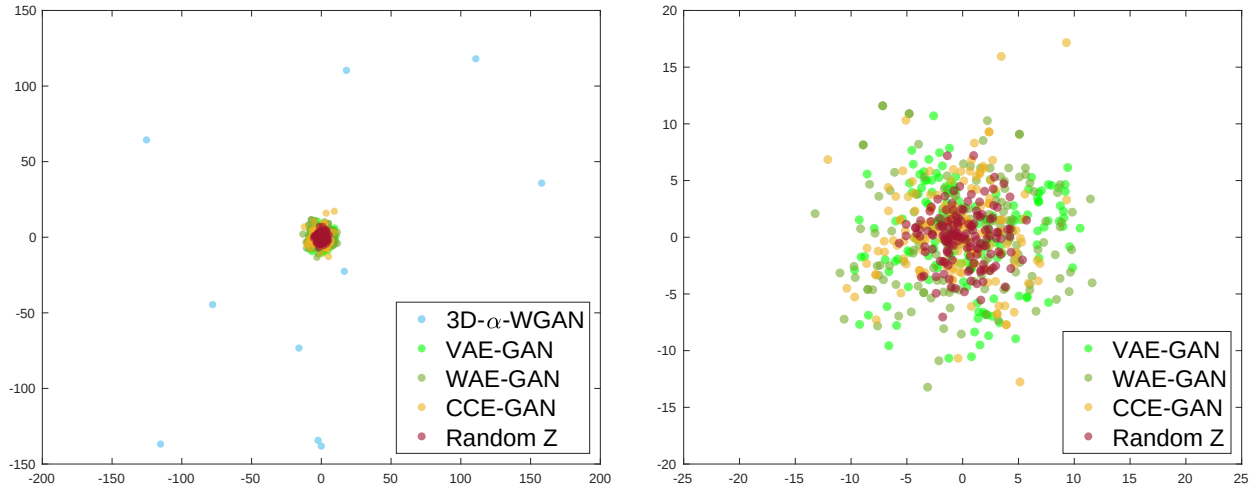


Figure 5: BraTS TSNE-PCA Plots. $\mathcal{Z}$-space ($N = 150$)



Based on the $\mathcal{Z}$ space TSNE-PCA plot in Figure 4, we can evaluate all the model Encoder's convergence to standard normal distribution. We can tell 3D-α-WGAN's Encoder produces an extremely sparse distribution compared to Gaussian. Once we zoom in into the center cluster, we can see that our two models, along with VAE-GAN can closely cover the Gaussian distribution. The same results re-appear in the evaluation of all models' Encoders trained on BraTS dataset.

Figure 6: ADNI TSNE-PCA Plots. $\mathcal{X}$-space ($N = 500$)

