Spring 2021

Perceptually Improved Medical Image Translations Using Conditional Generative Adversarial Networks

Anurag Vaidya
ajv012@bucknell.edu

Perceptually Improved Medical Image Translations Using Conditional Generative Adversarial Networks

By

Anurag J. Vaidya

A Thesis Submitted to the Honors Council

For Honors in The Department of Computer Science

April 1st, 2021

Approved by:

Advisor: _____

Dr. Joshua Stough, Department of Computer Science

Co-advisor: _____

Dr. Aalpen Patel, Geisinger Radiology

Department head: _____

Dr. Dan Cavanagh, Department of Biomedical Engineering

External reader: _____

Dr. Benjamin Wheatley, Department of Mechanical Enginee

Table of Contents

List of Tables

List of Figures

Abstract

Magnetic resonance imaging (MRI) can help visualize various brain regions. Typical MRI sequences consists of T1-weighted sequence (favorable for observing large brain structures), T2-weighted sequence (useful for pathology), and T2-FLAIR scan (useful for pathology with suppression of signal from water). While these different scans provide complementary information, acquiring them leads to acquisition times of ~1 hour and average cost of $2,600, presenting significant barriers. To reduce these costs associated with brain MRIs, we present *pTransGAN*, a generative adversarial network capable of translating both healthy and unhealthy T1 scans into T2 scans. We show that the addition of non-adversarial perceptual losses, like style and content loss, improves the translations, especially making the generated images sharper, and makes the model more robust. In previous studies, separate models have been created for healthy and unhealthy brain MRI. However, in a real world clinical setting, choosing between different models can become cumbersome for a medical professional. Moreover, we show that when *pTransGAN* is only trained on healthy data, it performs poorly on unhealthy data (and vice-versa). Thus, in this study, we also present a novel simultaneous training protocol that allows *pTransGAN* to concurrently train on healthy and unhealthy data. As measured by novel metrics that closely match perceptual similarity of human observers, our simultaneously trained *pTransGAN* model outperforms the models individually trained on just healthy and unhealthy data as well as previous literature models. Thus, in this study we present a perceptually improved algorithm to translate both healthy and unhealthy T1 brain MRI into their corresponding T2 scans.

Chapter 1: Introduction

1.1 Motivation

The human body is a complex system and has many different tissue types. Different imaging modalities—computerized tomography (CT), magnetic resonance imaging (MRI), ultrasound (US), positron emission tomography (PET)—may be used to characterize different facets of the same anatomy or provide insight into the molecular behaviors of the tissue or organs. Within the MRI imaging framework, there can also be more types of scans based on the different relaxation times of protons that are excited by the applied magnetic field. Some new techniques are becoming available where a synthesis from one k-Space sampling (the 2D or 3D Fourier transform of the MR image being measured) can lead to multiple synthesized sequences of images. For example, a typical MRI sequence consists of T1-weighted sequence (favorable for observing large brain structures), T2-weighted sequence (useful for pathology), and T2-FLAIR scan (useful for pathology with suppression of signal from water) (Figure 1). Acquiring these separate sequences can lead to exam time of 45 minutes to 1 hour. While these different scans provide complementary information (for example a tumor may not be seen in T1 but could show up in T2 scans), which is quintessential for diagnostic purposes, they also make the MRI a very expensive imaging modality (average MRI cost is $2,600 in the US). With more than 40 million MRI scans done each year in the US alone [1], an obvious question arises: **is there a way to take a T1 scan and use machine learning algorithms to predict the associated T2 scan, thus reducing the acquisition time and improved throughput in the MRI scanner?**

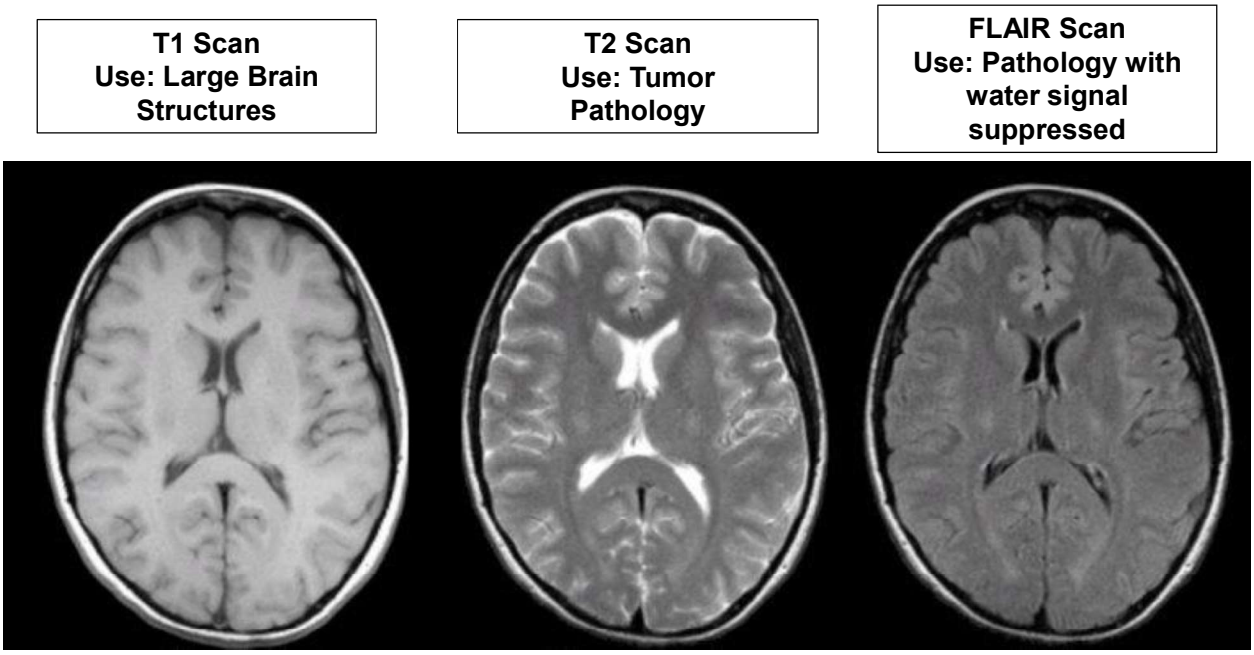


Figure 1: Various types of scans (T1, T2, and FLAIR) done in a typical MRI sequence, which give different perspectives on the same underlying physiological system.

Predicting an MR sequence is an active research area being pursued by academia as well as the private industry. One example, is Facebook's FastMRI project in collaboration with NYU [2]. Determining if later imaging sequences are required based on the first few images, can help reduce the time and cost of MRI. For example, a radiologist could acquire a T1 scan (~10 minutes) and then predict how the corresponding T2 scan. Based on the prediction, the doctor could assess the need for acquiring T2, FLAIR, and the rest of the MRI sequence. This could not only save time, thus allowing more patients to be scanned, but also reduce the monetary costs associated with MRI scans. Increasing the speed of MRI scans can also make them useful especially for the ER and stroke diagnosis. Due to shortages of time, a CT scan is the preferred diagnostic modality in ER [3]. This is especially true for ischemic stroke diagnosis, where timely diagnosis is critical. However, research has shown that diagnosis of stroke made with MRI are 67% more accurate than those made using CT scans. With faster scan times, the MRI could help make more accurate stroke

diagnosis in the ER. Moreover, using an algorithm to predict scans could also be a novel solution to optimizing scan times outside of the ER. For example, younger patients as well as senior citizens have difficulties staying still for long durations, causing motion artifacts in longer MR scans, leading to repeat scans and further costs. However, an algorithmically predicted scan would be devoid of such artifacts, and thus reduce the need for repeated scans. There are numerous initiatives like Image Gently [4], Image Wisely [5], and As Low As Reasonably Achievable [6], that aim to reduce unnecessary scans due to their extra monetary and time costs. Image translation algorithms can help acquire the same information in lesser time and also reduce imaging prices.

In all, the goal of this study is to create a machine learning algorithm that is capable of translating both healthy and unhealthy T1 scans into T2 scans. Leveraging such work, we hope to increase the pace of MRI acquisition and consequently limit the barriers created by long scans times and monetary expenses associated with MRI.

1.2 Previous Literature

Machine learning algorithms in the field of image-to-image translation are specifically designed to convert one possible representation of a system into another representation [7]. For example, taking a day-time photograph of a city landscape and translating it to a night-time photograph is a common image-to-image translation problem [7, 8] (Appendix C Figure 1). Image-to-image translation is a relatively new frontier in medical image analysis but has proven useful in denoising PET scans [9] as well as improving the resolution of MRI taken on low-grade equipment [10]. Often, two or more imaging modalities provide supplementary information and multiple acquisitions are required for a complete diagnostic procedure. Given enough training data, machine learning algorithms have been shown to translate between medical imaging modalities to shorten diagnostic procedures by eliminating unnecessary scans [9, 11].

This is a challenging task because image translation between modalities may introduce unrealistic features, weakening the diagnostic capacities of such techniques. However, in certain situations, like stroke diagnosis, the global image (referring to the entire image instead of components) is more important than the detailed image content (for example thickness of specific edges), thus making image translation a viable option. An example of a global characteristic in an MR scan could be the brain shape and volume; an example of minute anatomical detail could be the boundaries between grey and white matter. A relatively recent approach to image translation are the generative adversarial networks (GAN) introduced by Ian Goodfellow [12]. GANs learn the underlying distribution of available data to generate new data that is not only realistic but aimed to be indistinguishable from the original data. The driving idea of GANs is a competition between two neural networks— the generator and the discriminator. The generator model is given an input image from a source domain (T1 scans for example) and it predicts image data that follows the

distribution of a target domain (T2 scans for example). The discriminator is tasked with distinguishing real image data and data generated by the generator. The two networks, trained simultaneously (the loss function that is minimized is called adversarial loss) but with opposing goals, reach an equilibrium when the generator generates fake images that the discriminator cannot distinguish from real data [12].

Phillip Isola [7] introduced the Pix2Pix GAN algorithm for supervised image translation tasks. In supervised translation problems, there exists a correspondence between images of two datasets i.e. each x_i belonging to a source dataset has a corresponding y_i in the target dataset. The generator of the Pix2Pix algorithm is given an image from the input domain (sketch of a shoe) and translates it to a target domain (colored shoe image) by minimizing the adversarial loss as well as a pixel-to-pixel error (L1 error). Pix2Pix GAN and its variants have also been used in numerous medical image translation tasks over the past few years [9, 11]. Several modifications also have been made to pix2pix to improve the quality of the output images. For example, [13] used a standalone network to calculate the stylistic losses between images and transfer the texture of input image onto the translated image.

There have also been some unsupervised variants of the GAN framework that allow for image translation when paired data is not available. For example, [11] present an architecture based on the CycleGAN framework to translate T1 scans into T2 scans. To the best of authors' knowledge, only one study has been done on semi-supervised image translation for neuroimaging modalities [14]. This study tries to translate T1 MRI to FLAIR MRI by modifying the training protocol for the cycleGAN. When the model is trained on unpaired data, the adversarial loss of the cycleGAN framework is minimized. Paired data is then used to minimize the cycle loss of the cycleGAN framework to ensure consistent mapping between source and target datasets. Previous

semi-supervised schemes have been developed for chest abnormality classification [15], patch-based retinal vessel classification [16], and cardiac disease diagnosis [17].

A major limitation of the pix2pix architecture, and most of the previous attempts at medical image translation, is that its network only minimizes the L1 loss and adversarial losses, which work on the pixel level and assume that each pixel value is independent of other pixels. However, when translating images, there are higher level features, like texture, that arise from a group of pixels. Most of the previous work in medical image translation has largely focused on only reducing the L1 error and not accurately transfer the stylistic features, which assume some spatial relations between groups of pixels. Accurately translating stylistic features can make the output images more perceptually appealing to human observers, thus allowing for faster adoption of medical image translation technologies. Consequently, there are very few novel metrics that can measure the stylistic differences between images, just like humans view perceptual similarity. Thus, it is critical to also evaluate how well do different quantitative metrics measure stylistic level similarity.

Finally, while multiple algorithms exist that perform MR T1 and T2 translation on healthy or unhealthy data [11], to the best of authors' knowledge no single algorithm can perform equally well translation of both healthy and unhealthy T1 scans. A potential limitation of different models for healthy and unhealthy data, is that in a clinical setting, the doctor may not know beforehand which model to use. Moreover, there will always be a chance of missing out information the translated scan when a model from a different domain is used. Thus, for a clinically successful algorithm, it is critical that the model be able to accurately translate both healthy and unhealthy scans.

1.3 Major Contributions

From the above presented discussion of the literature, we see two common trends. Firstly, the medical image translation algorithms lack loss functions that will translate stylistic features from the input to the output image. This limits translation algorithms from creating images that are indistinguishable from ground truth images for a human observer. Secondly, there is no single model that perform equally well on healthy and unhealthy data. If applied in clinical settings, this would mean that different models need to be used for different scenarios, and doctors may not know *a priori* which model to use.

Thus, in this thesis, we propose a new conditional GAN framework, *pTransGAN,* for translating T1 scans into T2 scans, and a training protocol that makes the model perform equally well on healthy and unhealthy data. Based on the pix2pix architecture, but inspired by other works such as [9], *pTransGAN* provides a new generator and a discriminator for more stylistically accurate translation of T1 scans. *pTransGAN* is applicable to both healthy and unhealthy input images without any further changes to the model architecture. While *pTransGAN* may not be ready for diagnostic purposes, we aim to provide a framework through this study that can be used to determine global properties of translated T2 scans, which can help doctors determine if the full MRI sequence is necessary. The specific contributions of our work are as follows:

- *pTransGAN* presents a new generator that is capable of translating high frequency and low frequency components in medical images. This is achieved by the addition of non-adversarial perceptual losses, like the style and content loss. We present a holistic study of the benefits and shortcomings of the addition of non-adversarial losses to GAN training.

- *pTransGAN* makes use of a recently presented U-net inspired generator, which is capable of progressively refining the translated images. This is achieved via multiple ResNet inspired encoder-decoder blocks. The results of image generation by *pTransGAN* are presented on a healthy and unhealthy dataset.
- *pTransGAN* is not specific to healthy or unhealthy data. In this study, we present how without any additional changes to the model architecture, *pTransGAN* can be adapted for healthy and unhealthy data.
- We evaluate numerous training strategies to create a single model that can perform equally well on healthy and unhealthy datasets, thus better mimicking real life clinical scenarios the algorithm would have to face.
- We present a holistic evaluation of *pTransGAN* through both traditional image comparison metrics as well as novel metrics that closely resemble human perceptual similarity. For qualitative comparisons we also present numerous translation examples and discuss what features are generally missed or are translated well by *pTransGAN.*

Chapter 2: Preliminaries

2.1 Introduction

Before diving into how supervised methods (T1 and T2 scans come from the same patient, i.e. the data is paired) are used for image translation, one needs to better understand the general framework of generative adversarial networks (GANs). Thus, this chapter begins by exploring GANs and the loss functions used to train them. Next, we explain why there is a need to update the generic loss function of GANs, which are used to translate images from significantly different domains, like paintings, buildings, and medical images. We introduce perceptual losses, which can be used to translate the stylistic features like the texture and contrast between image domains. We then introduce the multimodal MRI datasets that we will be using for the supervised image translation problem of translating T1 MRI into T2 scans. We explain how these datasets were broken down into the training, testing, and hyperparameter optimization datasets. Finally, in order to compare the generated images and the ground truth images, several metrics are proposed. These evaluation metrics include both traditional image translation metrics as well as more recent metrics that approximate how humans perceive image similarity.

2.2 Generative Adversarial Networks

Generative adversarial networks (GANs) are a type of neural network architecture that consist of two primary networks: a generator and a discriminator. The generator model is provided with a latent variable (often random noise), $v \sim p_{noise}$, as input and learns to map it to an output domain y, i.e. $G: v \rightarrow y$. The discriminator (D) is a network that acts as a binary classifier and

learns to classify data samples ($x \sim p_{data}$) as real, i.e. $D(x) = 1$, and generated samples ($\bar{x} \sim p_{model}$) as fake, i.e. $D(\bar{x}) = 0$ (Figure 2).

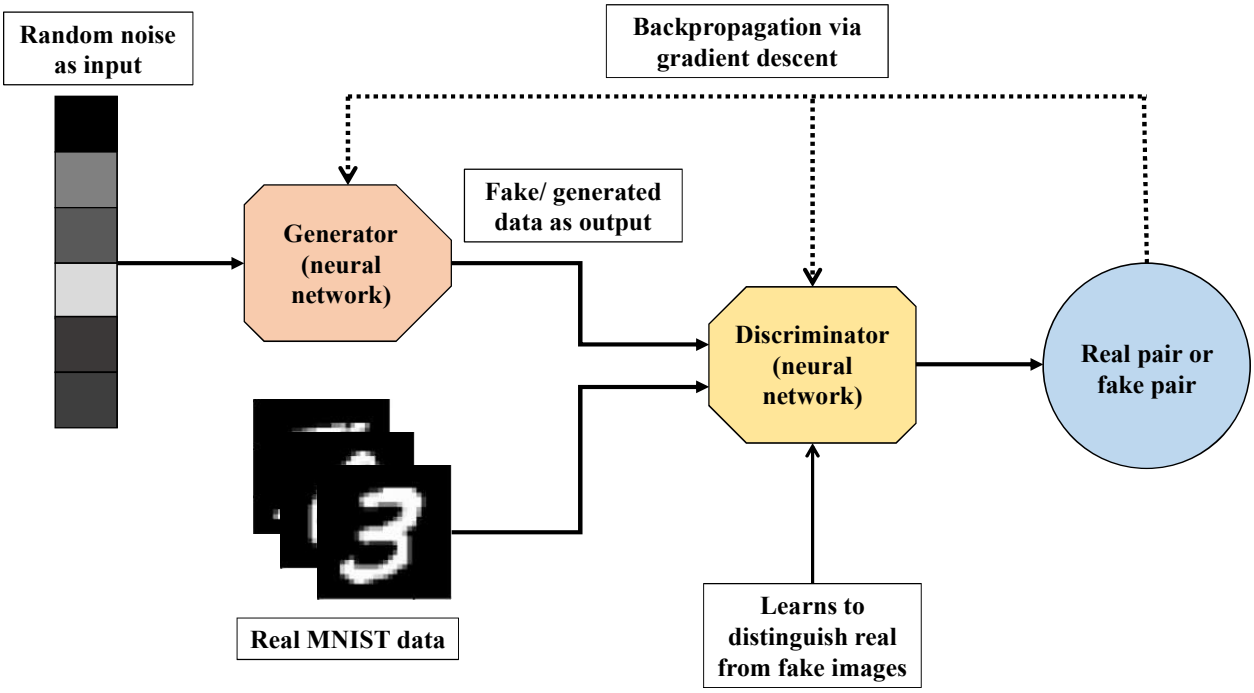


Figure 2: A pictorial representation of the GAN framework where a generator tries to fool a discriminator, which tries to better distinguish between real and fake images. This adversarial nature is then used to iteratively train the generator and discriminator.

