Probabilistic Brain Extraction in MR Images via Conditional Generative Adversarial Networks

Saeed Moazami, Deep Ray, Daniel Pelletier, and Assad A. Oberai

***Abstract*— Brain extraction, or the task of segmenting the brain in MR images, forms an essential step for many neuroimaging applications. These include quantifying brain tissue volumes, monitoring neurological diseases, and estimating brain atrophy. Several algorithms have been proposed for brain extraction, including image-to-image deep learning methods that have demonstrated significant gains in accuracy. However, none of them account for the inherent uncertainty in brain extraction. Motivated by this, we propose a novel, probabilistic deep learning algorithm for brain extraction that recasts this task as a Bayesian inference problem and utilizes a conditional generative adversarial network (cGAN) to solve it. The input to the cGAN's generator is an MR image of the head, and the output is a collection of likely brain images drawn from a probability density conditioned on the input. These images are used to generate a pixel-wise mean image, serving as the estimate for the extracted brain, and a standard deviation image, which quantifies the uncertainty in the prediction. We test our algorithm on head MR images from five datasets: NFBS, CC359, LPBA, IBSR, and their combination. Our datasets are heterogeneous regarding multiple factors, including subjects (with and without symptoms), magnetic field strengths, and manufacturers. Our experiments demonstrate that the proposed approach is more accurate and robust than a widely used brain extraction tool and at least as accurate as the other deep learning methods. They also highlight the utility of quantifying uncertainty in downstream applications. Additional information and codes for our method are available at: `https://github.com/bmri/bmri`**

***Index Terms*— Bayesian inference, Brain extraction, Conditional generative adversarial networks, Medical imaging, Neuroimaging, Skull stripping, Uncertainty quantification**

I. Introduction

MRI brain extraction, or skull stripping, is the process of segmenting brain parts, namely the cerebrum, cerebellum, and brain stem organs, in a whole head MR image. The output of this task is usually in the form of a 3D image volume of the brain with the complimentary parts eliminated, or a 3D binary mask volume, which distinguishes a unified brain from the background voxels. Brain extraction can be used directly by the end-user, as a pre-processing step in software packages such as FreeSurfer [1], [2] and Fsl [3], or by downstream tools in medical imaging applications, which include grey and white matter volume measurement [4], monitoring of neurological diseases such as multiple sclerosis (MS) [5] and Alzheimer's disease [6], brain atrophy estimation [7], [8], and brain lesion segmentation [9]. In other words, in most neuroimiaging tasks non-brain parts are eliminated from head MRI images before being used by the subsequent algorithms. Consequently, even if the downstream steps in the workflow are accurate, the final results can be highly erroneous if the brain extraction is performed poorly. The impact of brain extraction on the final results of neuroimaging studies and its prevalence highlights the importance of accuracy and confidence with which this task is performed.

Given that skull stripping forms the basis of a wide range of applications, many tools have been developed over the years to automate and optimize this task (see Section II). However, there are still several aspects that can be improved. These include further gains in accuracy, robustness (the ability to work with heterogeneous and diverse MR images), and speed. Another aspect that is yet to be explored is the ability to quantify the uncertainty in skull stripping. That is, providing the end-user with the best estimate of the brain along with measures of confidence in that estimate. The method proposed in this manuscript particularly contributes to this aspect of brain extraction. In Section IV-C, we have extensively discussed the significance and applications of uncertainty quantification in brain extraction.

This paper introduces a deep Bayesian inference framework in which the brain extraction task is performed using a conditional generative adversarial network (cGAN) [10]–[12] architecture. The cGAN model is trained using a set of pairwise MR images of the head (denoted by $\boldsymbol{h}$) and the corresponding extracted brain (denoted by $\boldsymbol{b}$) as ground truth. Then using Bayes' theorem, the model learns the distribution of the brain image conditioned on the corresponding image of the head, that is $p_{B|H}(\boldsymbol{b}|\boldsymbol{h})$. Thereafter, given any new MR image of the head, the model is able to efficiently sample from the conditional distribution, that is, providing multiple likely brains for the given head image. These samples, in turn, can be used to calculate important statistics of the distribution, including the pixel-wise mean and standard deviation. The former provides an estimate of the extracted brain image, whereas the latter quantifies the diversity in the model's prediction as uncertainty in the extraction, indicating the model's confidence in generating different regions of the extracted brain. This

Manuscript submitted March 12, 2022.

Saeed Moazami, Deep Ray, and Assad A. Oberai are with the Department of Aerospace and Mechanical Engineering, Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA (e-mails: saeedmoa@usc.edu, deepray@usc.edu, and aoberai@usc.edu, respectively).