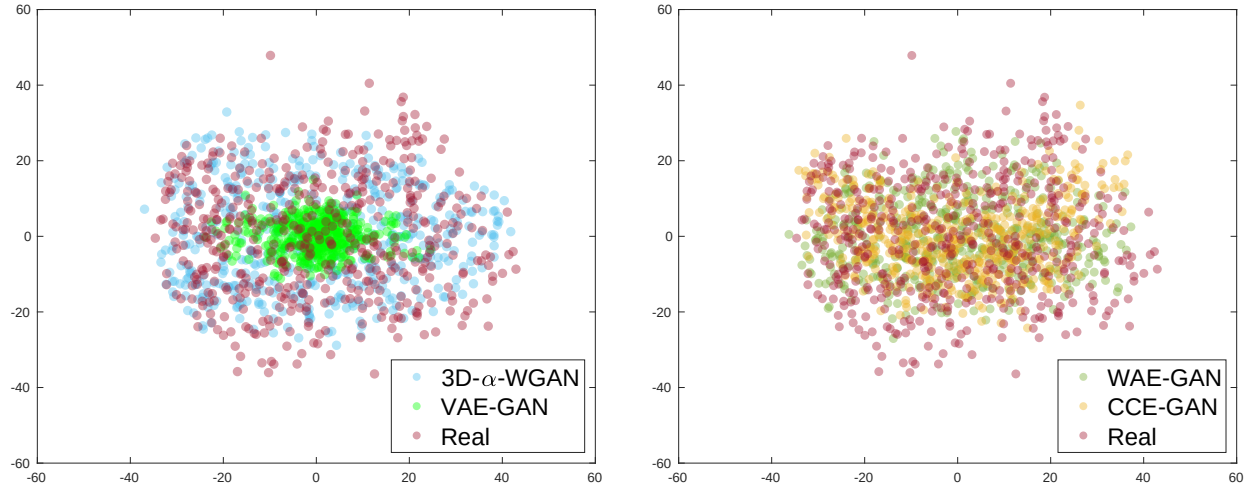
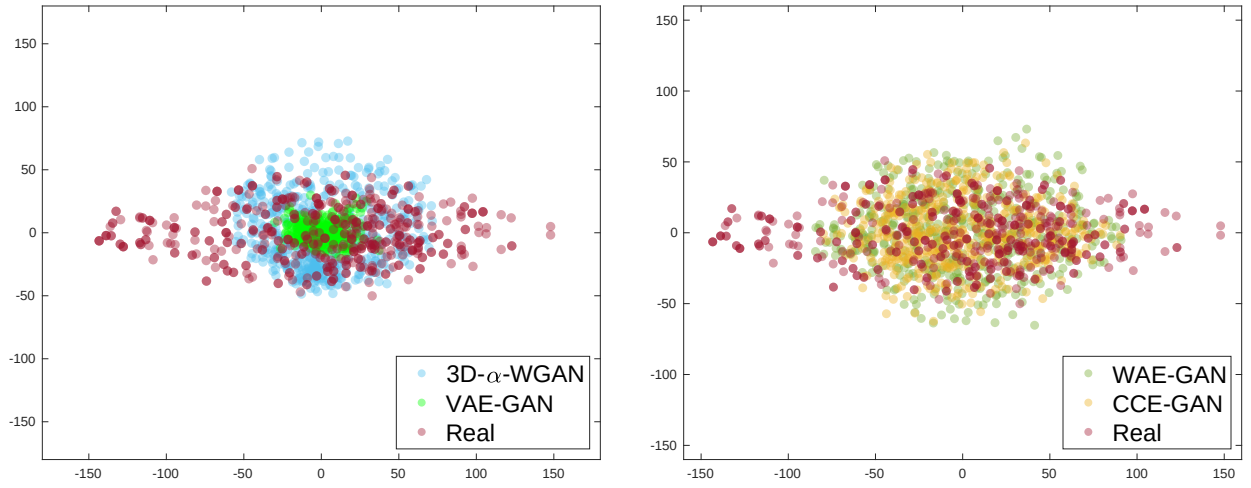
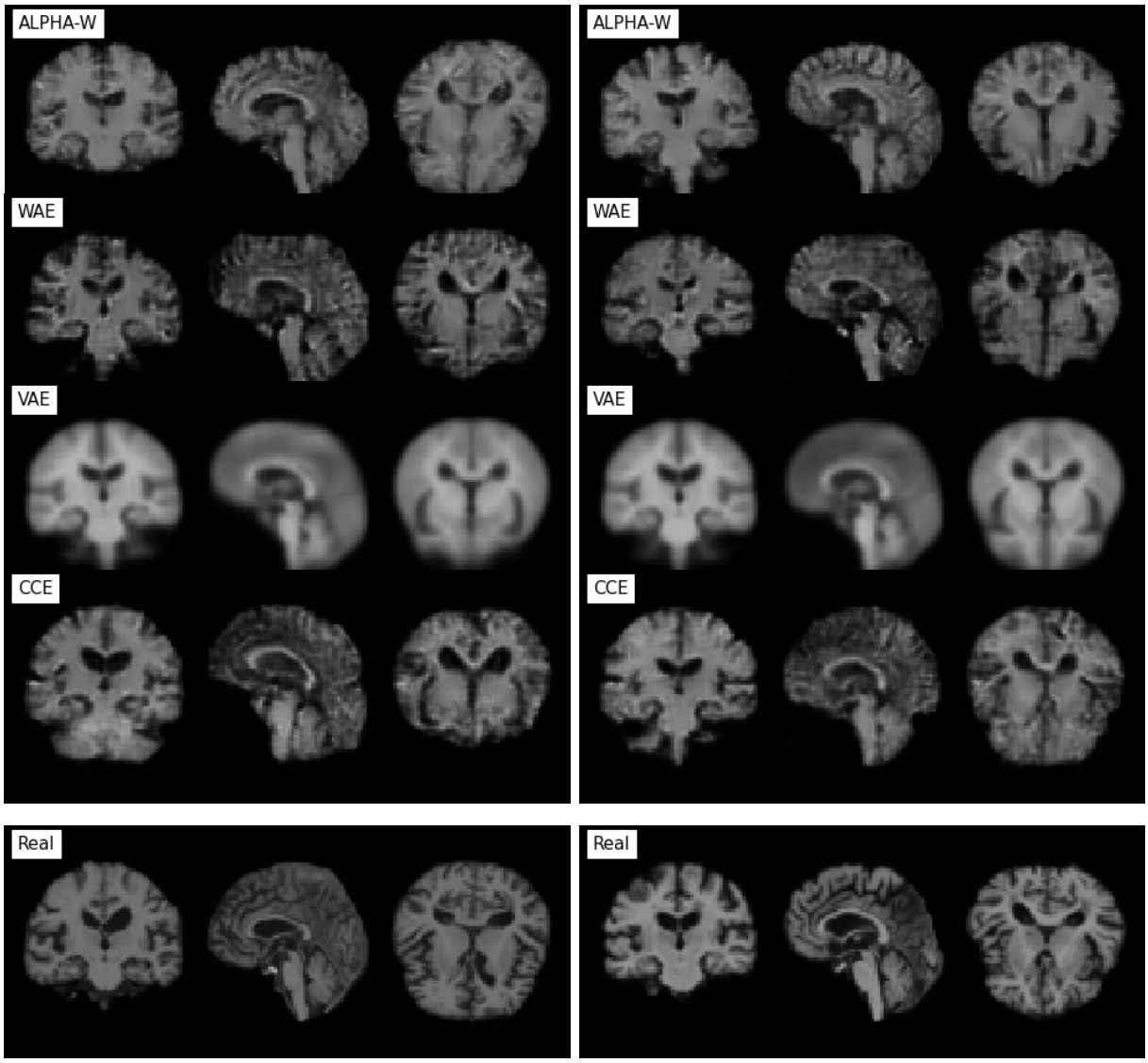