In the GAN framework, the generator and discriminator are pitted against each other. The discriminator tries to better distinguish between data from source and model distributions. On the other hand, the generator tries to better fool the discriminator by creating data that is indistinguishable from the source data, i.e. $p_{model} \approx p_{data}$. To achieve this, the following adversarial loss function ($\mathcal{L}_{GAN}$) can be used, where E represents the expected value:

$$\mathcal{L}_{GAN} = E_{x \sim p_{data}}[logD(x)] + E_{v \sim p_{noise}}[log(1 - D(G(v)))] \quad (1)$$

The adversarial competition between the discriminator and the generator is best represented by G trying to minimize the adversarial loss and D trying to maximize it [12]. Previous work has found that alternatively training the two networks while keeping one of them fixed helps with the vanishing gradients problem arising from completely training a discriminator for a fixed generator [9]. The adversarial loss also helps the generator in better modeling high frequency features like edges [18] while also avoiding to produce blurry results. GAN models do not converge to a local minimum but rather achieve the Nash equilibrium (or a saddle point) because the cost function of each sub-network is affected by the parameters of the other network [12].

2.3 Conditional GANs and image-to-image translation

The image generation and translation tasks are inherently different, and thus require the GAN to be modified. Unlike like the generator used for image generation, the generator for image translation tasks maps a source domain image ($x \sim p_{source}$ instead of $x \sim p_{noise}$) to its ground truth image in the target domain ($y \sim p_{target}$) [7], i.e. $G(x, v) = \hat{y} \sim p_{model}$. Image translation tasks can be considered as regression tasks assuming that the two domains represents different views of the same underlying system. Moreover, the loss function is further adapted in this conditional GAN (cGAN). Instead of manually constructed loss functions to measure the difference between the target and generated images ($\hat{y} \sim p_{model}$), the discriminator model is used as a binary classifier. The discriminator model learns to classify pairs of source and corresponding ground truth translated image as real ($D(x, y) = 1$). On the other hand, the discriminator learns to classify concatenations of source and generated images as fake ($D(x, \hat{y}) = 1$). The new loss function can be written as [9]:

$$\mathcal{L}_{cGAN} = E_{x,y}[logD(x, y)] + E_{x,v}[log(1 - D(x, G(x, v)))] \quad (2)$$

Previous studies have shown that cGAN, relying on just the adversarial loss, fail to produce consistent results in the target domain. [7, 19] showed that there could be variations in the global structure of generated images when cGAN are trained solely through adversarial loss. A pixel reconstruction loss, like the L1 loss, is suggested to avoid this issue. In this study, we implement the L1 loss or the mean absolute error (MAE) between the source and generated images as follows:

$$\mathcal{L}_{L1} = E_{x,y,v}|| y - G(x, v)||_1 \quad (3)$$

The overall loss function for cGAN becomes:

$$\min_{G} \max_{D} \mathcal{L}_{cGAN} + \lambda * \mathcal{L}_{L1} \tag{4}$$

In Equation 4, the hyperparameter $\lambda > 0$, is the weight given to the L1 loss and is also a training hyperparameter.

2.4 Perceptual Similarity

The learning objective of the cGAN defined in (4) is a per-pixel measure and assumes pixel-wise independence. However, this is very different from how humans measure perceptual similarity. Such perceptual similarity is dependent on the high-order image structure and may not actually constitute a distance measure, like defined in (3). For example, when an image is shifted by only a few pixels, the human brain will quickly recognize the perceptual similarity between the original and shifted images, but an L1 loss will perceive these two images as drastically different [13]. Moreover, using a pixel-wise difference loss has also led to blurry results [20]. The result of this is that global scale features are translated very well, but there is loss of detail and even distortions of high-frequency features.

Image translation tasks are further challenging because along with global consistency with target domain images, the translated images also need to exhibit the sharpness of high-frequency features. The computer vision community has found that the internal feature map activations of deep neural networks, trained on high-level classification tasks, are very good extractors of perceptual features. Previous work has included using the VGG-19 [21] network's feature space for neural style transfer [22], conditional image synthesis [9], and image super-resolution [13].

Thus, to attain perceptual similarity, along with adversarial and L1 losses, we include non-adversarial losses derived using the deep features of the VGG-19 network (referred to as VGG hereon), which are frequently used for image transfer tasks [13, 22]. The VGG network has 5 convolutional blocks, each having 2-4 convolutional layers and 3 fully connected layers. Even though the VGG network is pretrained on the diverse ImageNet dataset, it has the advantage of being a very deep network with numerous convolutional blocks acting as excellent feature

extractors of large receptive fields. These features can then help calculate the stylistic and content features in images. The resulting losses ensure that stylistic features like texture and contrast are transferred across domains along with maintaining image sharpness, fine details, and global consistency. Consistent with previous literature [9], two perceptual losses, style and content, are considered.

2.4.1 Style Loss

Computing the style loss consists of using a convolutional neural network to extract stylistic features, like texture, from the generated and ground truth target image and then minimizing the discrepancies between them. Determining the correlations across feature maps can help calculate the stylistic distribution in images. Let $F_{i,j}$ be the feature map extracted from the i^{th} convolutional block and j^{th} layer of the feature extractor for an image, x, from the source domain. Since previous studies [9, 22] have primarily used the first layer of convolutional blocks, $j = 1$ for all convolutional blocks considered, and is hereon omitted. The initial convolution layers have been shown to extract stylistic features, like texture, and are often in neural style transfer problems [13].

Each feature map, F_i, has the dimensions h_i, w_i, d_i, which correspond to the height, width, and depth of the map. Given an image y, calculating the Gram matrix (Equation 5), $Gram_i(y)$ of a feature layer i includes flattening the layer to get a "feature vector" and then taking the inner product of the feature maps in the h_i and w_i dimensions with themselves and averaging over all locations. The Gram matrix represents the feature correlations, which are the stylistic features of the image.

$$Gram_i(y) = \frac{1}{h_i * w_i * d_i} \sum_{h=1}^{h_i} \sum_{w=1}^{w_i} F_i(y) * F_i(y) \tag{5}$$

The style loss (Equation 6) is then defined as the square of the Frobenius norm of the difference between the feature correlations of the generated image, $\hat{y}$, and ground truth target image, y, over all the selected convolutional blocks.

$$\mathcal{L}_{style} = \sum_{i=1}^{Total\ Blocks} \lambda_{style,i} * \frac{1}{4d_i^2} * ||Gram_i(y) - Gram_i(\hat{y})||_F^2 \tag{6}$$

Here, $\lambda_{style,i} > 0$, weighs contribution of the i^{th} convolutional block to the overall stylistic loss. Appendix A discusses the identification of $\lambda_{style,i}$. It is a tunable hyperparameter that will be determined via a grid search algorithm. One should note that the Gram matrix function will return a matrix of the shape $d_i x\ d_i$, thus the Frobenius norm is scaled by $4d_i^2$.

2.4.2 Content Loss

While translating stylistic features, like texture, is critical, the discrepancies between the actual features extracted by the feature extractor also needs to be minimized. This is referred to as the content loss and does not capture textural features that the style loss would. Since the feature extractor model is excellent at extracting low frequency components of images, it serves in addition to the L1 loss in achieving global consistency and enhancement of low frequency features.

In defining the content loss, once again only the feature map of the first layer of convolutional blocks of the feature extractor are considered. The content loss between the target image y, and the translated image $\hat{y}$, is defined as follows:

$$\mathcal{L}_{content} = \sum_{i=1}^{Total\ Blocks} \lambda_{content,i} * || F_i(y) - F_i(\hat{y})||_F^2 \quad (7)$$

$\lambda_{content,i}$ scales the contribution of the i^{th} block to the overall content loss. It is a tunable hyperparameter that will be determined via a grid search algorithm. Appendix A discusses the identification of $\lambda_{content,i}$.

2.5 Datasets

This study utilized two independent datasets: the IXI dataset and the BRaTS2020 dataset. While developing computer vision algorithms for neural images, one must remember that the algorithms might encounter scans from both healthy patients as well as patients with disease-related abnormalities. Thus, a dataset of healthy scans (IXI) is complemented with a dataset showing brain tumors (BRaTS2020). Both of these datasets have paired images only, i.e. the T1 and T2 scans come from the same patient. For all datasets, the images were normalized to achieve comparable voxel sizes. Image intensities were normalized to the [0,1] range. Image acquisition protocol information for each dataset, number of images used for training and testing, and registration details are provided below.

2.5.1 IXI Dataset- Paired Healthy Data

The IXI dataset (http://brain-development.org/ixi-dataset/) contains scans (T1, T2, PD-weighted, MRA images, and diffusion weighted) from 577 patients. However, only the T1 and T2 scans in the axial direction were utilized in this study (Figure 3). The scans were not originally registered, and previous work has shown that paired registered images provide superior results in translation tasks [11]. Thus, the images were resampled to (1,1,1) spacing and were reordered to be closest to canonical (RAS+) orientation. The MNI mask [23] was then applied to the images so that all images have the same size. All images were sized to 256x256 pixels. The scans were not skull stripped. 461 patients were selected for training, and 58 for testing. Each patient had 180 scans in the axial direction; 11 images from the middle were chosen for each patient. Thus, a total of 5071 T1 and T2 training images and 638 T1 and T2 testing images were available. The data from remaining 58 patients were used for hyperparameter optimization (471 images for training

models and 167 images for testing). The data used for hyperparameter optimization was not used for training and testing of the final models. The T1- and T2 images utilized in this study here were acquired with the following parameters. T1-weighted images: TR=9.813ms, TE=4.603ms, flip angle=80, volume size = 256 × 256 × 150, voxel dimensions = 0.94 × 0.94 × 1.2 mm3, sagittal orientation. T2-weighted images: TR=8178ms, TE=100ms, flip angle=900, volume size = 256 × 256 × 150, voxel dimensions = 0.94 × 0.94 × 1.2 mm3, axial orientation.

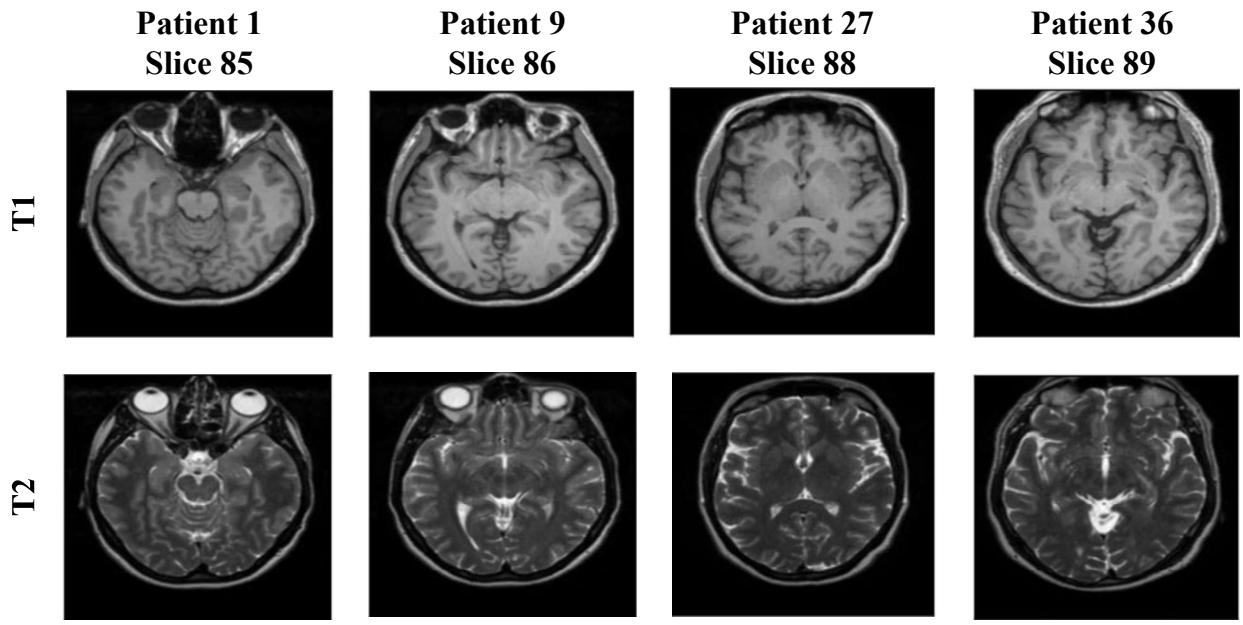


Figure 3: Four example scans from the IXI dataset. Scans are in the axial direction and are taken from healthy patients. The T1 and T2 images are paired.

2.5.2 BRaTS2020 Dataset- Paired Unhealthy Data

The BRaTS2020 dataset (https://www.med.upenn.edu/cbica/brats2020/) contains T1, T1-weighted, and T2, and T2-FLAIR, scans from 494 patients which show brain tumors or gliomas from n=19 institutions (Figure 4). Since the data is acquired from different sites, no common data acquisition protocol existed. Only the T1 and T2 images in the axial direction were used in this

study. The dataset is already skull stripped and registered, thus the only pre-processing that was done was that the images were normalized to [0,1] range. Data from 369 patients were reserved for training purposes and 62 patients were kept for testing. Each patient has 150 scans in the axial direction; 14 scans from the middle of the brain were used for each patient. This led to 5166 T1 and T2 training images and 868 testing images. All images were sized to 256x256 pixels. No data from the unhealthy dataset was used for hyperparameter optimization.

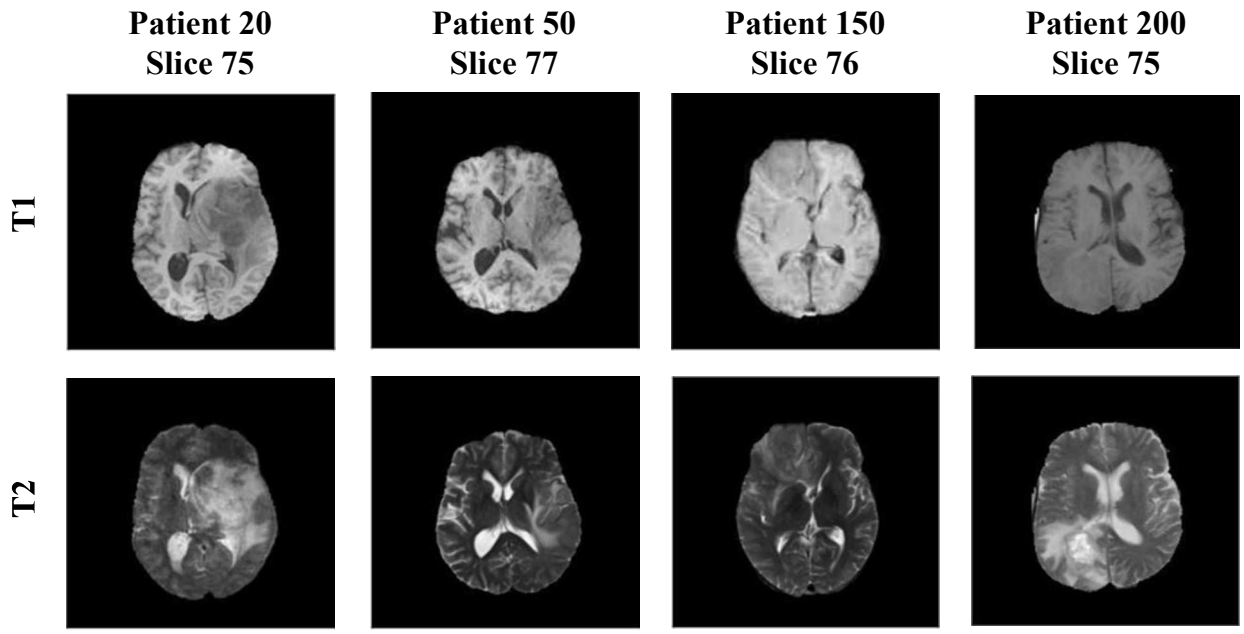


Figure 4: Four example scans from the BRaTS2020 dataset. Scans are in the axial direction and are taken from patients with gliomas. The T1 and T2 images are paired.

2.6 Image Comparison Metrics

The translated and target images were compared quantitatively. Previous work has suggested that there is no consensus in the computer vision community regarding which are the most informative metrics [24]. Thus, a mixture of traditional L2 norm-based metrics as well as more recent deep convolutional network based metrics were used. Table 1 explains how these metrics provide a comprehensive judgment of the quality of the translated images. Larger is better for PSNR, SSIM, UQI, and VIF. Since LPIPS and MSE are analogous to "losses" smaller is better for these measures. Moreover, an experienced radiologist was provided with the generated images and target images and asked to qualitatively compare the overall structure and the fine details.

Table 1: Qualitative metrics used to compare generated and ground truth images and their purpose

Metric	Source	Purpose
Peak Signal to Noise Ratio (PSNR)	[9]	Ratio between maximum possible power and power of corrupting/ erroneous signal. "Quality" of generated image compared to corresponding real image.
Structural Similarity Index (SSIM)	[25]	Compares corresponding pixels and their neighborhoods. Ignores aspects of image not relevant to human perception.

Mean Squared Error (MSE)	[9]	Compare true and generated pixel values
Learned Perceptual Image Patch Similarity (LPIPS)	[26]	VGG-based perceptual similarity close to human judgment
Universal Quality Index (UQI)	[27]	Measures loss of correlation, luminance distortion, and contrast distortion
Visual Information Fidelity (VIF)	[28]	Human visual system based metric that correlated 96% with human perceptual judgment

One must note that multiple studies have shown that traditional metrics like the Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE) are insufficient in assessing structured outputs like images [26, 27]. For example, blurring causes large perceptual but small PSNR and MSE changes. Hence these metrics do not correspond well with human perceptual judgment. However, they are presented so that the results of the current model can be compared with previous studies.

On the other hand, the Structural Similarity Index (SSIM) and Visual Information Fidelity (VIF) try to determine how close two images are from a human judgment standpoint. However, previous work has also suggested that human perceptual judgment may not be a distance-based function, thus limiting these metrics [29]. Nevertheless, the recently introduced Learned

Perceptual Image Patch Similarity (LPIPS) metric has been reported to outperform previous metrics as a measure of perceptual image quality, and thus it is also presented here [26].

Chapter 3: MRI Image Translation

3.1 Introduction

Supervised image translation spans the subset of conditional generative algorithms that are trained with an image from a source domain and its corresponding image from the target domain. We call such training data paired; in the context of our study the T1 and T2 MRI scans come from the same patient. Such data may not always be available in a clinical setting and may be expensive to compile for research purposes, but when available, can be used to teach the network how to translate minute anatomical features between imaging modalities (for example, how to translate the occipital horn between imaging modalities). Thus, this chapter is dedicated to exploring and evaluating conditional generative algorithms for paired MR data.

A general solution to the problem of supervised image translation using the GAN framework was provided in 2017 in the form of pix2pix by [7]. The proposed architecture consists of a generator that takes an image from the source domain (ex. daytime picture of a city landscape) and minimizes the adversarial and pixel-reconstruction losses to produce the corresponding image in the target domain (ex. nighttime picture of the same city landscape). The discriminator is a deep convolution network that acts a binary classifier that differentiates between ground truth and generated (fake) images. The network has been modified for various purposes: namely, to calculate style losses to transfer texture between domains [13] and reduce blurriness of images [30].

In this study, we make some important modifications to both the generator and discriminator networks of the pix2pix architecture and propose the *pTransGAN,* with the aim of translating a T1 scan into a T2 scan. We also investigate the addition of perceptual losses (style

and content specifically) to the traditional adversarial and pixel-reconstruction losses. Finally, we explore both healthy and diseased datasets of brain MRI.