Daniel Pelletier and Saeed Moazami are with the Department of Neurology, Keck School of Medicine, University of Southern California, Los Angeles, CA, USA (e-mails: dpelleti@usc.edu and saeedmoa@usc.edu).

information can be used by the end-user in downstream applications. For example, they may focus on regions with higher uncertainty for potential quality control or manual corrections. Also, they can use the cumulative measure of standard deviation as a surrogate of the estimated error in the extractions for the entire brain. In Section IV, we demonstrate that the proposed algorithm compares well with the state-of-the-art methods for MRI brain extraction in terms of accuracy and robustness and does so in acceptable computational time IV-C. We also demonstrate that it quantifies uncertainty in the brain extraction task and how this estimate may be used to detect out-of-distribution input images, and to assess the level of confidence in the output.

The rest of the paper is organized as follows. In Section II, a brief review of the related work is provided. In Section III, the proposed method is described in detail. Numerical results are presented and discussed in Section IV, followed by the conclusions and directions for future research in Section V. Additional information, such as implementation details, is provided in the Appendix.

II. Related Work

In Section II-A, we list a number of deep learning (DL)-based frameworks that are used in medical imaging. Then, in Section II-B, we focus on existing generative adversarial networks (GANs)-based strategies in this field. Finally, in Section II-C, we focus on DL methods in the context of brain extraction, briefly highlighting the underlying network architectures, the advantages of the methods, and their potential drawbacks.

A. Deep Learning in Medical Imaging

Recently, deep learning-based methods have received significant attention in medical imaging, with applications beyond brain extraction and for modalities not limited to MRI. A broad spectrum of medical tasks has been accomplished using DL, many of them with superior performance when compared to traditional methods [13], [14]. These methods also utilize various architectures and algorithms to perform medical imaging tasks. The first group of DL-based methods, commonly referred to as deep convolutional neural network (CNN or DCNN)-based works, use neural networks as function approximators in a supervised direct inference framework. That is, the input image passes through several convolutional layers to provide an output. The aim is to minimize a loss function calculated based on the difference between the prediction and the target, thus producing results as close as possible to the target (ground truth). As it will be discussed in Section III-H, we compare our method with an implementation based on DCNN as a benchmark. We also provide additional details for this method in Appendix C.

In addition to DCNN, numerous other architectures in DL can be used for medical imaging tasks. Variational autoencoder (VAE) methods [15] use an encoder network to map the input image to a low dimensional latent space and a decoder network to reconstruct the input as closely as possible to the original image. The latent representation can be used for a variety of analyses and tasks [16]. For instance, in image harmonization, the latent representation can be manipulated to produce a scanner-invariant version of the input image while preserving the biological characteristics of the image [17]. Graph convolutional networks (GCN)-based methods aim to transform medical imaging data into graph representations to employ inherently efficient graph-driven methods on them [18], [19], such as graph-based node classification for disease prediction [20] or tumor segmentation [21]. Additionally, reinforcement learning (RL)-based methods are used in medical imaging where relying directly on evaluation metrics, human feedback, or generally, the notion of reward is more favorable than defining a directly optimizable loss function. We direct interested readers to [22] for a review of deep RL-based methods in medical imaging. Transformers or self-attention models [23] are another novel group of DL-based methods initially introduced for natural language processing (NLP). Vision transformers (ViT) [24] split images into small patches and regard them as tokens or words, enabling ViTs to perform a range of tasks in computer vision and medical imaging [25]. In TransUNet [26], the authors propose a hybrid model that benefits from both vision transformer-based architecture and more traditional CNNs for multi-organ segmentation in medical imaging. Lastly, diffusion models are a class of DL-based models that gradually add Gaussian noise to the images and learn to recover the input image from the noisy one. A fully trained diffusion model can generate images that belong to the training data distribution by passing a random noise to it. Diffusion models were initially introduced for image denoising [27] but have since shown potential in handling a range of medical imaging tasks [28], such as brain tumor segmentation [29] and reconstructing MRI images with enhanced characteristics [30]. We also use a variant of diffusion models as a benchmark (see Section III-H and Appendix-C.) and compare its results against our method in Section IV.

B. Generative Adversarial Networks in Medical Imaging

Among DL-based methods, those based on GANs [31] have achieved remarkable popularity. While various architectures and loss terms have been proposed for GANs [32], they typically comprise two neural networks, namely a generator and a critic or discriminator, that are trained in an adversarial fashion with opposing objective functions. This structure enables GANs to solve a wide range of medical imaging tasks, some of which are described below.

In its basic form, the GAN is trained via an unsupervised learning approach so that the generator is able to learn the underlying distribution from which the samples in the (finite) training set are drawn. Once trained, the generator synthesizes new samples from the learned distribution. This method is particularly useful in addressing the well-known challenges of scarcity and imbalance in medical data [33]. Studies described in [34] and [35] are examples of GAN-based models that can generate realistic brain MR images using a small number of training samples. While these methods are useful, more control over generating data is needed in most applications. This is achieved through variants of conditional GANs [10] or image-to-image translation networks [36]. Within this category are

algorithms that aim to generate images from one modality based on samples from another modality (CT to/from MRI [37], for example), or those that synthesize a missing pulse sequence in MR imaging [38].