Figure 7: BraTS TSNE-PCA Plots. $\mathcal{X}$-space ($N = 500$)



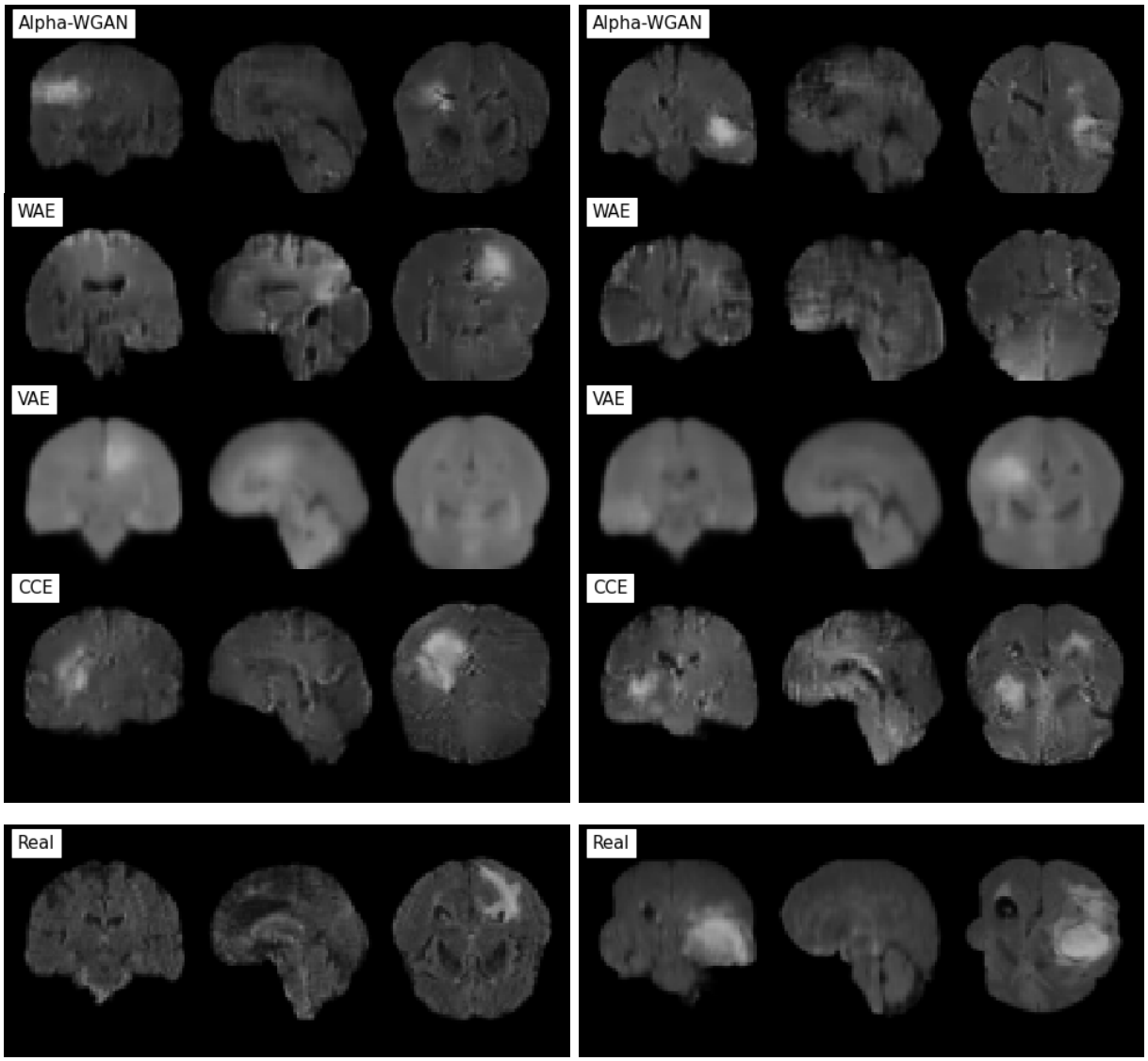
For ADNI-trained Generator's TSNE-PCA, we can see that 3D-α-WGAN and WAE-GAN have the most superior coverage to the dataset. VAE-GAN's blurriness causes mode to less distinguishable and thus results in a more clustered $\mathcal{X}$-space distribution. CCE-GAN's coverage to the ADNI dataset is slightly limited by the Cycle Consistent loss, but is still free of mode collapse. In the TSNE-PCA plot of the BraTS dataset evaluation, we can tell