3.2 Model Architecture

The *pTransGAN* model consists of two neural networks working in tandem to optimize the translation of input images from the T1 domain to T2 domain. The first model is the discriminator which is tasked with learning to discriminate real pairs (real T1 and real T2) and fake pairs (real T1 and generated T2). The second model is the generator, which learns how to better "fool" the discriminator into not being able to distinguish between generated and real images. In this section, we explain the architectural choices made while creating the discriminator and generator models for *pTransGAN*.

3.2.1 Discriminator Design

The discriminator is a deep convolutional network that acts as an image classifier. The inputs to the discriminator are the source image (T1) and the target image (T2). The output of the model is the likelihood of whether the target image is a generated image or the ground truth image. The discriminator model is a modification of the PatchGAN introduced in [7].

The PatchGAN design is based on the effective receptive field of the model, which maps one output activation of the model to an area of the input image. For example, the conventional discriminator in the pix2pix architecture has an effective receptive field of 70x70. In other words, this means that each output of the discriminator maps to an area of 70x70 pixels in the input image. Thus, a 70x70 PatchGAN will classify 70x70 pixels of the input image as real or generated translations. [7] found that a 70x70 PatchGAN resulted in superior performance of the generator compared to the performance from a 1x1 receptive field (PixelGAN) and 256x256 receptive field (ImageGAN) models. The receptive field is not the shape of the output (or feature map) from the discriminator model; it is relation between one output of the discriminator and the input image.

Starting from the output layer of the discriminator and working backwards, one can calculate the receptive field of the model through the follow Equation:

$$Receptive\ field = (output\_size\ -\ 1) * stride\ +\ kernel\_size \tag{7}$$

The conventional receptive field used in the pix2pix architecture is 70x70 [7]. However, previous literature has also suggested that using a smaller receptive field of 16x16 could help improve the sharpness of the image [9] because this causes the discriminator to consider smaller patches for comparison. In this study we explore two discriminator architectures with receptive fields of 70x70 (Figure 5) and 16x16 (Figure 6). For the 70x70 PatchGAN, the two input images (source and target images) are concatenated (channel-wise) and passed through 6 convolutions with 64, 128, 256, 512, 512, and 1 spatial filters. The stride for the first four convolutions is 2, and stride for the rest is 1. For the 16x16 PatchGAN, two convolutions with 64 and 128 spatial filters are applied to the input, both with a stride of 2. For both PatchGAN models, the convolution layers are followed by batch normalization and Leaky-ReLU ($\alpha = 0.2$). Finally, to get the output probability map, a convolution layer with output dimension of 1 and sigmoid activation is used in both PatchGAN models. Kernel size of 4x4 are used for all the convolutions in both discriminator models. Using the hyperparameter optimization process described in section Appendix A, it was determined that the 70x70 receptive field provided better results, thus it is used for the rest of *pTransGAN* experiment.

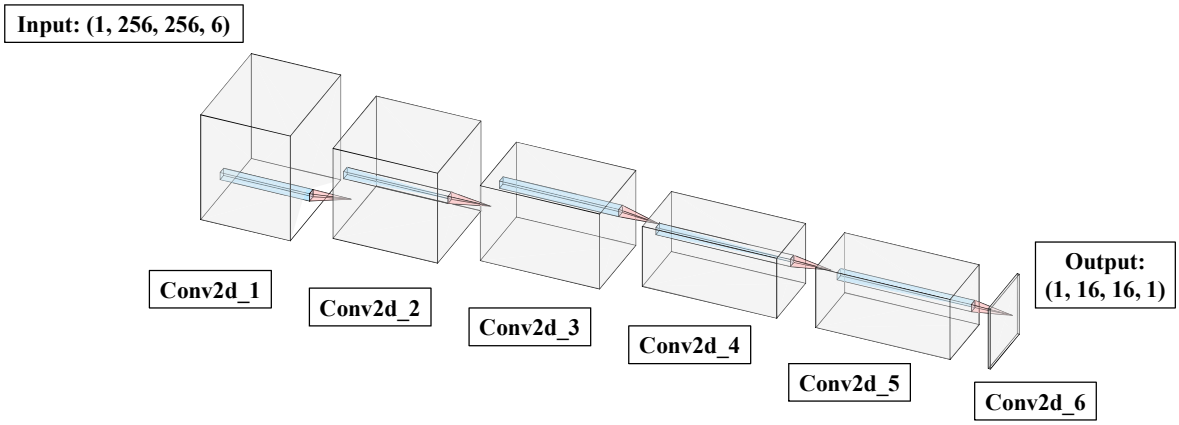


Figure 5:Visualizing the architecture of the 70x70 PatchGAN discriminator with six convolutional layers.

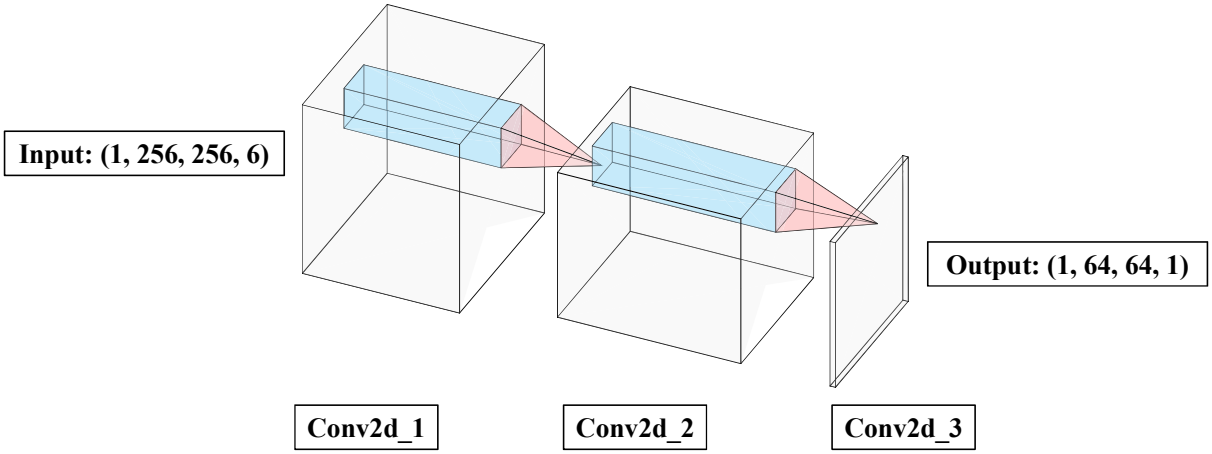


Figure 6: Visualizing the architecture of the 16x16 PatchGAN discriminator with two convolutional layers.

3.2.2 U-blocks

The fundamental building block of the generator in the *pTransGAN* is the U-block, which can be thought of as an encoder-decoder structure with skip connections. The U-block architecture is used since the task of image translation can be thought of as mapping an input tensor to a tensor

with a different surface appearance but with the same underlying structures. The U-block structure is inspired by [31] and was adapted for image translation by [7].

The U-block (Figure 7), a fully convolutional structure, takes a 256x256x3 image as an input and encodes it to a high-dimensional representation, called the bottleneck. This high-dimensional bottleneck is then up-sampled through the decoding path to acquire a translated image. Instead of taking in random noise as input, an image from the source domain is given to the U-block, thus the v subscript from Equation 3 can be omitted. The encoding path consists of seven convolutions with filters 64, 128, 256, 512, 512, 512, 512. The bottleneck consists of 512 filters and is followed by a ReLU activation. Each convolution has a kernel size of 4x4 and a stride of 2 and is followed by batch normalization and LeakyReLU ($\alpha = 0.2$). The decoding path has 7 convolutions with filters 512, 1024, 1024, 1024, 512, 128, and 64. Each convolution is again followed by a batch normalization and a LeakyReLU activations ($\alpha = 0.2$). However, the final layer has a Tanh activation. The output of the U-block is again a 256x256x3 array.

The decoding path inverts the down-sampling done by the encoding path and links the high-dimensional bottleneck to a 3-channel output image. The bottleneck is considered to be high-dimensional since it has ~250,000 elements ($height * width * depth$). In order to reduce overfitting, the first three layers of the decoding path have a dropout layer associated with them ($rate = 0.5$). All of the layers were initialized with a Glorot initializer [32]. The U-block also includes skip connections that connect the encoding and decoding channels, i.e. the encoding level features are concatenated with corresponding inputs from the decoding side. For example, the 3rd and 6th layers are connected, and 2nd and 7th layers are connected. One must note that the skip connections play a vital role in the U-block architecture. By concatenating the low level

information between the encoding and decoding paths, the skip connections preserve information that would have otherwise been lost when the encoder down-sampled the input images [9].

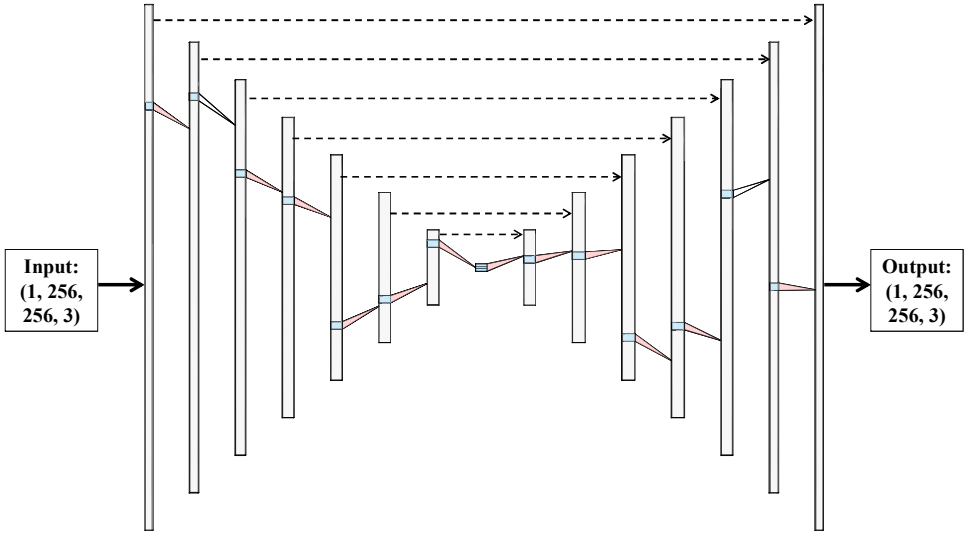


Figure 7: A pictorial representation of the U-block architecture. The dashed lines show the skip connections between the mirroring layers of the encoding and decoding paths.

3.2.3 Generator Architecture

Translation of medical images between domains is more challenging than non-medical translation problems because a high level of fidelity is required when translating very detailed structures. Moreover, losing this detailed information during translation will hinder the clinical applicability of image translation algorithms. Thus, we utilize a U-block based and ResNet [33] inspired generator model that can translate the detailed features in medical images (Figure 8).

We propose a generator in which U-blocks are connected in an end-to-end manner, such that the output image of one U-block is the input to the next U-block, and this continues through the N^{th} block. This is similar to the cascading residual blocks found in the ResNet structure. The consecutive nature of the U-blocks allows the generated image to be increasingly refined; the

encoding-decoding pairs provide an end-to-end translation. Even though the underlying idea between ResNets and the cascading U-blocks is similar, they have some significant differences. The ResNet features multiple "residual blocks" with only 2-4 convolution layers. In order to increase the generative power of *pTransGAN* our U-blocks have 14 convolutional layers, which makes them different from the residual blocks. Another significant difference is in the vanishing gradients problem. In ResNet, the identity mapping connects the input and output within the residual block, which helps with reducing the vanishing gradient problem. In the *pTransGAN* U-blocks, the intermediate skip connections pass low level information and also help in mitigating the vanishing gradients problems [9]. In the final U-block, the output

Each U-block contains 105,292,211 parameters (105,280,771 trainable and 13,440 non-trainable). Appendix A describes how extensive hyperparameter optimization was done to determine that 6 U-blocks (Figure 8) should be used in the *pTransGAN* architecture. This is also consistent with previous literature [9].

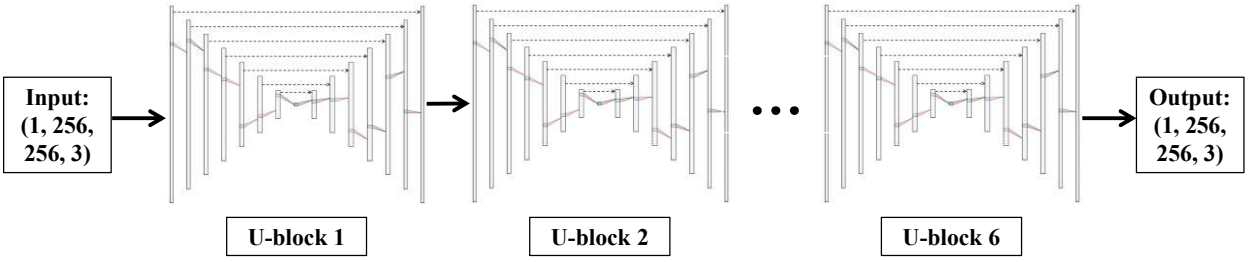


Figure 8: 6 U-blocks are used in the *pTransGAN* architecture. This was determined through extensive hyperparameter optimization.

3.3 Training Protocol

In all, the *pTransGAN* framework has a deep convolution neural network and a U-block based generator, which is penalized through pixel and adversarial losses. The generator is also trained via non-adversarial losses, i.e. the style and content losses, in order to generate translated images that closely match the ground truth images. *pTransGAN* is trained through the min max optimization of the following loss function:

$$\mathcal{L}_{pTransGAN} = \lambda_{cGAN}\mathcal{L}_{cGAN} + \lambda_{L1}\mathcal{L}_{L1} + \lambda_{style}\mathcal{L}_{style} + \lambda_{content}\mathcal{L}_{content} \quad (8)$$

Where λ_{cGAN}, λ_{L1}, λ_{style}, and $\lambda_{content}$ are hyperparameters that determine the contributions of the different loss functions. The weights of these hyperparameters, $\lambda_{style,i}$ and $\lambda_{content,i}$, and the optimization process used to identify these weights are discussed further in section *3.3.3.* For training of the model, we used the ADAM optimizer [34] with a learning rate of 0.0002 and momentum of 0.5. [35] showed that a batch size of 1 is the best for image translation tasks, thus a batch size of 1 was utilized. For training all the models, 100 epochs were used. Unlike previous literature [9], the generator and discriminator were trained simultaneously (Table 2). All training was done on a single Nvidia GeForce RTX 2080 Ti GPU. While training time was directly dependent on the size of the training dataset, on average the models trained for 96 hours. In comparison, the inference time was 5.5 seconds on average.

Table 2: Training protocol for the pTransGAN model.

Training protocol for *pTransGAN*
• Load paired training dataset$\{(x_j, y_j)\}_{j=1}^{N_{images}}$ where N is the size of training dataset • Set $N_{epochs} = 100$ • Load pre-trained VGG-19 feature extractor weight ImageNet weights • Initialize generator and discriminator model weights using Xavier initializer

for epoch in N_{epochs} **do**:
  for image in N_{images} **do**:
  • Train discriminator on a pair of real T1 and real T2
  • Train discriminator on a pair of real T1 and generated T2
  • Calculate $\mathcal{L}_{cGAN}, \mathcal{L}_{L1}, \mathcal{L}_{style}, \mathcal{L}_{content}$
  • Backpropagation to update discriminator and generator weights
  end for
end for

3.4 Statistical Measures

Even though the quantitative metrics presented in section *2.6* can help quantify the quality of the translated image in comparison to the ground truth image, they may not be very helpful in determining if the addition of more loss components (perceptual losses specifically), help in better image translation. Thus, to determine the statistical significance of the differences in translation of images with and without perceptual losses, we perform two types of paired Wilcoxon signed-ranked tests (one-sided) [36]. Both types of tests are performed on the difference between the six metrics presented in section *2.5* coming from translation with and without perceptual losses. When describing these tests as well as in future sections, the baseline model refers to the model without any perceptual losses. For the first type (for the metrics in which higher means better, i.e. PSNR, SSIM, UQI, and VIF), we have the null hypothesis that the median difference in the metrics from the baseline model and models trained with perceptual losses is not greater than zero. In the second type of test (for the metrics in which lower means better, i.e. LPIPS and MSE), we have the null hypothesis that for models with additional perceptual losses, the median difference between metrics from the baseline model will not be lower than those from the models with perceptual losses. We choose to do a Wilcoxon test instead of a paired t-test since we cannot assume that the differences in these metrics are normally distributed.

Chapter 4: Experiments and Results

4.1 Introduction

In previous literature of medical image translation with cGAN, very few models have made use of non-adversarial losses, like style and content, to ensure that the translated images have correct stylistic features like texture. Thus, we added two non-adversarial perceptual losses to *pTransGAN,* namely style and content loss. Since the effects of these individual loss components and whether the combination of adversarial and perceptual losses can lead to any synergistic effects, we first comprehensively evaluate the effect of addition of different loss components. We do this experiment on the healthy IXI dataset. Addition of non-adversarial losses can also lead to unforeseen instabilities in training of the discriminator; thus we evaluate multiple avenues to stabilize training with perceptual losses. Next, we transition to creating a single *pTransGAN* model that can perform equally well on both healthy and diseases datasets, thus providing a more clinically useful model.

4.2 Analyzing loss function components

Even though the *pTransGAN* framework largely builds on empirical pix2pix model, there are some significant additions to its loss function. In addition to training on the adversarial loss, *pTransGAN* trains on non-adversarial perceptual losses, like the style and content losses. We did this so that the generator could capture both global features, low frequency components, and minute high frequency details that are crucial to medical images. Hence to study the contributions of the different loss components to the performance of the *pTransGAN* model, we ran three experiments with combinations of the loss function components:

1. $\mathcal{L}_{pTransGAN,0} = \lambda_{cGAN}\mathcal{L}_{cGAN} + \lambda_{L1}\mathcal{L}_{L1}$
2. $\mathcal{L}_{pTransGAN,1} = \lambda_{cGAN}\mathcal{L}_{cGAN} + \lambda_{L1}\mathcal{L}_{L1} + \lambda_{style}\mathcal{L}_{style}$
3. $\mathcal{L}_{pTransGAN,2} = \lambda_{cGAN}\mathcal{L}_{cGAN} + \lambda_{L1}\mathcal{L}_{L1} + \lambda_{style}\mathcal{L}_{style} + \lambda_{content}\mathcal{L}_{content}$

For each experiment, we first trained and tested the model on the healthy images (IXI dataset). However, in a clinical setting, our algorithms would also have to perform image translation on unhealthy images. Thus, we tested the three models (trained on only the healthy dataset) on brain MRI showing tumors (the BRaTS2020 dataset). Testing the models on data coming from a distribution that the models have not seen before will also help us test their generalizability to alternate imaging conditions and clinical domains.

In order for equal comparison, all generators had identical architectures (6 U-blocks) and a 70x70 PatchGAN discriminator. All other training parameters remained same as discussed in section *3.4.* For quantitative comparison, all the evaluation metrics discussed in section *2.5* were used to characterize the performance on the test datasets. For qualitative comparison, we provide

a side-by-side comparison of source (T1), real target (T2), and model generated (T2 as well) images.