Another common problem in medical imaging that GAN-based methods have been applied to is domain shift. That is, the drop in the performance of a given machine learning model during testing due to the shift between the training and test data distributions. A number of GAN-based methods aim to perform domain adaptation to mitigate this problem [39], [40]. Authors in [41] propose an unsupervised domain adaptation approach by conducting image appearance transformation and domain-invariant feature learning. This method enables cross-modality cardiac image segmentation, that is, training a model on the more abundant MRI data and then applying it to CT images. Domain adaptation can also be used to address the scarcity of labeled data in medical imaging. In [42], the authors develop a reverse domain adaptation scheme with an adversarial neural network that transforms real medical data into a synthetic representation, while trying to preserve important information. The synthetic image distribution is then used to generate more labeled medical data.

GANs have also been shown to be successful in performing image segmentation. Some of the most important medical imaging tasks lie within this category and include organ [43], lesion [9], and tumor segmentation [44]–[46]. Like any other task, a variety of GAN-based methods and architectures have been introduced in this area [47]. The interested reader is referred to [48]–[51] for surveys on medical imaging using GAN-based methods and to [52] for GANs in brain MRI.

C. Deep Learning Based Brain Extraction

Manual segmentation of the brain in MR images is a labor-intensive and time-consuming task. This has led to a variety of automated methods devoted to solving this problem. In review papers (see [53] [54], for example), brain extraction tools are categorized as conventional, machine learning (ML)-based, and DL-based methods, with subcategories within each class. Despite the growth of interest in ML/DL methods, some tools based on conventional methods are still widely used, especially in medical research and clinical environments. These include Fsl-BET [55], BEaST [56], and AFNI 3dSkullStrip [57].

Similar to other tasks in the medical imaging area, DL-based brain extraction methods are typically more robust when working with images with intensity variation and slight image misalignment. As a result, they do not usually require extensive pre-processing and parameter adjustment. Moreover, since they can utilize GPUs, their speed can be readily scaled. In the following, we provide a summary of some of these methods.

Authors in [58] present one of the first DL-based brain extraction methods. They utilize a CNN to capture 3D features in head MRI images with a relatively shallower architecture compared to the most recent works. More recently, the methods that utilize the U-Net architecture [59] have been successful in performing different image-to-image tasks in medical applications [60]. Likewise, most DL-based brain extraction approaches utilize the U-Net architecture. These methods can be applied to head image volumes (3D), MRI slices (2D), or three individual slice inputs from axial, sagittal, and coronal directions jointly (2.5D).

In the auto-context convolutional neural network (Auto-Net) [61], multiple fully connected and convolutional neural network layers are combined within a U-net structure. Models are then trained to perform brain extraction using images sliced along three main directions. Multiple 2D patch sizes are used to capture the context of different spatial scales. Image segmentation metrics such as dice similarity coefficient (see Section III-F for metrics' definitions) are then evaluated for different groups, such as healthy subjects, fetal, and patients diagnosed with Alzheimer's disease (AD). In another work [62], the performance of a U-net based artificial neural network (HD-BET) is compared against a number of conventional tools using different types of MR images. The best results (average dice similarity coefficient) are obtained with 3D T1-weighted images. The complementary segmentation network (CompNet) [63] method utilizes two pathways to learn features from both the brain and complementary tissues, i.e., bones and other parts of the head. The two trained models then work together to perform brain extraction. This method aims to enhance the robustness of the model in the presence of atypical or pathological brain images as it incorporates information from outside the brain in the training process. The paper reports metrics for healthy subjects and for images with synthetic pathological conditions. Authors in [64] propose multiview U-net (MVU-Net) architecture, in which three models are trained to extract the brain using 2D U-nets. The model requires head MRI images in three directions and fuses the three results linearly to produce a final brain mask. Also, the authors compare the evaluation metrics of U-Net-based architectures with and without skip connections. Moreover, the work presented in [65] investigate 3D U-net structures for brain extraction. The effect of a variety of loss functions is investigated in brain extraction task using a 2D U-Net architecture in [66].

U-Net-based methods also vary based on the training data characteristics. In [67], the authors introduce SynthStrip, a single model trained entirely on synthetic data with a wide range of modalities, intensity distributions, and artifacts. Further, the U-Net-based architectures have also been shown to be applicable to non-human brain images. In [68], a U-Net model is pre-trained on human imaging data. Then, a transfer-learning framework is used to update the model on non-human primates (NHP) data using fewer available training samples. Additional works that focus on the brain extraction task in rodents, such as MRI images of mice and rats, can be found in [69], [70]. Finally, in [71], the authors evaluate their U-Net-based method on several image modality combinations as inputs, for example, T1, T1 Gadolinium contrast-enhanced, T2, and T2-FLAIR together.