Figure 8: ADNI brains samples. Note that all generated and real images are independently generated or sampled. Thus, the generated images cannot be directly compared to the real images in their corresponding columns.



that there are some dominant features distort the distribution to a non-Gaussian shape. 3D-α-WGAN's and clustered VAE-GAN Generators remain producing Gaussian distributions which lack some coverage to the true $\mathcal{X}$-space distribution. WAE-GAN and CCE-GAN, on the other hand, converge to the dataset better.

Figure 9: BraTS brains samples. Note that all generated and real images are independently generated or sampled. Thus, the generated images cannot be directly compared to the real images in their corresponding columns.



4.0 Conclusions

We have presented an improved solution to better replicate a 3D brain dataset. Using our method, the latent space have achieved a closer convergence to the standard Gaussian Generator's prior and thus led to better image quality and approximation of the target dataset distribution. We realize there are still room for training optimization and further development. 3D-α-WGAN's superior ability to counter mode collapse, as shown by the TSNE-PCA plot, implicates an opportunity to slightly relaxed the Wasserstein loss in exchange for more modes coverage. Regarding future works, we plan to focus more on the characteristic datasets generating that are scarce in quantities. We are looking to use arbitrary conditioning on the latent space or neural attention type method to achieve the goal.

Bibliography

[1] Darwin Castillo, Vasudevan Lakshminarayanan, and Maria J Rodriguez-Alvarez. Mri images, brain lesions and deep learning. *arXiv preprint arXiv:2101.05091*, 2021.

[2] Shaoguo Cui, Lei Mao, Jingfeng Jiang, Chang Liu, and Shuyu Xiong. Automatic semantic segmentation of brain gliomas from mri images using a deep cascaded neural network. *Journal of healthcare engineering*, 2018, 2018.

[3] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[4] Changhee Han, Hideaki Hayashi, Leonardo Rundo, Ryosuke Araki, Wataru Shimoda, Shinichi Muramatsu, Yujiro Furukawa, Giancarlo Mauri, and Hideki Nakayama. Gan-based synthetic brain mr image generation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 734–738. IEEE, 2018.

[5] Jyoti Islam and Yanqing Zhang. Gan-based synthetic brain pet image generation. *Brain informatics*, 7(1):1–12, 2020.

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[7] Jens Kleesiek, Gregor Urban, Alexander Hubert, Daniel Schwarz, Klaus Maier-Hein, Martin Bendszus, and Armin Biller. Deep mri brain extraction: A 3d convolutional neural network for skull stripping. *NeuroImage*, 129:460–469, 2016.

[8] Gihyun Kwon, Chihye Han, and Dae-shik Kim. Generation of 3d brain mri using auto-encoding generative adversarial networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 118–126. Springer, 2019.

[9] Sayeri Lala, Maha Shady, Anastasiya Belyaeva, and Molei Liu. Evaluation of mode collapse in generative adversarial networks. *High Performance Extreme Computing, IEEE*, 10, 2018.

[10] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.

[11] Mahmoud Mostapha and Martin Styner. Role of deep learning in infant brain mri analysis. *Magnetic resonance imaging*, 64:171–189, 2019.

[12] Yao Qin, Konstantinos Kamnitsas, Siddharth Ancha, Jay Nanavati, Garrison Cottrell, Antonio Criminisi, and Aditya Nori. Autofocus layer for semantic segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 603–611. Springer, 2018.

[13] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

[14] Guorong Wu, Minjeong Kim, Qian Wang, Yaozong Gao, Shu Liao, and Dinggang Shen. Unsupervised deep feature learning for deformable registration of mr brain images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 649–656. Springer, 2013.

[15] Ekin Yagis, Selamawet Workalemahu Atnafu, Alba García Seco de Herrera, Chiara Marzi, Riccardo Scheda, Marco Giannelli, Carlo Tessa, Luca Citi, and Stefano Diciotti. Effect of data leakage in brain mri classification using 2d convolutional neural networks. *Scientific reports*, 11(1):1–13, 2021.