4.3 Perceptual Losses Create Sharper Images But Destabilize Training

Figure 9 shows how did the different average loss components (adversarial, L1, style, and content) change over 100 epochs of model training. In each graph, the different curves represent the models trained with the three combinations of loss functions discussed in *4.2.* In the figures hereon, adversarial loss is referred to as BCE (binary cross entropy) and L1 loss as MAE (mean absolute error). Finally, the total average loss per epoch is also presented.

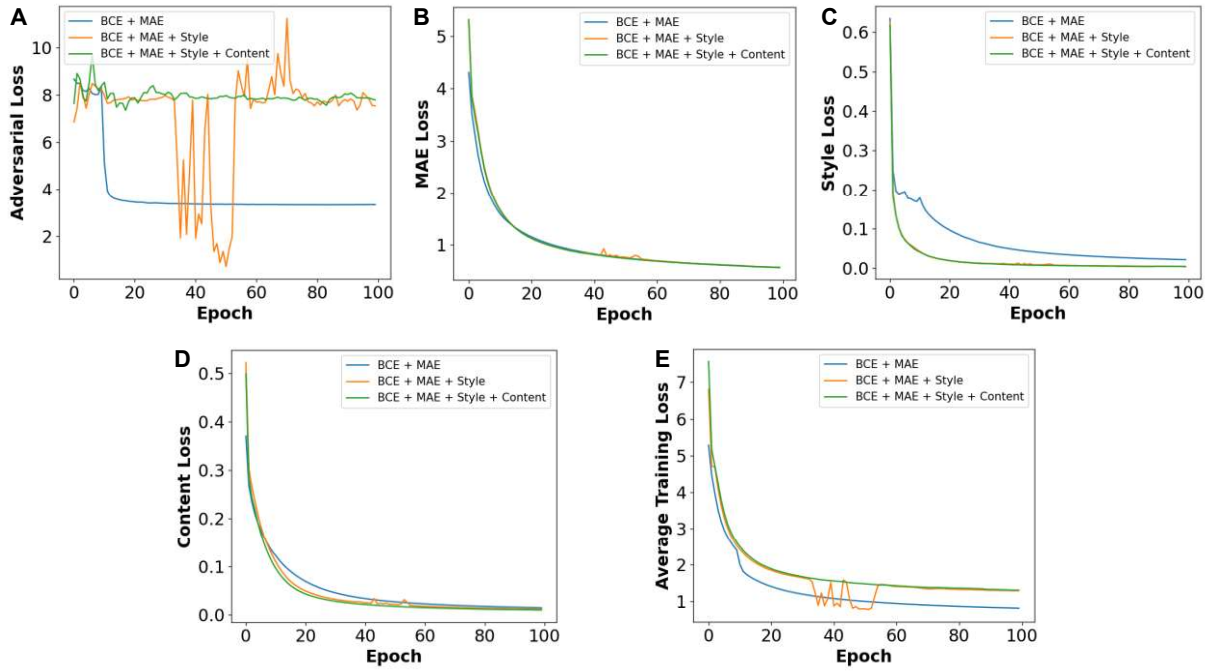


Figure 9: Plots showing how the different average loss components (A: adversarial loss (BCE), B: L1 loss (MAE), C: style loss, D: content loss) change over the 100 training epochs for models training on adversarial and L1 loss (blue), adversarial, L1, and style (orange), and adversarial, L1, style, and content (green). The average total loss is also presented (E).

Figure 9A shows that when perceptual losses are added in addition to adversarial losses, the training of the adversarial loss becomes very unstable (large oscillations seen). Since the

training of a GAN entails finding an equilibrium between the losses of the generator and the discriminator, large variations in adversarial loss will lead to poorer training. Previous studies have shown that applying the spectral normalization to the weights of the discriminator, increasing the learning rate of the discriminator, and training the discriminator more often than the generator can all lead to more stable training. In Appendix B, we show that increasing the discriminator learning rate and applying spectral normalization can stabilize the adversarial training, and thus are used hereon.

When only the adversarial and MAE loss are minimized, style and content loss also consequently decrease, albeit at a significantly slower pace (Figure 9C for example). This happens because when the MAE between two images decreases, the stylistic features of the images also match up better. Two images with zero MAE between their pixels would have the same stylistic features. But when MAE is not zero, we could improve the style loss without changing the MAE. From Figure 9B, we see that the average MAE loss for all three models is approximately the same at the end of 100 epochs. However, the style loss is visibly lower for models training on perceptual losses than the model that trained just on adversarial loss. This means that our style loss function is actually extracting stylistic features and minimizing the style loss between images which the MAE loss cannot capture. This is not the case for the content loss (Figure 9D) since all the models achieve a similar content loss value at the end of 100 epochs. Finally, while comparing the average training loss across the models, the loss for the models training on both perceptual and adversarial losses is higher since there are more loss components to account for. Figure 9C shows that the decrease in stylistic loss when training on adversarial and MAE loss is not as smooth as it is when the models are training with non-adversarial losses as well. When not training with perceptual loses, we see that around epoch 5 and 15, there is are gradual increases in stylistic losses as well.

Thus, in order to ensure a constant and faster reduction of perceptual losses, we recommend training image translation algorithms with both adversarial and perceptual loss functions.

All of the three models were tested on an unseen healthy images dataset. There is only a small quantitative improvement with the addition of perceptual losses, with all the metrics displaying similar spreads (Figure 10 and 11). However, when the Wilcoxon tests are performed on the difference between the metrics, with taking the adversarial and MAE loss trained model as the baseline model, we find that models with perceptual losses outperform the baseline models in the SSIM and LPIPS metrics (p-value < 0.001). Since the LPIPS loss is significantly lower for models with perceptual losses, this indicates that their image translations are likely to be more perceptually similar to ground truth images for a human observer since LPIPS most closely matches human visual similarity. For the other metrics, namely PSNR, MSE, VIF, and UQI, we find that the differences are not statistically significant.

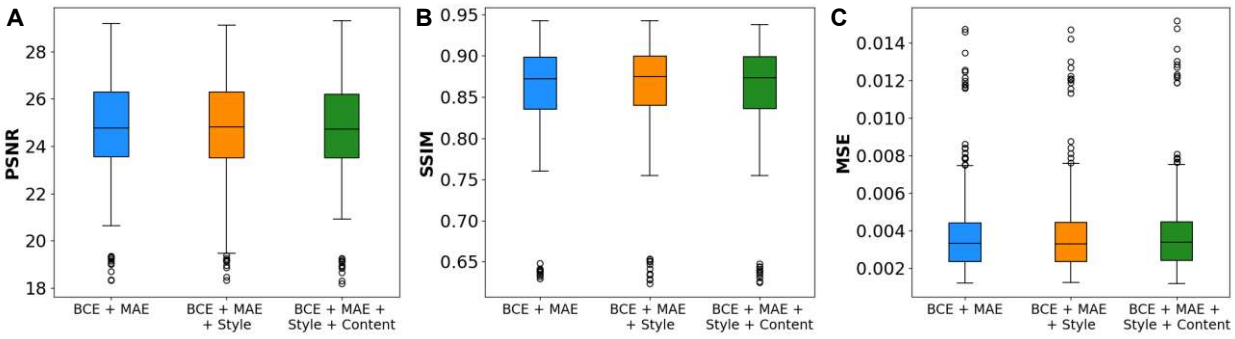


Figure 10: Comparing the traditional metrics (A: PSNR, B: SSIM, and C: MSE) for the models that trained on adversarial and MAE loss (blue), adversarial, MAE and style loss (orange), and adversarial, MAE, style and content loss (green). Models tested on healthy IXI dataset.

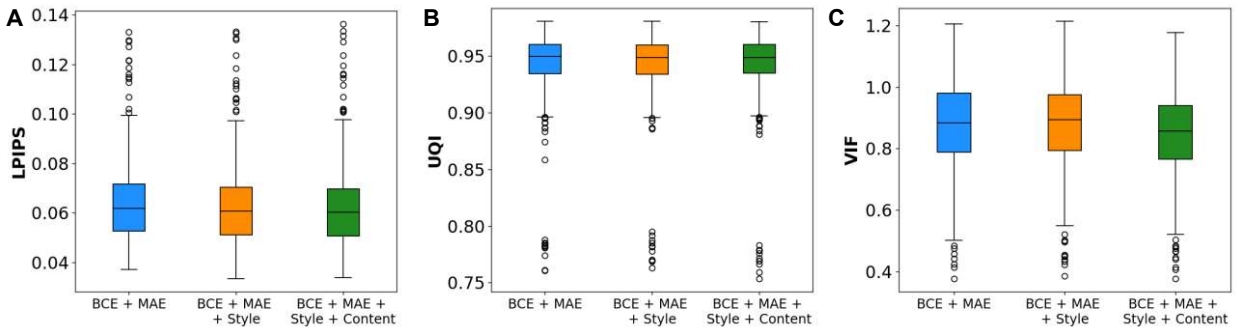


Figure 11: Comparing the novel metrics (A: PSNR, B: SSIM, and C: MSE) for the models that trained on adversarial and MAE loss (blue), adversarial, MAE and style loss (orange), and adversarial, MAE, style and content loss (green). Models tested on healthy IXI dataset.

For qualitative comparisons, we now present examples of generated T2 images along with their source T1 and corresponding ground truth T2 images (Figure 12) using the models that were trained on just adversarial losses and models trained on both adversarial and perceptual losses. The models were taken from the end of training, i.e. after being trained for 100 epochs. In addition, we present a "difference image" that helps plots the mean absolute error between the pixels of the generated and respective ground truth image.

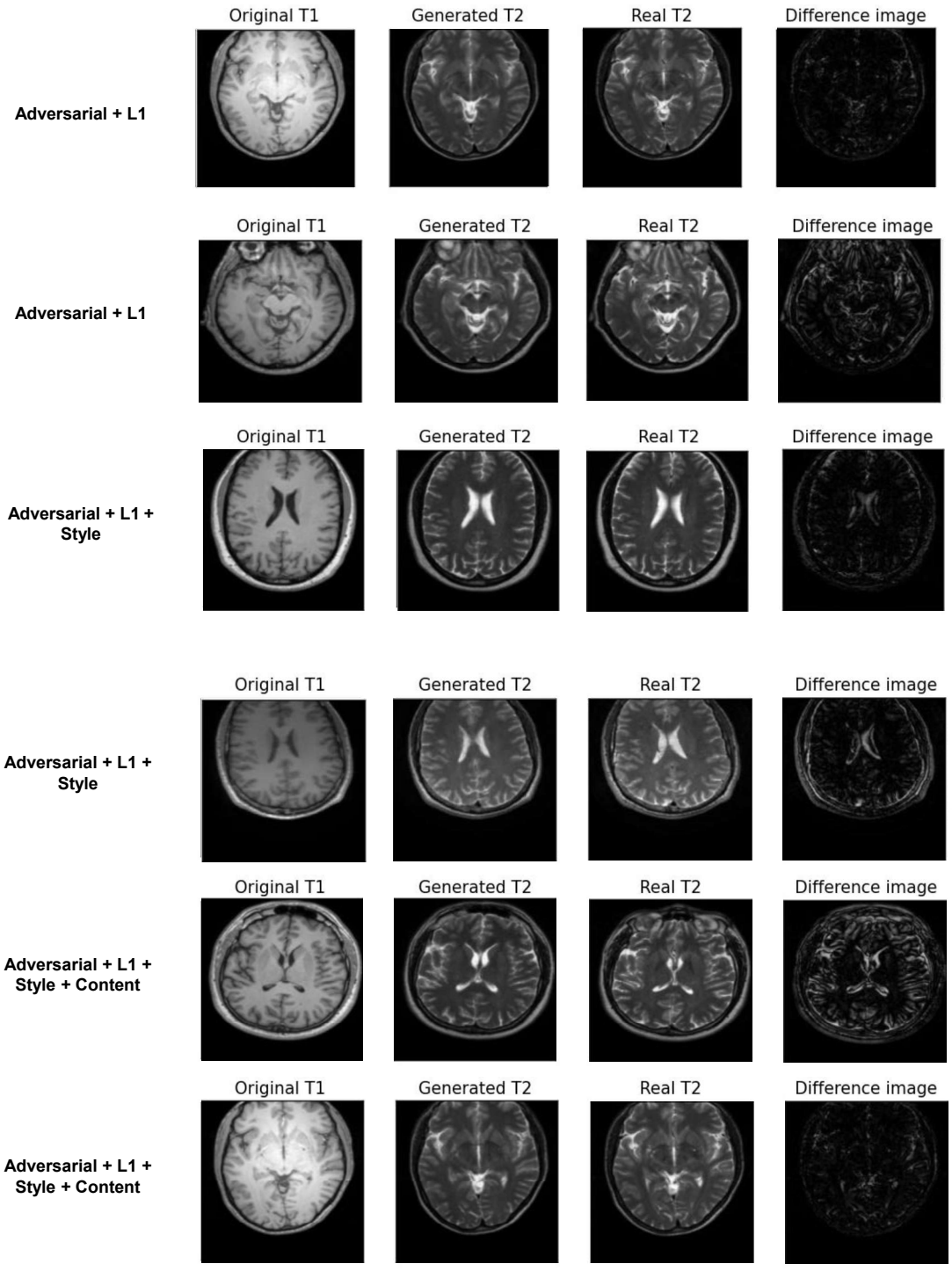


Figure 12: Examples of source T1, generated T2, and corresponding ground truth T2 images for the models trained on adversarial and perceptual losses.

We also notice that even if the brain volumes are different, the algorithm generalizes well between them to produce high quality translations, something that is necessary for real world applications since not all scans may not come from the same MRI machine or even the same imaging institution. Moreover, the algorithm produces realistic translations for scans from various locations in the axial direction. The algorithm is also successful in translating both the global scale features (for example brain volume, color of the ventricles, correct anatomical position of various features) as well as the minute anatomical features (for example, the shape of the ventricles, curvature of the anterior and posterior horns of the ventricles, etc.).

We see that generally the difference image is the brightest in the regions of the skull, something that could be improved by skull stripping the MRI scans before training models on them. Another area of difference is near the boundary between gray and white matter. We also see that, generally, the features such as the white matter, are sharper in the images generated by models trained on perceptual losses in addition to adversarial losses. To further highlight this point, we now compare the translation created by the baseline model and the model with perceptual losses when the same source T1 image is provided (Figure 13).

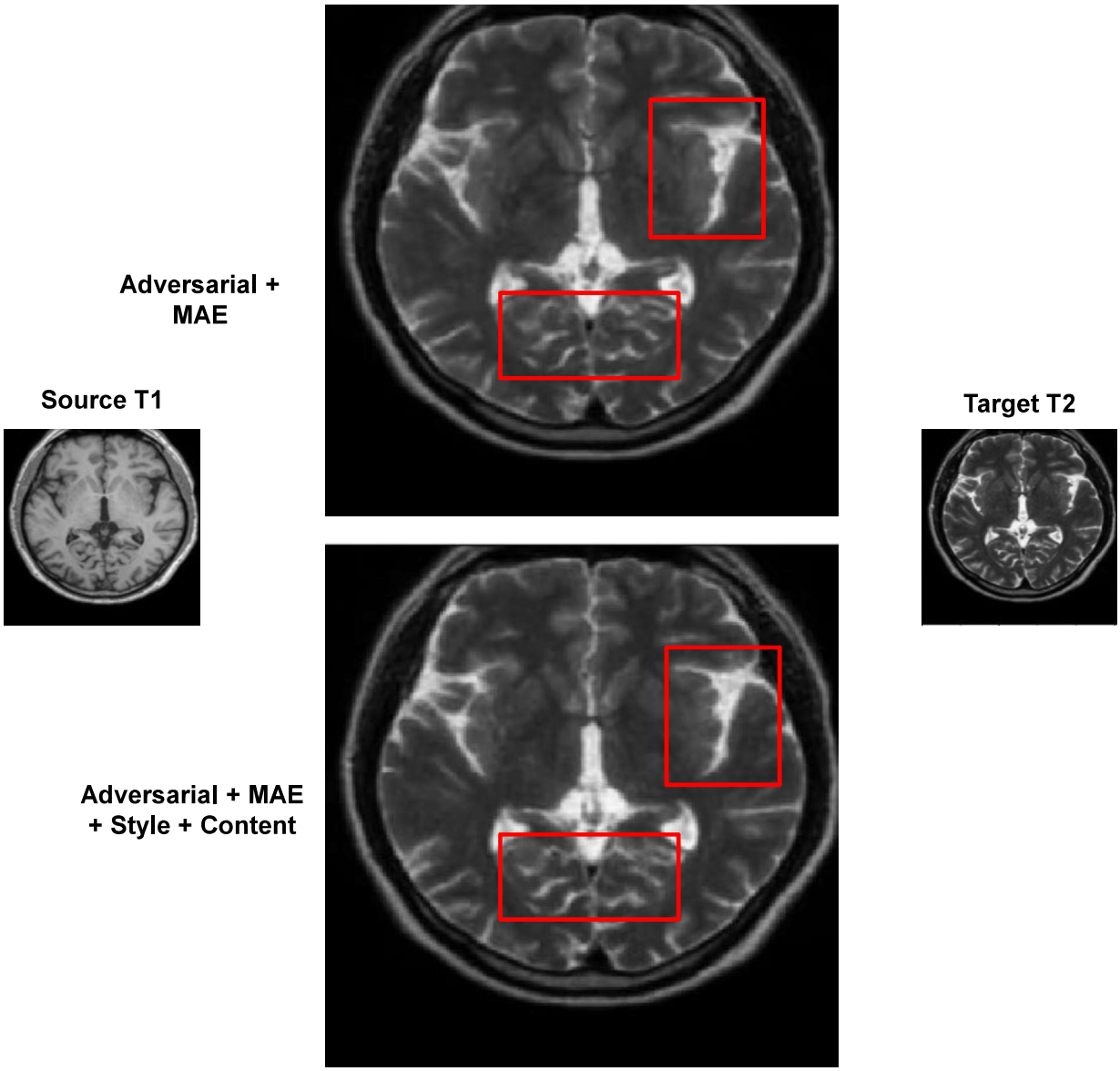


Figure 13: Comparison of the generated T2 image from the baseline model and model trained on adversarial, MAE, style, and content losses, when the same source T1 image is provided. The red boxes show the location in the MR scan where anatomical features are sharper.

With the addition of perceptual losses, we notice that the translated features are now sharper and more distinctive in the generated images (red box in Figure 13). While both the images may look similar to an untrained eye, the enhanced distinctiveness and clarity of the minute anatomical features can potentially help in better diagnosis. The anatomical features created by the

models training on perceptual losses also match very closely with the features seen in the ground truth image (Figure 14). Once again we see how the boundary shape and region of the white matter in the generated image closely correspond with those in the ground truth image. However, one must note that there are still some minute anatomical features missing or blurry in the generated T2. For example, the yellow box in Figure 14 shows a close up of the posterior horns of the ventricles and we see how some of the minute anatomical features (the black spots) are present in the generated T2 but are blurry. Regardless, we believe that this blurriness is less than what we would see if no perceptual losses were used in training. In all, we conclude that addition of perceptual losses improves the quality of translated T2 images.