Most brain extraction algorithms in the literature rely on U-Net-based or similar encoder-decoder architectures, with a few exceptions, such as graph-based methods used in [72]. Also, they rely on deterministic methods that generate a single segmented brain image from a given input head image and

do not provide estimates about the certainty of their output. In contrast to this, the method developed in this manuscript performs brain extraction in a probabilistic context. That is, rather than producing a single image of the brain, it produces an ensemble of images which can then be used to extract the most likely extracted brain and to assess the uncertainty in this prediction. To accomplish this, it relies on the ability of a cGAN model to learn and efficiently sample from a conditional distribution [11], [12], [73].

While there is no prior work in quantifying uncertainty in brain extraction, there has been work on probabilistic image segmentation [74]. This includes the use of a collection of models to generate an ensemble of segmented images [75]. It also includes the combination of a U-Net with a conditional variational autoencoder [76] to generate multiple segmented images using a single model. More recently, conditional diffusion models [77], which are the conditional counterpart of diffusion models [78], [79], have also been employed successfully for image segmentation [29], [80], [81]. The cGAN developed in this study, and the conditional diffusion models share a common feature in that they both efficiently generate samples of the segmented image drawn from a complex conditional distribution by transforming a random vector drawn from a simple probability distribution via deep neural networks. In the cGAN, this transformation is accomplished by a single, highly expressive generator network, whereas in a conditional diffusion model, it is accomplished by multiple iterates of a relatively less complex U-Net. The fact that a single forward pass of a diffusion model requires multiple iterates makes the forward pass of the cGAN less expensive than that of a conditional diffusion model. On the other hand, in order to train the generator, the cGAN relies on adversarial learning, which can be hard to realize and prone to mode collapse (lack of diversity in outputs), whereas the diffusion network uses score matching, which is typically more stable [82]. The application of conditional diffusion models to brain extraction and their comparison with the cGAN developed in this work is an interesting avenue for research that can be explored in the future.

III. Materials and Methods

This section is dedicated to describing the brain extraction task and its formulation as a Bayesian inference problem. The proposed method and its workflow, the processing steps, datasets used to train and test the model, and the evaluation metrics are also explained. Additionally, a number of benchmark DL-based and non-DL-based methods which are used to assess the performance of the proposed algorithm are described.

A. Problem formulation

The brain extraction task is defined as a Bayesian inference problem that is solved using a deep generative adversarial network. In particular, given paired sample images of the head and the extracted brain, a conditional GAN (cGAN) [10] is trained to produce samples from the conditional distribution.

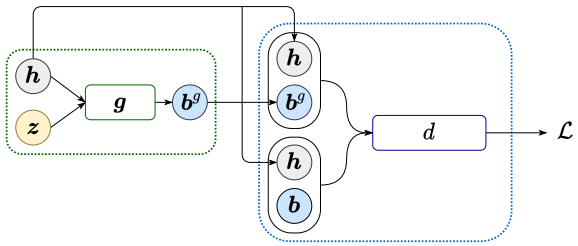

Fig. 1: The conditional generative adversarial network (cGAN) architecture used in the proposed method. The generator $\boldsymbol{g}$ receives real head image $\boldsymbol{h}$ and generates a brain $\boldsymbol{b}^g$ for any random latent vector $\boldsymbol{z}$. The critic d distinguishes between the generated brain images $\boldsymbol{b}^g$ and real ones $\boldsymbol{b}$ paired with $\boldsymbol{h}$.

The image size of a 3D MR image is denoted by $N_1 \times N_2 \times N_3$, where N_1, N_2, and N_3 are the number of voxels in the coronal, sagittal, and axial directions, respectively. An axial slice of the head, denoted by $\boldsymbol{h} \in \Omega_H \subset \mathbb{R}^{N_H}$, is a two-dimensional array of size $N_H = N_1 \times N_2$ pixels. The pixel values range from zero to one ($\boldsymbol{h} \in [0,1]^{N_H}$) after preprocessing discussed in Section III-D. The corresponding image of the brain is denoted by $\boldsymbol{b} \in \Omega_B \subset \mathbb{R}^{N_B}$, and is of the same size as $\boldsymbol{h}$ ($N_B = N_H$). In the brain image, the intensity of all pixels that are not a part of the brain is set to zero.

We assume that a dataset $S = \{(\boldsymbol{h}^{(i)}, \boldsymbol{b}^{(i)})\}_{i=1}^{N}$ containing N pairwise head and brain MR images from several subjects is available, and each sample in this dataset is drawn from a joint density function $P_{BH}(\boldsymbol{b}, \boldsymbol{h})$, which is unknown. The goal is to utilize this dataset to develop an algorithm for efficiently sampling from the conditional distribution $P_{B|H}(\boldsymbol{b}|\boldsymbol{h})$. That is, given a new image of the head, we wish to generate sample images of the brain conditioned on it. These samples can be used to generate the mean brain image and a standard deviation image, which allows us to quantify the uncertainty in the prediction. As described below, this is done via a modified version of the cGAN model [83] [11].

B. Conditional GAN (cGAN) Model

1) Model formulation: As illustrated in Fig. 1, the cGAN comprises two deep neural networks, namely, a generator $\boldsymbol{g}$ and a critic d. The generator, $\boldsymbol{g} : \Omega_H \times \Omega_Z \longmapsto \Omega_B$, accepts as input an image of the head, $\boldsymbol{h}$, and for any input instance of the latent vector $\boldsymbol{z} \in \Omega_Z \subset \mathbb{R}^{N_Z}$ generates an output image of the brain. That is, $\boldsymbol{b}^g = \boldsymbol{g}(\boldsymbol{h}, \boldsymbol{z})$. The latent vector is drawn from the distribution P_Z, which is the standard multivariate normal distribution. For a given $\boldsymbol{h}$, by sampling $\boldsymbol{z}$ from P_Z, the generator generates an ensemble of brain images. These can be thought to be drawn from a conditional distribution, $\boldsymbol{b}^g \sim P^{\boldsymbol{g}}_{B|H}$. The precise form of this distribution is determined by the weights of the generator, and the goal of the training procedure is to make this distribution as close as possible to the true conditional distribution $P_{B|H}$.

The critic, defined as $d : \Omega_H \times \Omega_B \longmapsto \mathbb{R}$, is responsible for distinguishing between image pairs from the true dataset, i.e., $(\boldsymbol{h}, \boldsymbol{b}) \sim P_{BH}$, and the pairs that their brain image is generated by the generator network, i.e., $(\boldsymbol{h}, \boldsymbol{b}^{\boldsymbol{g}})$, where $\boldsymbol{b}^{\boldsymbol{g}} \sim P^{\boldsymbol{g}}_{B|H}$. The

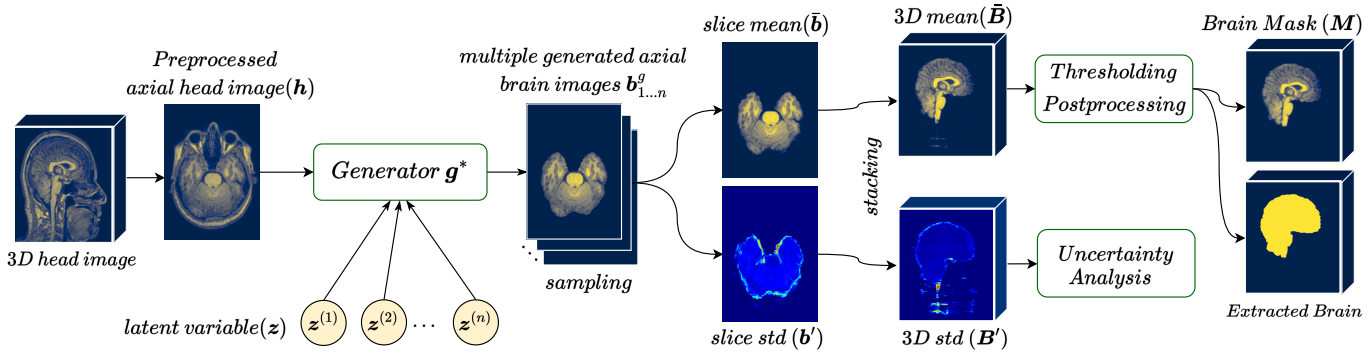

Fig. 2: Workflow of the proposed brain extraction method after training. It starts from loading the 3D image and feeding it as pre-processed axial slice images to the fully trained generator $\boldsymbol{g}^*$. For any given head slice image $\boldsymbol{h}$, n random samples of the latent vector $\boldsymbol{z}$ are passed to $\boldsymbol{g}^*$ to generate n brain samples $\boldsymbol{b}^{\boldsymbol{g}}_{1\ldots n}$. The samples are used to compute a single axial image of pixel-wise mean and standard deviation ($\bar{\boldsymbol{b}}$ and $\boldsymbol{b}'$, respectively) for any input slice image. These slice images are then stacked to form 3D mean and standard deviation images ($\bar{\boldsymbol{B}}$ and $\boldsymbol{B}'$). $\boldsymbol{B}'$ is used for uncertainty analysis, discussed in Sction IV-C. Threshholding and post-processing is performed on $\bar{\boldsymbol{B}}$ to generate the extracted brain mast $\boldsymbol{M}$, which is applied to the input 3D image to produce the final extracted brain $\hat{\boldsymbol{B}}$.