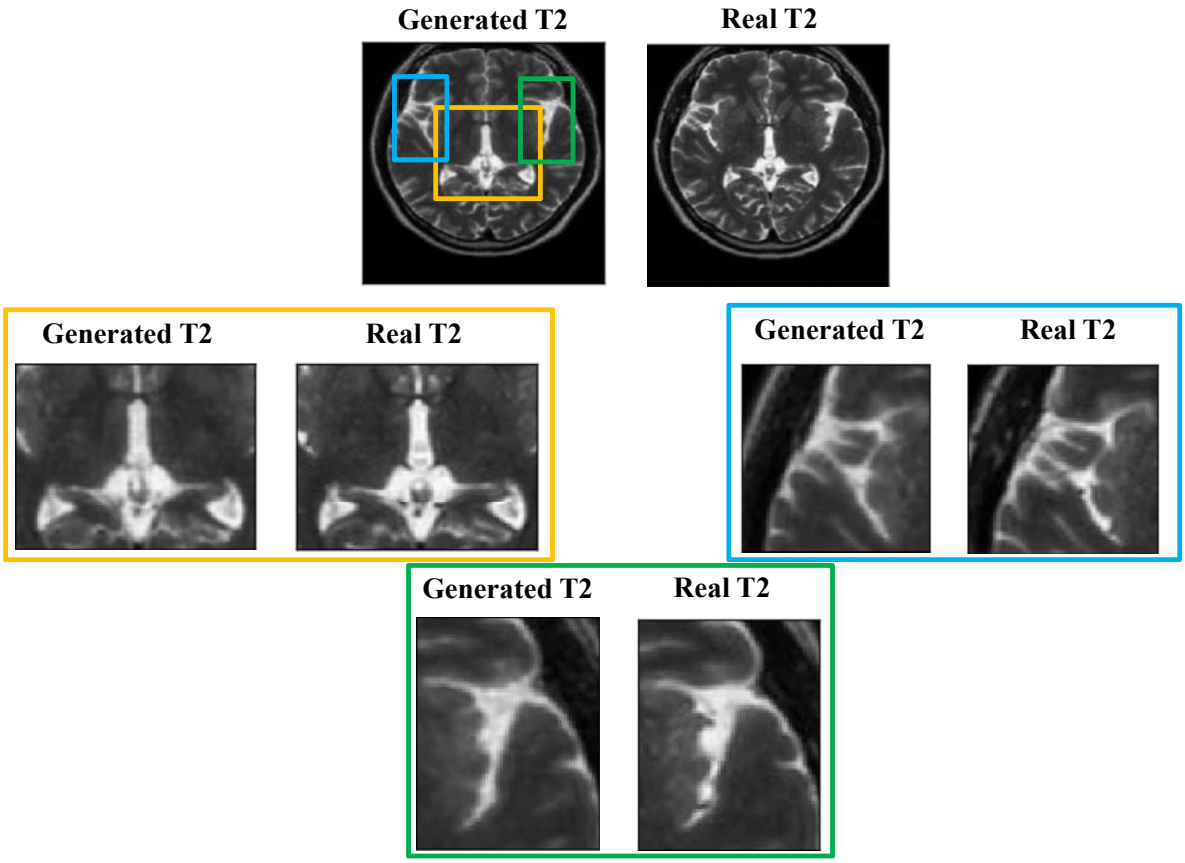


Figure 14: Zoomed in snapshots comparing the different anatomical features in the T2 scan generated by the model training on perceptual losses and the ground truth T2 scan.

One notices that there are some outliers in the metrics presented in Figures 10 and 11. Such outliers, solely determined from metrics, could imply poor translation, which in a clinical or diagnostic setting could lead to catastrophic effects. However, such outliers are not the result of a poor translation, but poor source data. For example, there can be a mismatch between the slices of the source T1 and ground truth T2 images from the testing dataset (Figure 15A) or the resolution of the scan is very poor due to patient movements (Figure 15B).

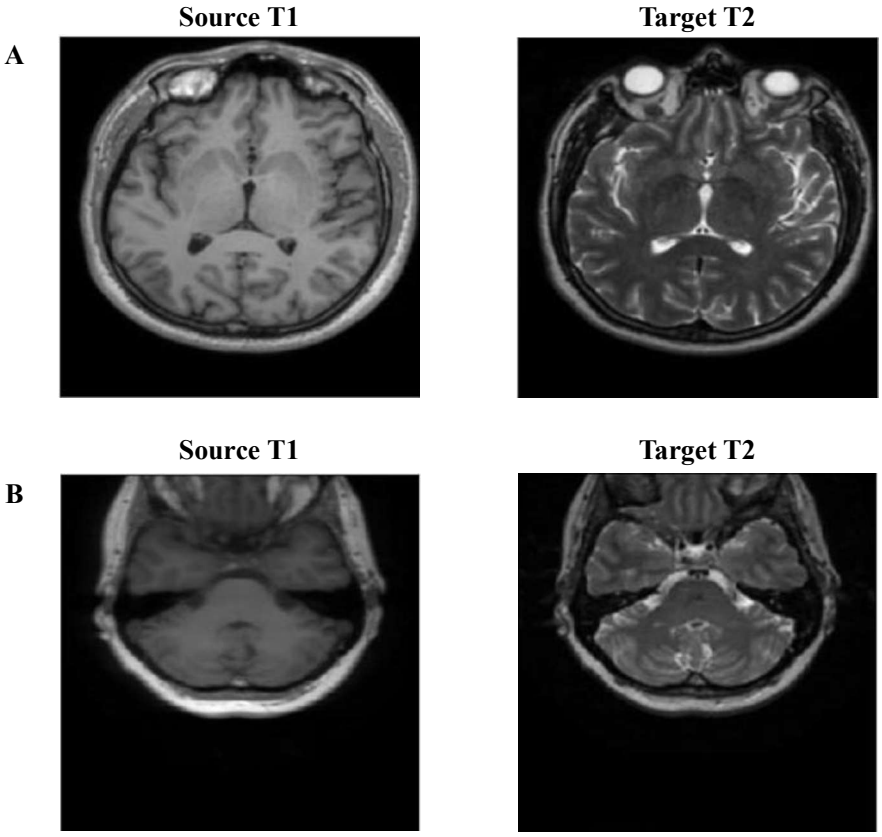


Figure 15: Outliers in translation metrics are not necessarily a result of poor translation but could happen due to (A) a mismatch between the source T1 and ground truth T2 and (B) due to originally blurry source T1 images.

4.4 Model with perceptual losses outperforms baseline model on unhealthy data

In a real world medical setting, an image translation algorithm will have to produce accurate translations for both healthy and unhealthy brain MRI. The unhealthy scans are even more critical since they are one of the primary ways for a doctor to assess the progression of disease. In order to determine if adding perceptual losses to the GAN helps in better image translation, we also tested *pTransGAN* (trained on only healthy data) on an unseen unhealthy dataset. Figure 16 and 16 show how the *pTransGAN,* trained on the different loss function configurations, performed on the unhealthy test dataset.

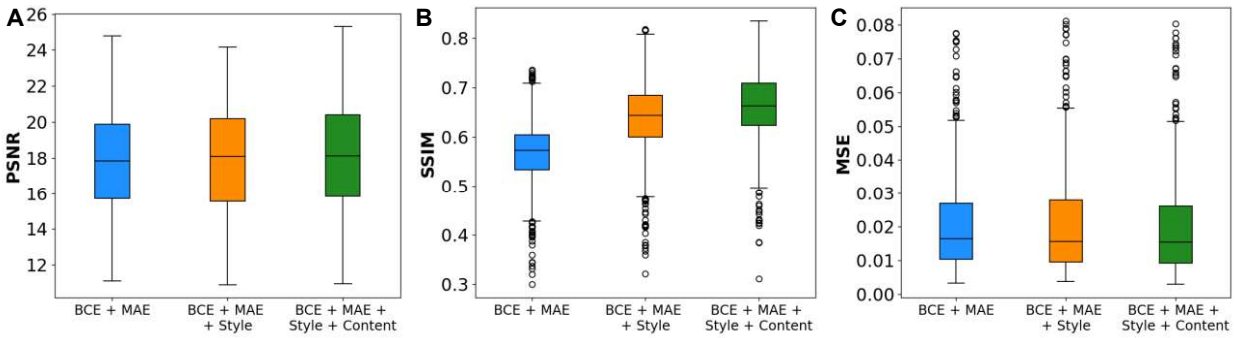


Figure 16: Comparing the novel metrics (A: PSNR, B: SSIM, and C: MSE) for the models that trained on adversarial and MAE loss (blue), adversarial, MAE and style loss (orange), and adversarial, MAE, style and content loss (green). Models tested on unhealthy BRaTS2020 dataset.

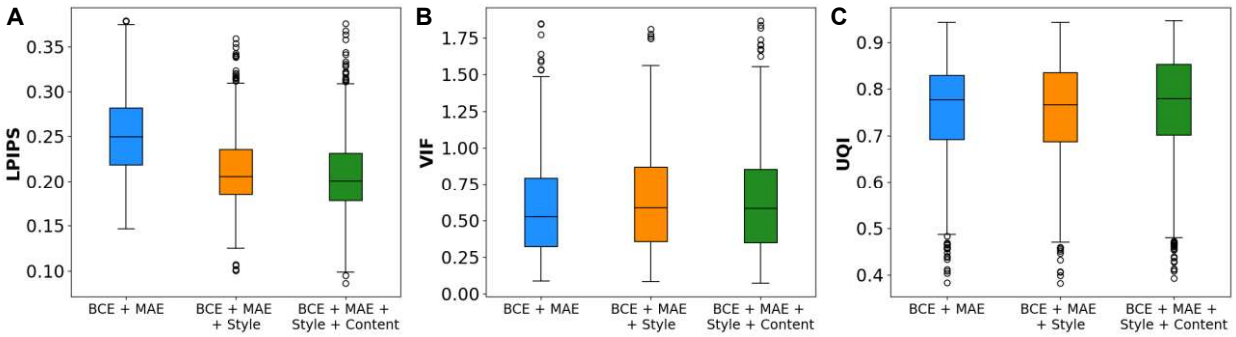


Figure 17: Comparing the novel metrics (A: LPIPS, B: UQI, and C: VIF) for the models that trained on adversarial and MAE loss (blue), adversarial, MAE and style loss (orange), and adversarial, MAE, style and content loss (green). Models tested on unhealthy BRaTS2020 dataset.

When *pTransGAN* is trained with perceptual losses, then it performs significantly better on an unseen unhealthy dataset as measured by both traditional metrics (Figure 16) and novel human perceptual similarity estimating metrics (Figure 17). Performing the Wilcoxon tests on the differences between the metrics from the baseline model and the models trained with perceptual losses also confirmed that the latter outperformed the baseline models in all of the metrics. We believe that this primarily happens because the baseline model creates more significant artifacts during translation as well as is not able to create sharp features like the model trained on perceptual losses can (Figure 18). For example, in the first example presented in Figure 18, both the baseline and perceptual loss models partially recreate the tumor boundary, however the baseline model performs worse on the whole image metrics since it has more artifacts (left side) and the anatomical features (for example the white matter) in that image are blurrier. Nevertheless, the performance of the model with perceptual losses trained on healthy data and tested on unhealthy data is still poor and not suitable for clinical use, which we show how to better in the latter sections.

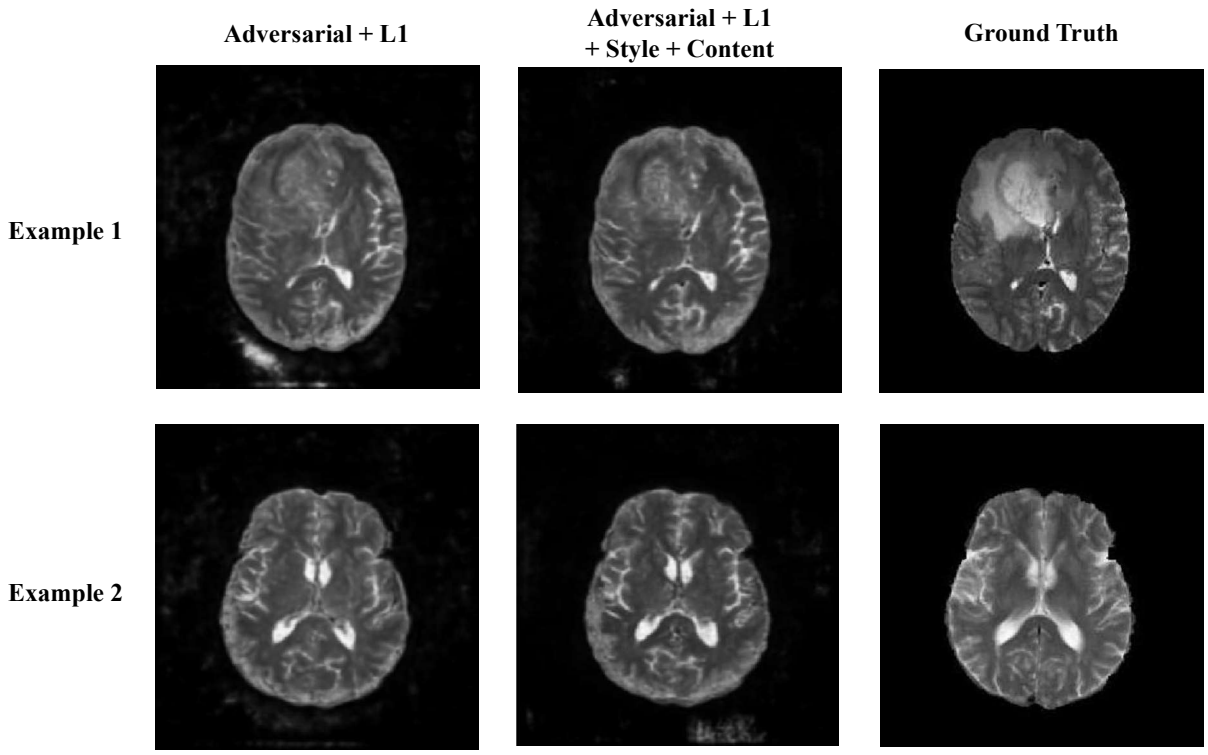


Figure 18: Comparing the translated T2 images (from the baseline model and model trained on perceptual losses) with the ground truth T2 image. Two examples are presented with varying degrees of tumor presence.

4.5 Translation of Unhealthy T1 MRI to T2 scans

Since in the medical world, there could be both healthy and unhealthy T1 scans that need to be translated to T2 scans, and a generative method must be capable of translating both healthy and diseased scans. It is specifically important to get the minute anatomical features of unhealthy scans correct as they are often used in diagnostic purposes.

In section *4.3,* we saw that the models trained solely on the healthy dataset perform better on the unhealthy dataset when perceptual losses such as style and content loss are incorporated. However, *pTransGAN* trained solely on the healthy dataset does not perform as well on the unhealthy dataset as it does on the healthy test dataset. Hence, in order to better the performance of *pTransGAN* on the unhealthy data, we train the model on the BRaTS2020 dataset as outlined in the previous training protocol and evaluate it on both the unhealthy and healthy test data.

For this experiment, we solely work with *pTransGAN* model that trains on both adversarial and perceptual losses since we previously showed that training on perceptual losses can improve the generalizability of the image translation models. The generator architecture had 6 U-blocks and the discriminator was 70x70 receptive field model. We applied the spectral normalization method to the weights of the discriminator to stabilize adversarial training and also increased the learning rate of the discriminator to 0.0008. All other training parameters remained same as discussed in section *3.4.* All the evaluation metrics discussed in section *2.5* were used to characterize the performance on the test datasets (BRaTS2020 and IXI).

4.6 Evaluating pTransGAN on unhealthy dataset after training on unhealthy data

The performance of *pTransGAN,* after it is trained on just the unhealthy dataset, is significantly improved as shown by the metrics in Figures 19 and 20, which also the metrics from *pTransGAN* trained on healthy data but tested on unhealthy data. The paired Wilcoxon tests on the differences between the metrics from these models confirm that the differences are statistically significant (p-value < 0.001) for all six of the metrics.

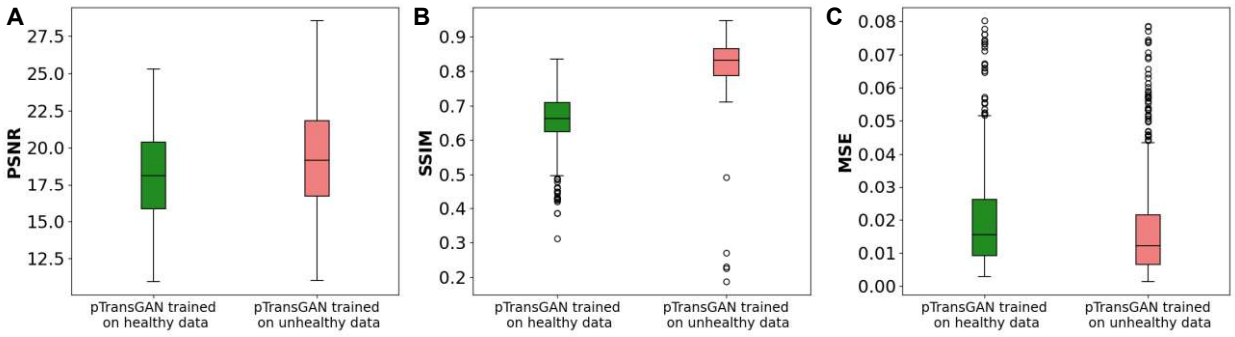


Figure 19: Comparing the traditional metrics (A: PSNR, B: SSIM, C: MSE) for *pTransGAN* models trained on just healthy data (green) and on just unhealthy data (red).

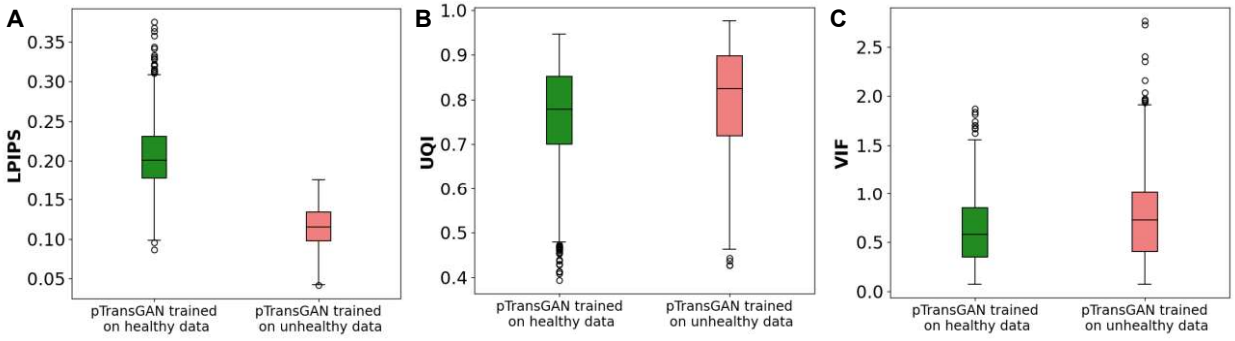


Figure 20: Comparing the novel metrics (A: LPIPS, B: UQI, C: VIF) for *pTransGAN* models trained on just healthy data (green) and on just unhealthy data (red).

pTransGAN when trained and tested on the unhealthy dataset is capable of highlighting brain tumors that can be seen in T2 scans (Figure 21). We see that the tumor is only faintly seen as a darker mass of tissue in the T1 scan but shows up brightly in ground truth T2 scan. *pTransGAN* generated T2 scan also shows this tumor tissue. The model is able to accurately capture the tumor boundary and shape, however it cannot fully capture the brightness of the tumor. This can be seen in the difference image as well as the zoomed in images. However, there are certain instances when the tumor in T1 scan does not manifest as a completely dark mass (Figure 22). In such a situation, our model is able to identify the mass as a tumor and translate it into a T2 scan but is not able to fully capture the minute anatomical features as well as the shape of the tumor (Figure 22).

The *pTransGAN* model trains on perceptual losses like style and content loss in addition to the adversarial and MAE losses. Due to this, *pTransGAN* is able to sometimes able to create more accurate representations of tumors in T2 scans than in the ground truth images (Figure 23). This could happen as a result of the data collection protocol. For example, if there is patient head movement during the MRI acquisition phase, the scan become blurry; there might be no movement during T1, leading to a sharp T1 source scan, but there could be motion during T2 scan time, causing blurry T2 scan. With the use of *pTransGAN,* such blurriness and image artifacts can be reduced. For example, in Figure 23, we see that the tumor boundary is hazy in the T2 scan, however the boundary predicted by the model is sharp, thus better presenting the spread of the tumor in the brain.

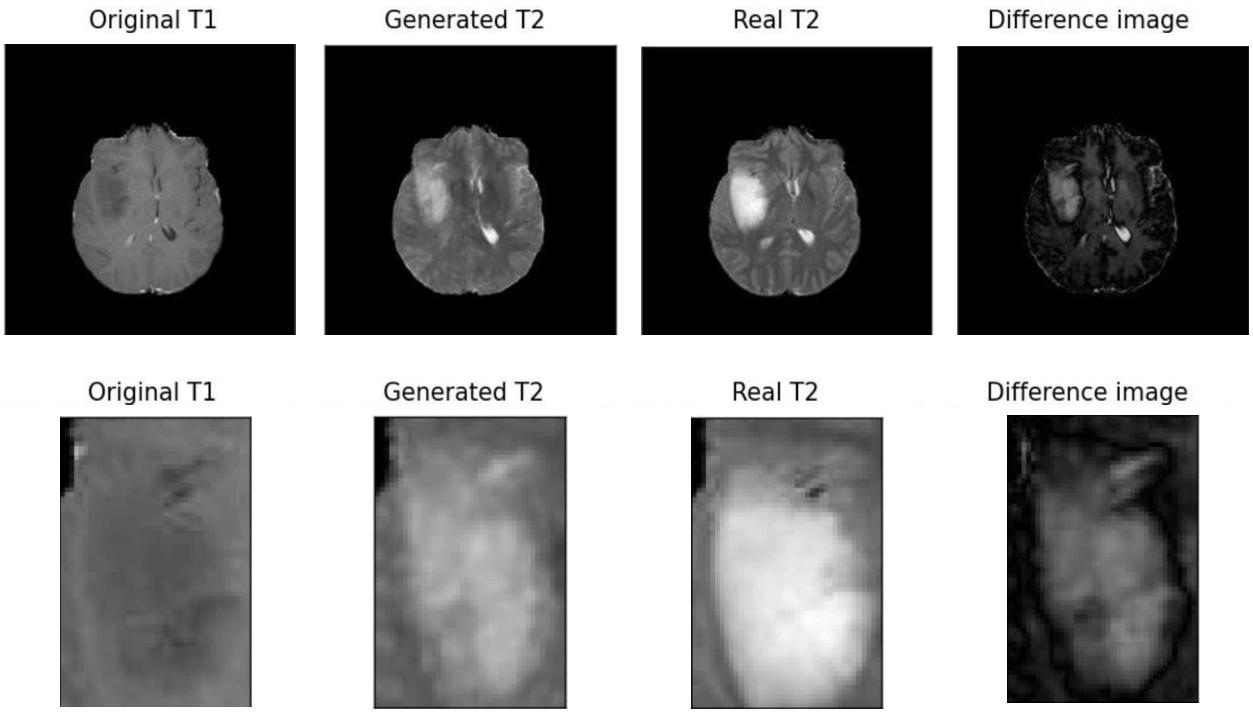


Figure 21: *pTransGAN* trained on unhealthy data is capable of translating brain tumors which do not clearly show up in the T1 scans but are seen as bright masses of tissue in T2 scans. The zoomed in Figure shows how *pTransGAN* is capable of accurately capturing the tumor boundary however does not fully show the brightness of the tumor tissue.

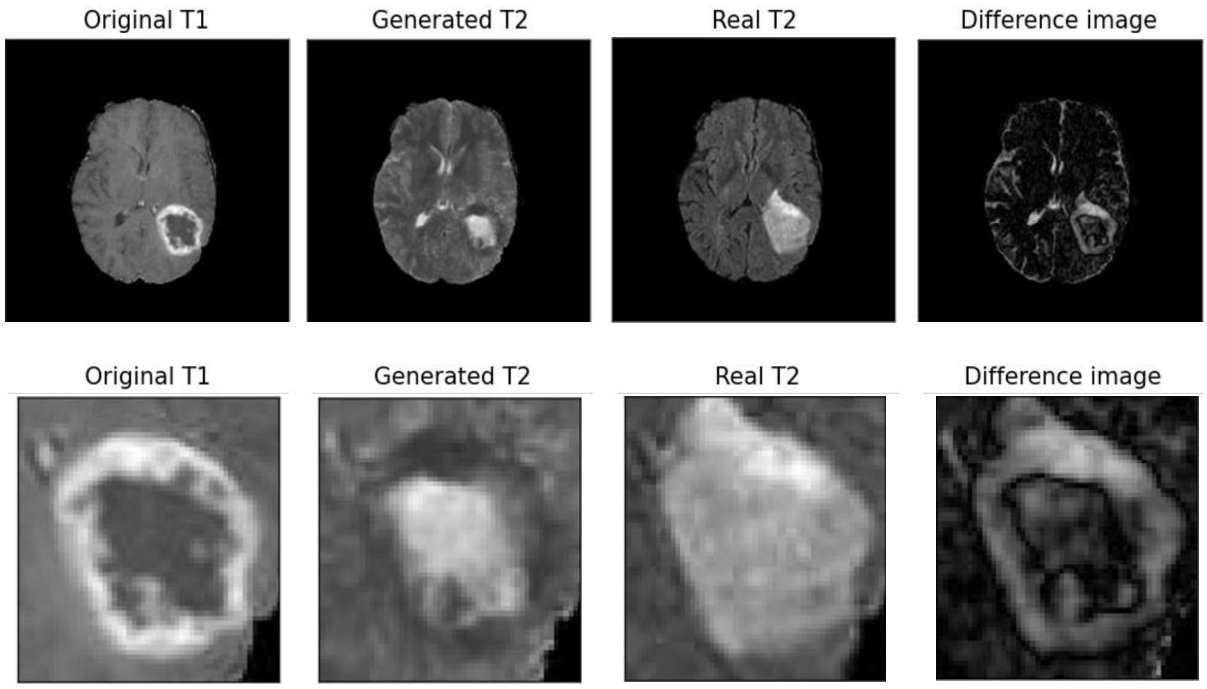


Figure 22: *pTransGAN* is capable of translating the global features of brain tumors which show up as a combination of dark and bright tissues, however it misses the minute anatomical details, which is shown by the zoomed in Figure.

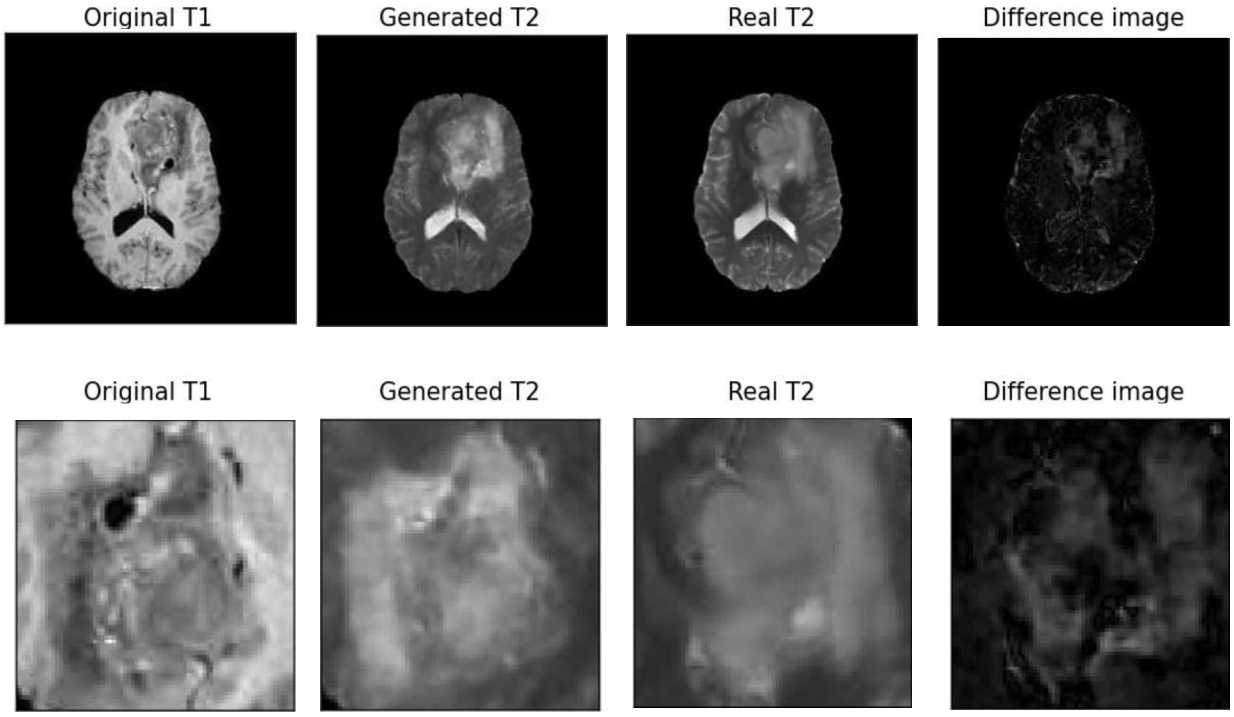


Figure 23: *pTransGAN* is capable of producing sharper T2 scans when the ground truth scans show blurriness. The zoomed in image shows how the boundary of the tumor is sharpers in the model prediction.

However, if we use the healthy dataset to test the *pTransGAN* model trained on unhealthy dataset, we see a significant drop in the quality of translation, as shown by Figures 24 and 25. This is undesirable in a real world clinical setting because if the model receives a healthy image, giving a poor translation or introducing artifacts should significantly hamper its usability. We see that the healthy images produced by *pTransGAN* trained on unhealthy data are generally very blurry (Figure 26) and are not able to capture high frequency features. The model trained on unhealthy data cannot accurately translate the boundary of ventricles and the white matter. However, the global features, like the shape of the brain and location of large anatomical features, is still achieved.

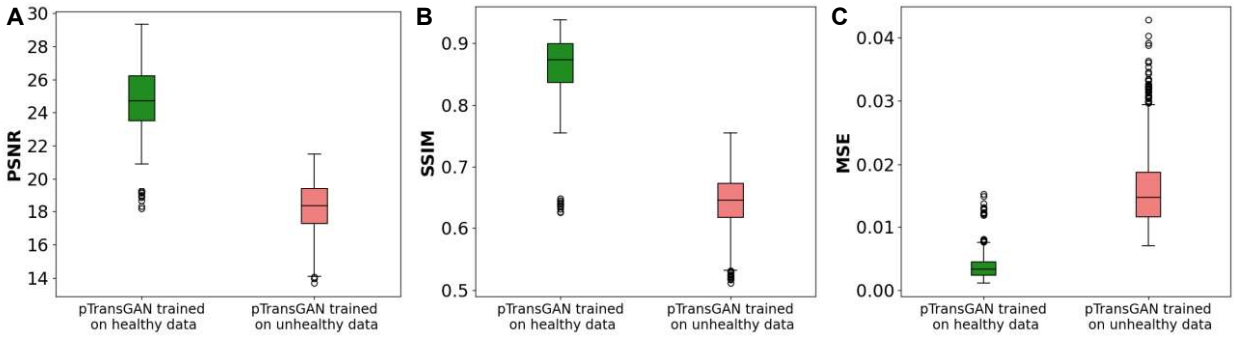


Figure 24: A significant drop in PSNR (A) and SSIM (B) whereas an increase in MSE (C) is seen when *pTransGAN* trained on unhealthy dataset is tested on the healthy dataset.

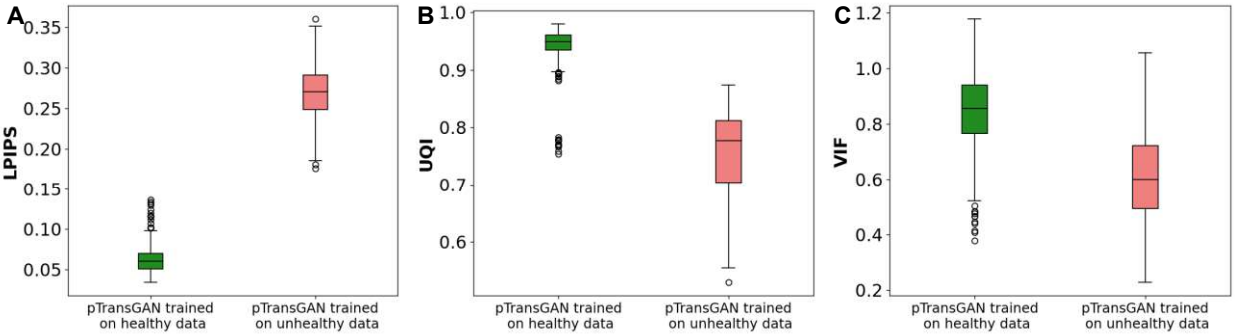


Figure 25: A significant increases in LPIPS (A) and drops in UQI (B) and VIF (C) is seen when *pTransGAN* trained on unhealthy dataset is tested on the healthy dataset.

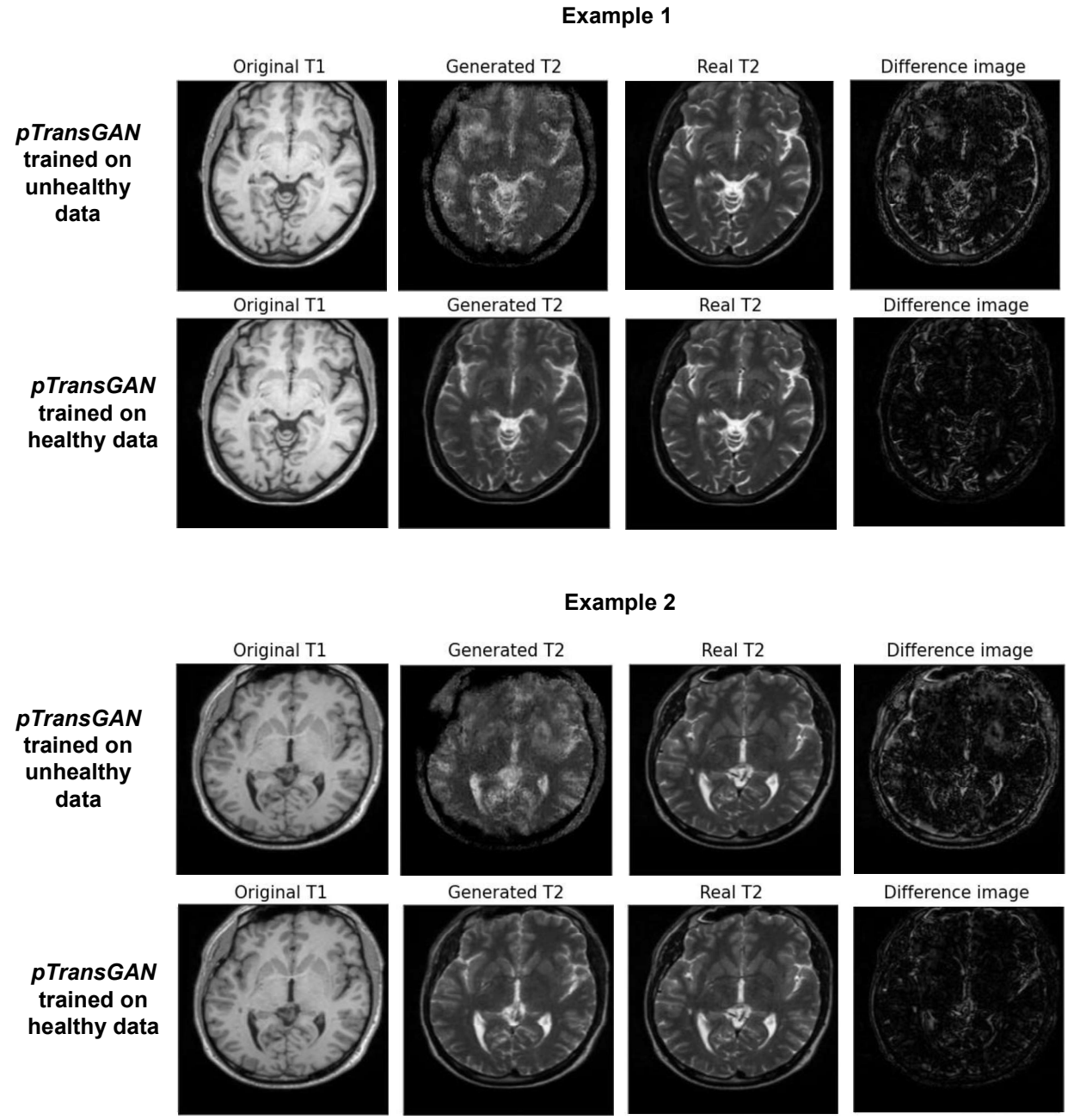


Figure 26: Two examples of healthy T1 scans translated by *pTransGAN* trained on unhealthy datasets. In both the examples, minute features are not accurately translated, and boundaries are blurry. However, global features are translated accurately to a great extent.

4.7 Creating a Single Model for Healthy and Unhealthy MRI

In the previous sections, we saw that when *pTransGAN* is trained on healthy data, it gives excellent translation results for healthy testing data, but not satisfactory performance on unhealthy data. Conversely, when *pTransGAN* is trained on unhealthy data, the performance on unhealthy test data increases, but the model's ability to translate healthy images decreases significantly. In a real world setting, *pTransGAN* needs to successfully transfer both healthy and unhealthy images, so in this section we explore and compare two training protocols that enable *pTransGAN* to train on both healthy and unhealthy datasets. The experiments we ran are as follows:

1. Sequential training: train *pTransGAN* on healthy dataset and then train the same model on unhealthy dataset (training protocol in Table 5)
2. Simultaneous training: for every image, train the *pTransGAN model* for three times on a healthy image and then train the same model three times on an unhealthy image (training protocol in Table 6)

For both of these experiments, we used the 6 U-block variant of *pTransGAN* and the 70x70 receptive field discriminator, whose weights are stabilized by spectral normalization. We also increased the learning rate of the discriminator to 0.0008 and maintained the generator's learning rate at 0.0002 (Appendix B). In both experiments, the models were trained for 100 epochs. Since the healthy training data was smaller than unhealthy training data, we truncated the unhealthy dataset by 75 images. All other training parameters remain the same as previously discussed. To evaluate these models, the healthy and unhealthy testing datasets were used and the six metrics (PSNR, SSIM, MSE, LPIPS, UQI, and VIF) were calculated.