critic is trained to attain larger values for images from the true dataset and smaller values for images generated by the generator. This is done via the Wasserstein GAN loss function [84] as the difference between the values of the critic for true and generated images:

$$\mathcal{L}(d,\boldsymbol{g}) = \underset{\substack{(\boldsymbol{h},\boldsymbol{b})\sim P_{BH}\\ \boldsymbol{b}^g\sim P^{\boldsymbol{g}}_{B|H}}}{\mathbb{E}}[d(\boldsymbol{h},\boldsymbol{b}) - d(\boldsymbol{h},\boldsymbol{b}^g)], \tag{1}$$

where $\mathbb{E}$ denotes the expectation. Then, the following min-max problem is solved concurrently to find the optimal critic and generator:

$$d^*(\boldsymbol{g}) = \underset{d}{argmax}\,[\mathcal{L}(d,\boldsymbol{g}) + \lambda\mathcal{GP}]\,, \tag{2}$$

$$\boldsymbol{g}^* = \underset{\boldsymbol{g}}{argmin}\,\mathcal{L}(d^*(\boldsymbol{g}),\boldsymbol{g}), \tag{3}$$

where in (2) $\mathcal{GP}$ is the gradient penalty term [85] given by:

$$\mathcal{GP} = \underset{\epsilon\sim\mathcal{U}(0,1)}{\mathbb{E}}[(\|\partial_{\mathfrak{b}} d(\boldsymbol{h},\mathfrak{b})\| - 1)^2], \tag{4}$$

with λ coefficient that is a hyper-parameter (set to be 10 in this work). $\mathcal{GP}$ is used to enforce the critic to be 1-Lipshitz with respect to $\mathfrak{b} = \epsilon\boldsymbol{b} + (1-\epsilon)\boldsymbol{b}^{\boldsymbol{g}}$ being an average of actual and generated brains weighted using ϵ, a random number selected from a uniform distribution $\mathcal{U}(0,1)$. The 1-Lipshitz constraint is required to mathematically assure that finding the optimal generator $\boldsymbol{g}^*$ is equivalent to minimizing the (mean) Wasserstein-1 distance between the learned conditional distribution $P^{\boldsymbol{g}^*}_{B|H}$ and the target $P_{B|H}$ (see [11] for more details) and to numerically introduce additional regularisation which is useful to stabilize the training.

Convergence in the Wasserstein-1 implies weak convergence [86] of the distributions. Thus, in the converged limit, for the trained generator (also denoted by $\boldsymbol{g}^*$), we have:

$$\underset{\boldsymbol{b}\sim P_{B|H}}{\mathbb{E}}[l(\boldsymbol{b})] = \underset{\boldsymbol{b}\sim P^{\boldsymbol{g}^*}_{B|H}}{\mathbb{E}}[l(\boldsymbol{b})] = \underset{\boldsymbol{z}\sim P_Z}{\mathbb{E}}[l(\boldsymbol{g}^*(\boldsymbol{z},\boldsymbol{h}))], \tag{5}$$

where l is any continuous bounded function defined on Ω_B. In other words, for a given image $\boldsymbol{h}$, computing the expected value of any function of the brain $\boldsymbol{b}$ over the conditional distribution is the same as computing the expectation over the latent space of the same function applied to images obtained by passing the latent vector through the fully trained generator $\boldsymbol{g}^*$. Since the dimension of the latent space is typically small ($N_Z = 128$ in this study), and the cost of forward propagation through the generator network is low, this sampling process is a computationally feasible task to perform.

2) Generator and critic architecture: Schematic diagrams for generator $\boldsymbol{g}$, critic d, and all of their sub-blocks are provided in Figs. 9 through 12 in Appendix-A.

The generator $\boldsymbol{g}$, shown in Fig. 9, is implemented using a deep U-Net neural network architecture. Its input consists of the image of the head, $\boldsymbol{h}$, and the latent vector, $\boldsymbol{z}$. It includes a convolution layer, followed by three down-sampling blocks, one central customized residual network (ResNet) block, three up-sampling blocks, and two convolution layers (see Appendix-A for a detailed description). As the input is transmitted through the down-sampling blocks, its spatial resolution reduces, while the number of features increases by a factor of two. Exactly the opposite happens in the up-sampling block. Further, information from a given level of spatial resolution is directly transmitted from the down-sampling branch to the up-sampling branch via skip connections. Stochasticity is introduced in the network by utilizing the latent vector $\boldsymbol{z}$ to perform conditional instance normalization [87] operations at multiple spatial scales. A Sigmoid output function (applied pixel-wise) ensures the predicted brain image is bounded between 0 and 1.

For the critic network, shown in Fig. 10, the input is a set of paired images of the brain and the head. The input is passed through a convolution layer followed by three down-sampling and ResNet blocks. Similar to the generator network, down-sampling blocks reduce the spatial size of their input

by a factor of two and double the number of channels. Further, conditional instance normalization is replaced by layer normalization. The down-sampling blocks are followed by two dense layers, with the final output being a scalar.