Table 3: Sequential training protocol used to train pTransGAN on both healthy and unhealthy datasets.

Sequential Training protocol for *pTransGAN*
• Load paired healthy training dataset$\{(x_j, y_j)\}_{j=1}^{N_{images}}$ where N is the size of dataset • Set $N_{epochs} = 100$ • Load pre-trained VGG-19 feature extractor weight ImageNet weights • Initialize generator and discriminator model weights using Xavier initializer
for epoch in N_{epochs} **do**: **for** image in N_{images} **do**: • Train discriminator on a pair of real T1 and real T2 • Train discriminator on a pair of real T1 and generated T2 • Calculate $\mathcal{L}_{cGAN}, \mathcal{L}_{L1}, \mathcal{L}_{style}, \mathcal{L}_{content}$ • Backpropagation to update discriminator and generator weights **end for** **end for**
• Load paired unhealthy training dataset$\{(x_j, y_j)\}_{j=1}^{P_{images}}$ where P is the size of dataset • Set $N_{epochs} = 100$ • Load pre-trained VGG-19 feature extractor weight ImageNet weights • Load *pTransGAN* trained on healthy dataset
for epoch in N_{epochs} **do**: **for** image in P_{images} **do**: • Train discriminator on a pair of real T1 and real T2 • Train discriminator on a pair of real T1 and generated T2 • Calculate $\mathcal{L}_{cGAN}, \mathcal{L}_{L1}, \mathcal{L}_{style}, \mathcal{L}_{content}$ • Backpropagation to update discriminator and generator weights **end for** **end for**

Table 4: Simultaneous training protocol used to train pTransGAN on both healthy and unhealthy datasets.

Simultaneous Training protocol for *pTransGAN*
• Load paired healthy training dataset$\{(x_j, y_j)\}_{j=1}^{N_{images}}$ where N is the size of healthy dataset • Load paired healthy training dataset$\{(x_j, y_j)\}_{j=1}^{N_{images}}$ where N is the size of unhealthy dataset • Set $N_{epochs} = 100$ • Load pre-trained VGG-19 feature extractor weight ImageNet weights • Initialize generator and discriminator model weights using Xavier initializer • Initialize number of simultaneous iterations, N_G, to 3
for epoch in N_{epochs} **do**: **for** image in N_{images} **do**: **for** iteration in N_G **do:** • Train discriminator on a pair of healthy real T1 and real T2 • Train discriminator on a pair of healthy real T1 and generated T2 • Calculate $\mathcal{L}_{cGAN}, \mathcal{L}_{L1}, \mathcal{L}_{style}, \mathcal{L}_{content}$ • Backpropagation to update discriminator and generator weights **for** iteration in N_G **do:** • Train discriminator on a pair of unhealthy real T1 and real T2 • Train discriminator on a pair of unhealthy real T1 and generated T2 • Calculate $\mathcal{L}_{cGAN}, \mathcal{L}_{L1}, \mathcal{L}_{style}, \mathcal{L}_{content}$ • Backpropagation to update discriminator and generator weights **end for** **end for** **end for**

4.8. Comparing Simultaneous and Sequential Learning Protocols

When *pTransGAN* is trained sequentially, first on the healthy dataset and then on unhealthy dataset, we see that the model catastrophically fails when tested on the healthy dataset (the first task). However, on the unhealthy test dataset, *pTransGAN* performs similar to the model trained on unhealthy data (Figure 27, 28). In other words, it seems like *pTransGAN* "forgot" what it learned from the first task of training on the healthy data. This is consistent with previous literature [37] showing that models trained sequentially on different tasks tend to "forget" what they learned from the previous tasks. We see that both traditional and novel metrics for the sequential learning model are worse than the model trained solely on the healthy data and the simultaneous training model, which result in blurry images and missing structures (Figure 29). However, we see that the simultaneous training model performs approximately similar to the model trained on just healthy data. The difference between metrics from simultaneous training and training on just healthy data are statistically significant for just PSNR and SSIM (p-value < 0.001). However, on the novel metrics, the difference is not statistically different, indicating that for a human observer images generated by the simultaneous training model will be perceptually similar to those from the model trained on just healthy data.

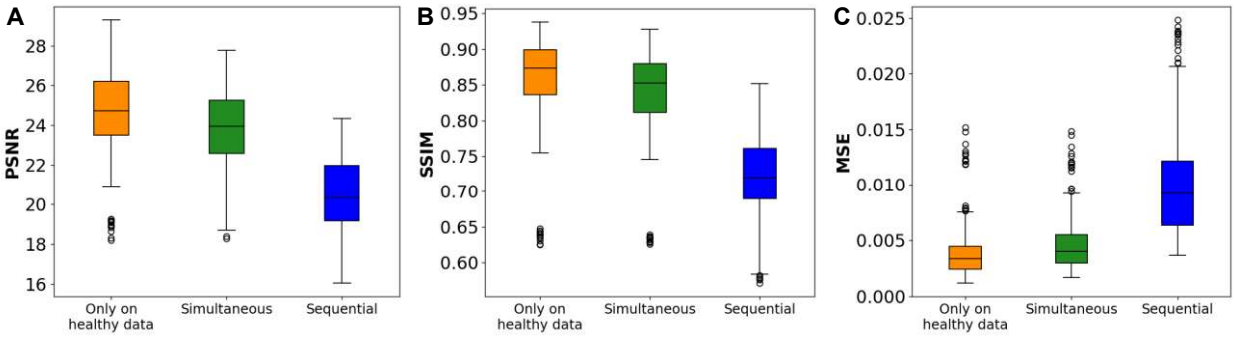


Figure 27: Traditional metrics when pTransGAN, trained in a simultaneous and sequential fashion on both healthy and unhealthy, is tested on the healthy dataset. For comparison, we also present the metrics from pTransGAN trained and tested just on the healthy data.

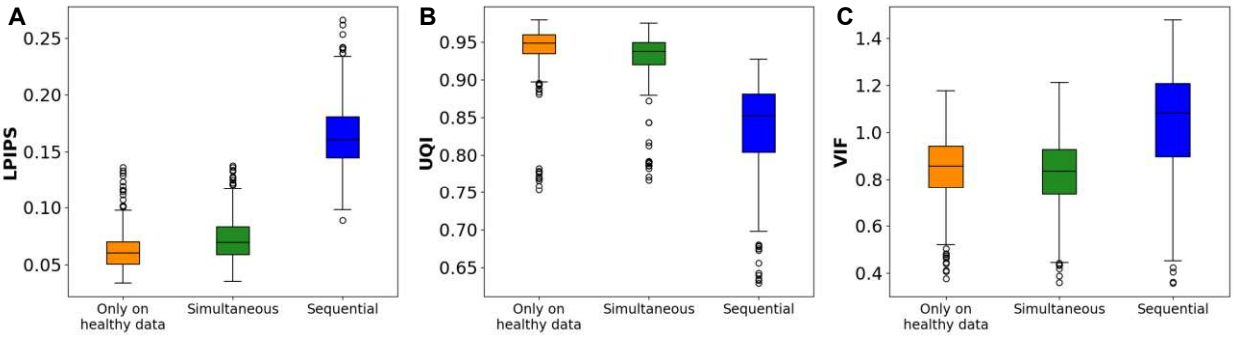


Figure 28: Novel metrics when pTransGAN, trained in a simultaneous and sequential fashion on both healthy and unhealthy, is tested on the healthy dataset. For comparison, we also present the metrics from pTransGAN trained and tested just on the healthy data.

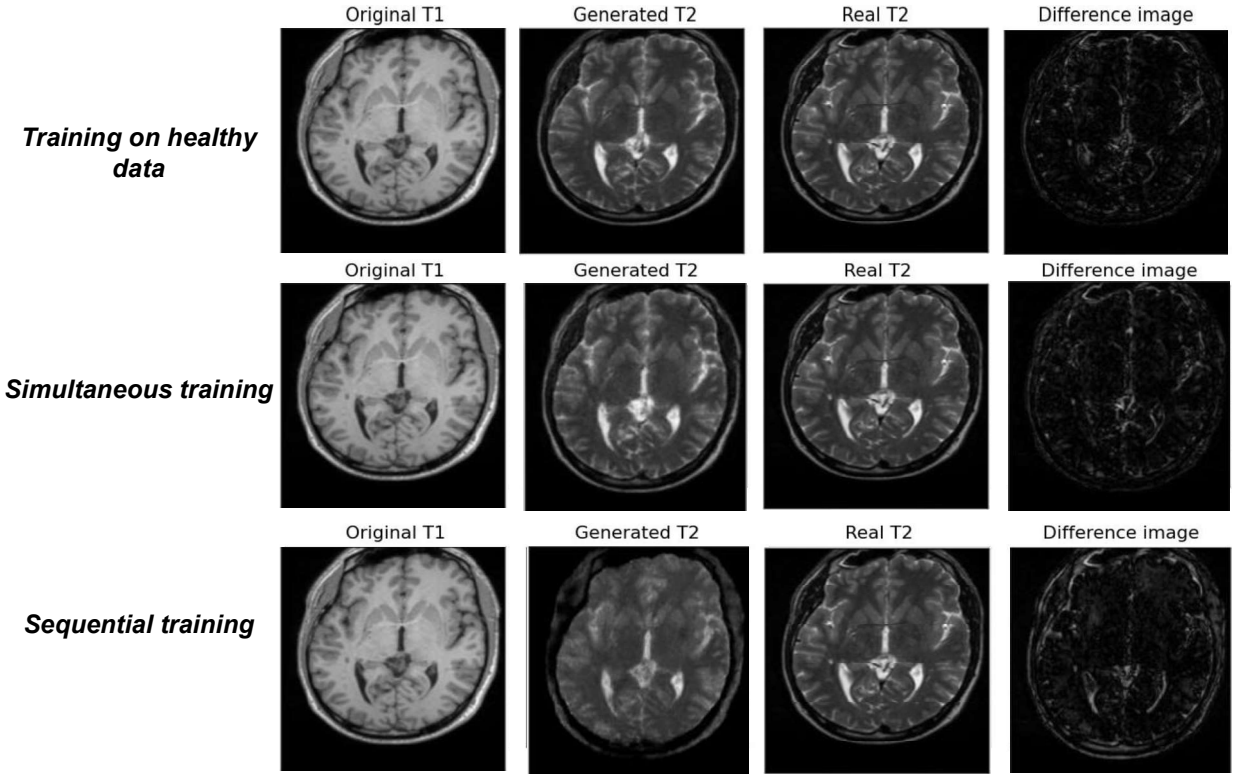


Figure 29: Comparing the translation of a healthy T1 MRI by the three training protocols: training on just healthy data, simultaneous training, and sequential training, which produces the worst results.

When *pTransGAN* trained via sequential and simultaneous training protocols is tested on the unhealthy dataset, we see that sequential trained model performs equally as well as the model trained solely on unhealthy data (Figure 30, 31). However, the simultaneous training model outperforms the sequentially trained model and the model trained on unhealthy data in PSNR, UQI, and MSE, which shows that the knowledge from training on healthy data helped the model perform better on unhealthy data (Figure 30, 31).

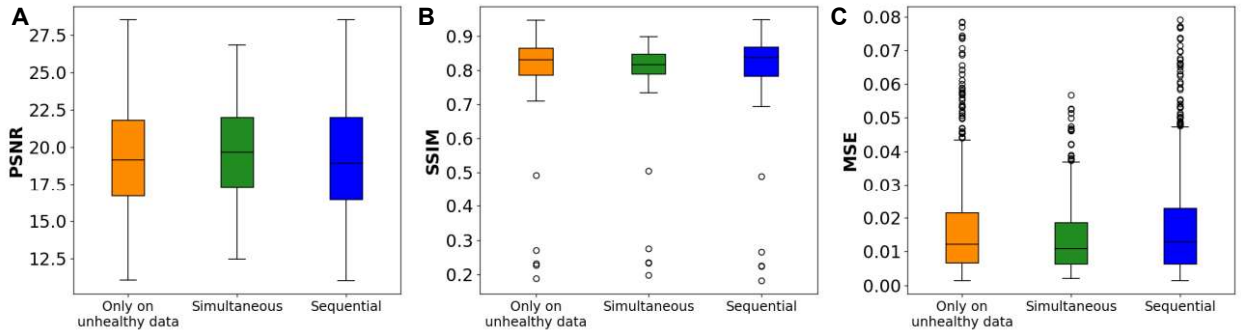


Figure 30: Traditional metrics when pTransGAN, trained in a simultaneous and sequential fashion on both healthy and unhealthy, is tested on the unhealthy dataset. For comparison, we also present the metrics from pTransGAN trained and tested just on the unhealthy data.

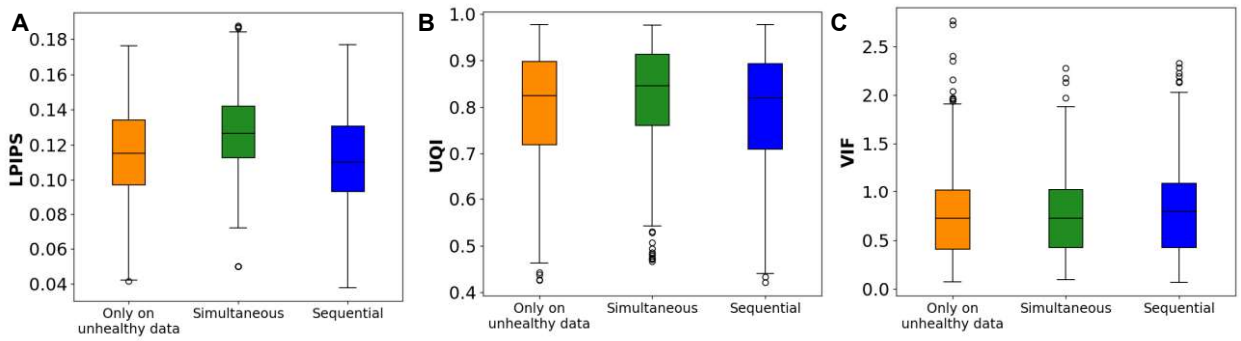


Figure 31: Novel metrics when pTransGAN, trained in a simultaneous and sequential fashion on both healthy and unhealthy, is tested on the unhealthy dataset. For comparison, we also present the metrics from pTransGAN trained and tested just on the unhealthy data.

Chapter 5: Conclusion and Discussion

In this study, we present an end-to-end conditional generative adversarial framework, *pTransGAN,* that is capable of translating both healthy and unhealthy T1 brain MRI into T2 MRI with high fidelity. In addition to using adversarial losses for training, our framework leverages non-adversarial perceptual losses, specifically style and content losses, to create sharper translated images. To progressively refine the translated image, the generator of the *pTransGAN* model uses 6 coupled U-blocks (which can be thought of as encoder-decoder pairs with residual connections). Overall, *pTransGAN* was able to translate global structures and accurate minute anatomical features between the T1 and T2 domains. *pTransGAN* with perceptual losses was tested on both healthy and unhealthy data, and a simultaneous training protocol was designed so that a single model could perform well on both healthy and unhealthy MR scans.

The *pTransGAN* generator not only trains on the adversarial losses but also on two non-adversarial perceptual losses. We show that the model performance on the healthy dataset quantitatively increases with the addition of perceptual losses (Figures 10 and 11). Specifically, *pTransGAN* trained using perceptual losses outperforms the baseline model in the SSIM and LPIPS metrics, whereas the other metrics are not significantly different between the two models. However, when comparing the translated images from the two models (Figure 13), we see that T2 scan created by *pTransGAN* trained on perceptual losses has sharper edges. For example, the distinction between the white and gray matters is more distinct when perceptual losses are used, and minute anatomical features are more distinctive (for example compare the falx cerebri in Figure 13). Such perceptual differences that are visible to the human observer are not captured by the quantitative metrics. Given that *pTransGAN* with perceptual losses outperforms the baseline

model in both quantitative and qualitative inspections, we conclude that the addition of perceptual losses produces translations that are globally homogenous and also accurate on minute anatomical features. This being said, there is still room for improvement as *pTransGAN* is not able to maintain the shape of some very minute anatomical features (Figure 14). The features are not missed, but their global structure is not maintained to a high degree of accuracy. Such an issue may not be related to the model architecture but could arise due to the limitedness of the training data. Such an issue could be solved by diversifying the training dataset (for example, T1 scans could be acquired from different population groups or from different MRI machines).

Since the image translation community is significantly used to quantitative metrics to compare results of different models, these metrics are included in our study. The performance of *pTransGAN* with perceptual losses compares well with that of previous literature [11] in terms of PSNR. However, the SSIM of *pTransGAN* is slightly lower than that presented in [11]. This could have happened since the IXI dataset was registered in [11], which was not done in this study. Nevertheless, as shown previously, the traditional metrics generally cannot capture the perceptual differences between images. Previous literature in T1-T2 brain MRI translation has not used the novel metrics (i.e. LPIPS, UQI, and VIF). However, we show that when using perceptual losses, the novel metrics are improved as well, especially the LPIPS loss, which has been shown to very closely model human perceptual similarity.

In addition to creating sharper translated images, the generalizability of *pTransGAN* is significantly improved with the addition of perceptual losses. When *pTransGAN* with perceptual losses is trained on healthy data but tested on unhealthy it significantly outperforms the baseline model (Figure 17). This shows that with the addition of perceptual losses, the *pTransGAN* model is able to learn more about brain MRI scans, which may explain why it performs better on the

unseen unhealthy data. However, the performance on the unhealthy data for the trained on healthy data is not close to the literature standards [11] and the translated images have artifacts in the background (Figure 18). This significantly limits the effectiveness of *pTransGAN* because in clinical situations, such an algorithm needs to translate both healthy and unhealthy scans.

pTransGAN trained on unhealthy data with perceptual losses included not only achieves excellent global translation results on unhealthy data, but also produces homogenous translations of details in unhealthy brain T1 MRI (Figure 21). For example, *pTransGAN* is able to accurately translate the tumor tissue between the T1 and T2 modalities. We show that the model is not only able to maintain global properties of the tumor, such as the tumor boundary and texture, but also accurately the texture of the unhealthy tissue. While translating the tumorous tissue, *pTransGAN* is also able to maintain high fidelity in the healthy regions. However, *pTransGAN* is not yet ready for diagnostic purposes. When there are multiple contrasts in the tumor, *pTransGAN* is not able to translate the unhealthy scans (Figure 22). Even in such an intricate situation, *pTransGAN* is able to maintain the global properties of the tumor and also accurately translate the healthy parts of the scan. We also show that *pTransGAN* can create sharper translated images than the ground truth T2 scan when there are motion artifacts in the actual T2 scan (Figure 24). Previously, it has been shown that an architecture similar to *pTransGAN* is capable of correcting MR motion artifacts [9].