C. Brain Extraction using cGAN Model

The brain extraction workflow is shown in Fig. 2. It begins by loading a head image volume and performing the pre-processing step discussed in III-D. The head MRI image is then fed as individual axial slices, $\boldsymbol{h}$, to the trained generator neural network $\boldsymbol{g}^*$ to perform the sampling process. This involves generating n latent vectors, $\boldsymbol{z}$, by sampling from a Gaussian distribution. These vectors and the head image are then passed through the generator $\boldsymbol{g}^*$ which generates n likely brain samples, $\boldsymbol{b}^{\boldsymbol{g}}_{1\ldots n}$, for the given head slice image. These brain images are used to compute a single pixel-wise mean image, $\bar{\boldsymbol{b}}$,

$$\bar{\boldsymbol{b}} = \frac{\sum_{i=1}^{n} \boldsymbol{g}^*(\boldsymbol{z}^{(i)}, \boldsymbol{h})}{n}, \boldsymbol{z}^{(i)} \sim P_Z, \tag{6}$$

That is, the value of any pixel of the single computed brain image $\bar{\boldsymbol{b}}$ is the average of intensities of the same pixel in n sampled brain images. Thereafter, the 3D brain volume image, denoted by $\bar{\boldsymbol{B}}$, is constructed by stacking these individual slices for the whole head.

Similarly, an image of the pixel-wise standard deviation, $\boldsymbol{b}'$, is calculated by,

$$\boldsymbol{b}' = \sqrt{\frac{\sum_{i=1}^{n} (\boldsymbol{g}^*(\boldsymbol{z}^{(k)}, \boldsymbol{h}) - \bar{\boldsymbol{b}})^2}{n}}, \boldsymbol{z}^{(i)} \sim P_Z. \tag{7}$$

In the equation above, the power of 2 and square root are interpreted as pixel-wise calculations in the images. The standard deviation image slices are also stacked to yield a volumetric image of pixel-wise standard deviation, denoted by $\boldsymbol{B}'$.

D. Pre-processing

The 3D images are loaded and reoriented to axial direction. Then they are minimally pre-processed by eliminating hyper intense noise voxels. We calculate a clipping value of the upper 99.99% percentile of the values within each 3D image and set the intensity of voxels with greater values to the clipping value. We also perform uniform min-max intensity normalization for 2D axial slices and resize them to $N_1 \times N_2$. No spatial normalization, or non-linear intensity modification, such as magnetic field bias correction, is applied to the images.

E. Post-processing

The final step in this workflow involves applying a thresholding filter to the output mean image $\bar{\boldsymbol{B}}$. The result is a three-dimensional binary image, or the mask, and is denoted by $\boldsymbol{M}$. In the mask, the brain voxels are denoted by 1, and the background is denoted by 0. We use a local thresholding scheme (see `threshold_local` in [88]), where a value is adaptively calculated for each voxel based on the mean value of the surrounding pixels. Voxels with intensity greater than the calculated threshold are set to one, and others are set to zero. Additionally, in the mask generated by the cGAN, we observe some small scattered collections of incorrectly labeled voxels. These occur as false positives outside the brain, typically in the lower and upper slices, and near the intersection of the cerebrum and the cerebellum, and rarely as false negatives within the ventricles. To address this issue, we apply two morphological filters to $\boldsymbol{M}$ that remove islands and cavities smaller than a specified minimal volume (see `remove_small_objects` and `remove_small_holes` operations in [89]). The mask obtained after these filters is not very sensitive to the value of minimal volume parameter as the size of the islands and cavities are much smaller than the size of the brain. In the results section, we report metrics for the mask $\boldsymbol{M}$ after applying these post-processing filters. Finally, the mask image $\boldsymbol{M}$ is transformed back to the original orientation and size of the input head image. The final extracted brain $\hat{\boldsymbol{B}}$ is the result of voxel-wise multiplication of this mask and the input head image.

F. Evaluation metrics

The dice similarity coefficient (DSC), positive predictive value (PPV), sensitivity (Se), and the ratio of predicted to target brain volume (VR) are used to evaluate the performance of the brain extraction. These quantities are defined as,

$$DSC = \frac{2\,|\boldsymbol{M} \cap \boldsymbol{T}|}{|\boldsymbol{M}| + |\boldsymbol{T}|}, \tag{8}$$

$$PPV = \frac{|\boldsymbol{M} \cap \boldsymbol{T}|}{|\boldsymbol{M}|}, \tag{9}$$

$$Se = \frac{|\boldsymbol{M} \cap \boldsymbol{T}|}{|\boldsymbol{T}|}, \tag{10}$$

$$VR = \frac{|\boldsymbol{M}|}{|\boldsymbol{T}|}. \tag{11}$$

In the equations above, $\boldsymbol{M}$ is the predicted, and $\boldsymbol{T}$ is the ground truth, binary volume mask. They each represent a set of voxels in three-dimensional space that attain a value of 1 for brain and 0 otherwise. The operator $|\cdot|$ denotes the sum of the absolute values of image voxel intensities and $\cap$ denotes the intersection of two binary images, i.e., a binary image that attains one only for those voxels that have a value of one in both images.