When the *pTransGAN* model trained on unhealthy data and tested on healthy data, the performance drops significantly. Similarly, *pTransGAN* trained on healthy data and tested on unhealthy data shows poor translation capabilities. To generate a single model that can perform well on both healthy and unhealthy data we tested two training protocols (sequential training and simultaneous training) without making any changes to the model architecture. When *pTransGAN* was sequentially trained on healthy and then the unhealthy data, we saw that the performance of

the model significantly worsened on the healthy data (the first learning task) but remained high on the unhealthy data (the second learning task). The model seems to have "forgotten" what it had learned on the first task, which has been known to be a problem with the sequential learning approach [37]. However, when the model is trained simultaneously with the healthy and unhealthy, the model performs well on both the datasets. Moreover, in some metrics, the simultaneous learning approach outperforms the models trained on just healthy or unhealthy data (Figure 30 and 31). A potential limitation of this approach is that since the model has to train three times on each healthy and unhealthy image in every epoch, the training can be very time consuming (~5600 seconds per epoch). However, this limitation can be solved by using multiple GPUs for faster training and testing times.

However, the *pTransGAN* model is not free from limitations, with improvements necessary for the model to be used in a clinical setting. Currently, *pTransGAN* is built to be used with 2D medical images, which is computationally efficient for running experiments on loss functions, stabilizing training, and finding protocols to create a single model for both healthy and unhealthy data. However, much brain imaging data is 3D, and the volumetric information is critical for various medical tasks. We aim to adapt *pTransGAN* to volumetric data as well as multi-channel inputs. To make *pTransGAN* suitable for diagnostic purposes, we also aim to increase the number of epochs for training as well as train on more diverse datasets. In previous studies, registration of images across the datasets being used as well as skull-stripping have been shown to improve translation results, especially the PSNR, SSIM, and MSE metrics [11]. The authors believe that the results from simultaneous training protocol would further improve if both the IXI and BRaTS2020 datasets were registered against a common brain MRI mask, for example the MNI average brain mask [38].

There are a number of future directions for this work on image translation, from learning on unpaired data to translating between alternative imaging modalities. A Majority of the available medical data is unpaired, i.e. the T1 and T2 scans come from different patients. For example, Facebook and New York University have curated a dataset of 6.970 scans of unpaired brain MRI. In recent years, rapid progress has also been made in developing novel architectures that optimize translation with unpaired data. For example, the cycleGAN architecture introduced by [14] and variational encoders created by [39] are some of the algorithms that leverage unpaired data to achieve highly accurate MR translations. Another interesting question is how can we best use both unpaired and paired data for image translation. We aim to explore how the large number of unpaired images can be used to achieve translation on a global scale and use the limited set of paired images to learn to translate the minute anatomical features. Previously, such semi-supervised approaches have been used to simultaneously train on paired and unpaired datasets to achieve superior translation results [14, 15]. Finally, in the medical imaging domain, MRI is only one of the many imaging modalities. In a large number of medical examinations, more than one medical imaging modality is used to get a more holistic view of the patient. Thus, while improving translation of MR scans, we also hope to make contributions in achieving translation across imaging domains, for example if CT scans can be translated into their corresponding MRI scans.

Chapter 6: References and Appendices

References

[1] OECD, "Number of magnetic resonance imaging (MRI) units in selected countries as of 2019," *Statista*, 2020. .

[2] F. Knoll *et al.*, "fastMRI: A Publicly Available Raw k-Space and DICOM Dataset of Knee Images for Accelerated MR Image Reconstruction Using Machine Learning," *Radiol. Artif. Intell.*, vol. 2, no. 1, p. e190007, 2020.

[3] A. M. Mills, A. S. Raja, M. G. Hospital, B. Imaging, and J. R. Marin, "HHS Public Access," vol. 22, no. 5, pp. 625–631, 2016.

[4] M. J. Goske *et al.*, "The ' Image Gently ' campaign : Increasing CT radiation dose awareness The ' Image Gently ' campaign : increasing CT radiation dose awareness through a national education and awareness program," no. June 2014, 2008.

[5] J. A. Brink and E. S. Amis, "Image Wisely : A Campaign to Increase Awareness about Adult Radiation Protection 1 n EDITORIAL," vol. 257, no. 3, 2010.

[6] T. Slovis, "The ALARA concept in pediatric CT: myth or reality?," *Radiology*, vol. 223, pp. 5–6, 2002.

[7] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5967–5976, 2017.

[8] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using

Cycle-Consistent Adversarial Networks," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 2242–2251, 2017.

[9] K. Armanious *et al.*, "MedGAN: Medical image translation using GANs," *Comput. Med. Imaging Graph.*, vol. 79, pp. 1–16, 2020.

[10] R. Gupta, A. Sharma, and A. Kumar, "Super-Resolution using GANs for Medical Imaging," *Procedia Comput. Sci.*, vol. 173, no. 2019, pp. 28–35, 2020.

[11] S. Ul *et al.*, "Image Synthesis in Multi-Contrast MRI with Conditional Generative Adversarial Networks," *IEEE Trans. Med. Imaging*, vol. 90, no. 312, pp. 1–1, 2019.

[12] I. J. Goodfellow, J. Pouget-abadie, M. Mirza, B. Xu, and D. Warde-farley, "Generative Adversarial Nets," pp. 1–9.

[13] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9906 LNCS, pp. 694–711, 2016.

[14] H. Nguyen, S. Luo, and F. Ramos, *Semi-supervised Learning Approach to Generate Neuroimaging Modalities with Adversarial Training*, vol. 12085 LNAI. Springer International Publishing, 2020.

[15] A. Madani, M. Moradi, A. Karargyris, and T. Syeda-Mahmood, "Semi-supervised learning with generative adversarial networks for chest X-ray classification with ability of data domain adaptation," *Proc. - Int. Symp. Biomed. Imaging*, vol. 2018-April, no. Isbi, pp. 1038–1042, 2018.

[16] A. Lahiri, K. Ayush, P. K. Biswas, and P. Mitra, "Generative Adversarial Learning for

Reducing Manual Annotation in Semantic Segmentation on Large Scale Miscroscopy Images: Automated Vessel Segmentation in Retinal Fundus Image as Test Case," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 794–800, 2017.

[17] B. Lecouat *et al.*, "Semi-Supervised Deep Learning for Abnormality Classification in Retinal Images," pp. 1–9, 2018.

[18] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 105–114, 2017.

[19] H. Zhao, H. Li, and L. Cheng, "Synthesizing filamentary structured images with GANS," *arXiv*, pp. 1–10, 2017.

[20] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2536–2544, 2016.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[22] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2414–2423, 2016.

[23] L. Collins, N. Peter, P. Terrence, and E. Alan, "Automatic 3D intersubject registration of

MR volumetric data in standardized Talairach space.," *J. Comput. Assist. Tomogr.*, vol. 18, no. 2, pp. 192–205, 1994.

[24] A. Borji, "Pros and cons of GAN evaluation measures," *Comput. Vis. Image Underst.*, vol. 179, pp. 41–65, 2019.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[26] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, no. 1, pp. 586–595, 2018.

[27] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, 2002.

[28] H. R. Sheikh and A. C. Bovik, "IMAGE INFORMATION AND VISUAL QUALITY," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 15, no. 2, pp. 430–44, 2006.

[29] A. Tversky, "Features of similarity. - 1977 - Tversky.pdf," *Psychol. Rev.*, vol. 84, no. 4, pp. 327–352, 1977.

[30] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *33rd Int. Conf. Mach. Learn. ICML 2016*, vol. 4, pp. 2341–2349, 2016.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical

image segmentation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, pp. 234–241, 2015.

[32] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, 2010.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016.

[34] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[35] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance Normalization: The Missing Ingredient for Fast Stylization," no. 2016, 2016.

[36] K. Krishnamoorthy, "Wilcoxon Signed-Rank Test," *Handb. Stat. Distrib. with Appl.*, pp. 339–342, 2020.

[37] J. T. Vogelstein *et al.*, "A general approach to progressive learning," *arXiv*, 2020.

[38] B. Evans, A.C., Collins, D.L., Milner, "An MRI-based stereotaxic atlas from 250 young normal subjects.," *Proc 22nd Annu. Symp. Soc. Neurosci.*, vol. 18, p. 408, 1992.

[39] A. Sriram *et al.*, "End-to-End Variational Networks for Accelerated MRI Reconstruction," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12262 LNCS, pp. 64–73, 2020.

[40] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv*, 2018.

[41] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 6627–6638, 2017.

[42] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for GANs do actually converge?,” *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 8, pp. 5589–5626, 2018.

Appendix A: Hyperparameter Optimization

A1. Discriminator Receptive Field

To determine whether a small aperture receptive field (70x70) or a large aperture receptive field (16x16), the *pTransGAN* model was trained on a subset of the IXI dataset. This set contained 461 T1 and T2 images and was tested on a set of 167 images. This hyperparameter optimization dataset contained images from 58 patients. The training and testing dataset were not reused when training and testing the final model. Given that the stride of discriminator is 2, with the 70x70 receptive field, we get 138,384 patches in the input image. Similarly, with the 16x16 receptive field, there are 230,400 patches in the input image. To compare the performance of *pTransGAN* with these two receptive fields, six metrics (PSNR, SSIM, MSE, LPIPS, UQI, and VIF) were recorded on the testing set (Appendix A Table 1 A and B). In contrast to previous literature ([9]), the larger and more conventional receptive field of 70x70 was found to outperform the larger aperture receptive field in all metrics. Thus the discriminator with the receptive field of 70x70 was used for all experiments.

Appendix A Table 1A: The 70x70 receptive field discriminator outperforms the 16x16 discriminator in all of the traditional metrics. Larger is better for PSNR and SSIM and smaller is better for MSE.

Metric	PSNR (dB)		SSIM		MSE	
	70x70	**16x16**	**70x70**	**16x16**	**70x70**	**16x16**
Minimum	**19.2**	19.1	0.63	**0.64**	**0.001**	0.002
Maximum	**29.1**	27.7	**0.94**	0.91	**0.012**	0.014
Mean	**24.9**	23.4	**0.88**	0.84	**0.003**	0.005

Appendix A Table 1B: The 70x70 receptive field discriminator outperforms the 16x16 discriminator in all of the perceptual metrics. Smaller is better in LPIPS and larger is better in UQI and VIF.

Metric	LPIPS		UQI		VIF	
	70x70	16x16	70x70	16x16	70x70	16x16
Minimum	**0.040**	0.051	**0.77**	0.76	**0.45**	0.42
Maximum	**0.134**	0.140	**0.98**	0.96	1.13	**1.19**
Mean	**0.068**	0.087	**0.95**	0.92	**0.85**	0.84

A2. Number of U-blocks in pTransGAN generator

In order to determine the number of U-blocks that would maximize the performance of the generator, four generators were created with different number of U-blocks (one, three, six, and seven blocks). This set contained 461 healthy T1 and T2 images and was tested on a set of 167 images. This hyperparameter optimization dataset contained images from 58 patients. To compare the performance of *pTransGAN* with different number of U-blocks, six metrics (PSNR, SSIM, MSE, LPIPS, UQI, and VIF) were recorded on the testing set (Appendix A Table 2). The 6 U-blocks generator outperformed the other models in all metrics expect for the VIF, which interestingly was the best with the one U-block model. Since the amount of improvement began to diminish after 3 U-blocks and 6 U-blocks gave better results than 3 U-blocks, the 6 U-blocks model was selected for the remainder of the experiments.

Appendix A Table 2: The minimum, maximum, and mean of the metrics PSNR, SSIM, MSE, LPIPS, UQI, and VIF for the images generated by models with 1, 3, and 6 U-blocks.

Metric	1 U-block			3 U-blocks			6 U-blocks		
	Min	**Max**	**Mean**	**Min**	**Max**	**Mean**	**Min**	**Max**	**Mean**
PSNR (dB)	18.04	23.97	21.25	19.22	28.0	23.92	**19.25**	**29.1**	**24.9**
SSIM	0.62	0.86	0.78	0.63	0.93	0.85	**0.63**	**0.94**	**0.88**
MSE	0.004	0.016	0.008	0.002	0.012	0.004	**0.001**	**0.012**	**0.003**
LPIPS	0.14	0.26	0.18	0.046	0.139	0.082	**0.040**	**0.134**	**0.068**
UQI	0.73	0.80	0.96	**0.78**	0.97	0.92	0.77	**0.98**	**0.95**
VIF	0.57	**1.40**	**1.03**	0.48	1.15	0.86	**0.45**	1.13	0.85

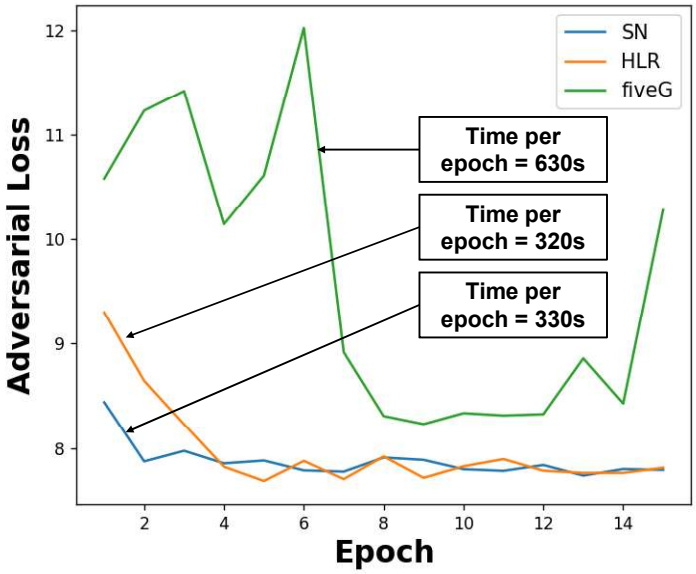
A3. Weights Parameters for the Overall Loss Function

The loss function (Equation 8) that is optimized in a min max fashion to train the *pTransGAN* has four main components: the adversarial, MAE, style, and content loss. Each loss has a weight hyperparameter associated with it. The weight parameters λ_{cGAN} and λ_{L1} were fixed to 1 and 100 respectively following the recommendations from [7]. A grid search algorithm was employed to identify the optimum values of λ_{style} and $\lambda_{content}$ are 0.001 and 0.00001. While calculating style and content loss through the VGG feature extractor, each convolutional block is assigned different weights (Equation 6, 7). The $\lambda_{style,i}$ parameters, which determines the contribution of the i^{th} convolution block of the feature extractor, were set so that the 2nd, 3rd, and 4th convolution blocks had the greatest contributions (in that order). $\lambda_{content,i}$ were set so that all but the last convolution block had equal contributions and the last block had 1/10 the contribution as the others. When using the grid search algorithm and determining these empirical weights, we used our hyperparameter optimization dataset. This set contained 461 healthy T1 and T2 images and was tested on a set of 167 images. This hyperparameter optimization dataset contained images from 58 patients.

Appendix B: Stabilizing Adversarial Training of pTransGAN with Perceptual Losses

In Figure 9A, we showed that when training *pTransGAN* with perceptual losses, the adversarial loss shows significant oscillations, causing the training to be unstable. This is undesirable since training of a GAN depends on establishing an equilibrium between the generator and the discriminator. Thus, to stabilize the training of the discriminator when *pTransGAN* is training with perceptual losses, we tried three different approaches.

First approach involved using spectral normalization (SN). We normalized the weight matrix of each layer of the discriminator, $\theta_{D,i}$, by its spectral norm [40]. Normalizing by the spectral norm ensures that the Lipschitz constant of the discriminator is limited to 1. Ideally, spectral norm is calculated through singular value decomposition of a matrix, however this can be computationally expensive. Instead, the spectral norm is approximated by the power iteration method (10 iterations) is used as described in [40]. The second approach tested is increasing the learning rate of the discriminator to four times that of the generator [41] (HLR). In this approach, the higher learning rate of the discriminator causes the generator and discriminator to achieve the Nash equilibrium faster. Finally, previous work has also suggested updating the discriminator more often than the generator [42]. Thus, in the final experiment we update the discriminator 5 times for each time the generator is trained (5G). We once again use the hyperparameter optimization dataset for training the model (with perceptual and adversarial losses) for fifteen epochs and keep track of the adversarial loss (Appendix B Figure 1). Moreover, the training time for each epoch is also tracked in order to measure how computationally heavy each approach is.



Appendix B Figure 1: Adversarial loss for *pTransGAN* training with adversarial, MAE, style, and content loss, but with spectral norm applied to the discriminator (blue, SN), learning rate of discriminator increased (orange, HLR), and training the discriminator five times for every instance of training the generator (green, fiveG).

Through Figure 1, we see that the most stable training is provided by the spectral norm and then increasing the learning rate. Changing how many times the discriminator is trained relative to the generator leads to more unstable training. In order to keep the training of *pTransGAN* stable, hereon, we train the model's discriminator with an increased learning rate of 0.0008 (4 times that of the generator) and with spectral normalization applied to the weights of the discriminator.

Appendix C: Miscellaneous Figures

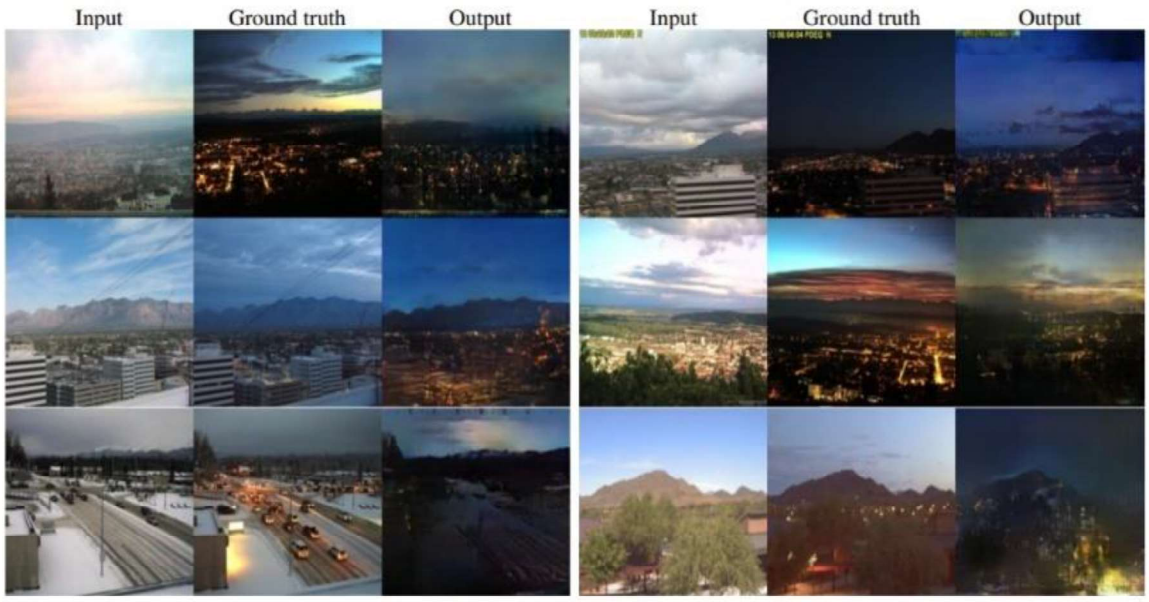


Figure 1: Results of a machine learning algorithm translating day-time city scape photos to night-time photos