The DSC metric measures the similarity between two binary images and ranges from zero to one, where a value of one indicates a perfect match ($\boldsymbol{M} = \boldsymbol{T} \Leftrightarrow DSC(\boldsymbol{M}, \boldsymbol{T}) = 1$). This is why we have selected DSC as the primary similarity metric to evaluate the performance of brain extraction. The metrics PPV and Se also range from zero to one, with one being the most desired value. However, a value of one does not necessarily indicate a perfect match. For example, a mask $\boldsymbol{M}$ that has a single voxel set correctly to one and all others set to zero will yield a PPV of one. Similarly, a mask M that has all voxels set to one will yield a Se of one. Neither of these masks would likely be a particularly favorable approximation of the true mask $\boldsymbol{T}$. Volume ratio (VR) can vary from zero to infinity, with one being the optimal value. We note that

the entire brain is considered in the calculation of the metrics, and no part, like the ventricles, for example, is excluded.

G. Datasets

For training and evaluation of the proposed model, we use four publicly available datasets separately, and then an extended set that combines them all.

- The Neurofeedback Skull-stripped (NFBS) dataset consists of 125 anonymized (defaced) 3D T1-weighted MR images of 21 to 45 year old subjects, with a variety of clinical and subclinical psychiatric symptoms [90]. The dataset contains paired images of the head and the brain, where brain extraction is performed using the BEaST method [56] and then corrected manually. Out of 125 images, 70, 5, and 50 are used for training, validation, and testing the model, respectively.
- The Calgary-Campinas-359 (CC-359) dataset contains 359 3D T1-weighted images in six bins: three vendors (GE, Philips, and Siemens) at two magnetic field strengths (1.5T and 3T). Each bin has 60 subjects (except Philips 1.5T with 59). CC-359 also contains paired images of head and brain, where the brain images are the output of a separately trained supervised model to provide a consensus from a number of brain extraction tools [91]. Out of 359 images, 132 (22 from each bin) are used for testing, 18 (3 per bin) for validation, and the remainder for training.
- The LONI Probabilistic Brain Atlas (LPBA) [92] dataset provides 40 native space 3D T1 MRI and their paired brain images, from which 26 are used for training, 2 for validation, and 12 for testing.
- The Internet Brain Segmentation Repository (IBSR v2) [93] contains 18 pair head and brain images. We use a split of 10, 1, and 7 for training, validation and testing.
- Finally, we construct a fifth dataset by combining the NFBS, CC-359, LPBA, and IBSR datasets. This dataset is labeled as extended (EXT) in this paper. The same subjects for training, validation, and testing are used for this dataset, where an aggregate of approximately 29,050 pairs of head-brain slices (19,200, 6,400, 2,600, and 850 from CC-359, NFBS, LPBA, and IBSR) are used for training. The main output model of this work uses this dataset for training (see Section IV).

In addition, we use an independent internal dataset to evaluate whether the quantification of uncertainty provided by the probabilistic models can be used as a measure to evaluate their performance and to determine when manual quality control (QC) is required (see Section IV-C). The dataset contains 249 MRI images of multiple sclerosis (MS) patients.

H. Comparison

We compare the performance of the proposed cGAN method against six other methods. These include two versions of BET [55], which is a commonly used non-DL based tool. The first version is obtained by using the default parameter values [94], and the second is obtained by performing a grid search to select the "optimal" parameters, and is referred to as BET-optimal (BET-O). We also compare the proposed cGAN method against the state-of-the-art DL-based methods. To that end, we conducted a literature search to identify some of the most accurate algorithms. Out of these, we decided to use two models, CompNet [63], and HD-BET [62], for the comparison since they yielded the most accurate results. We re-trained the CompNet model using the same data used to train our model. For HD-BET, we use the pre-trained model available for this method. Additionally, we include a direct inference model based on supervised error minimization. This model is based on a DCNN architecture that mimics the architecture of the generator network in the cGAN model. Finally, as discussed in Section II-A, diffusion models have received significant attention in the field of medical imaging due to their promising performance. Moreover, these models can produce an ensemble of outputs, leading to the ability to generate uncertainty maps similar to our model. Hence, we compare the performance of our model against a diffusion model-based implementation as a benchmark. To that end, we re-train the model based on the work presented in [29], which is a conditional version of the denoising diffusion probabilistic model (DDPM), originally inspired from [95]. The implementation of the DCNN and DDPM methods is discussed in detail in Appendix-C.

IV. Results and Discussion

The results of the proposed algorithm are presented in this section. In Section IV-A, we provide a qualitative discussion about the typical outputs of our method. A quantitative analysis is presented in Section IV-B to demonstrate the accuracy and robustness of our method in comparison to other methods. Finally, we discuss the estimates for uncertainty and their advantages in Section IV-C. Where appropriate, technical and neuro-imaging implications of the results are discussed.

A. Brain Extraction Results

We demonstrate the performance of the proposed algorithm by presenting samples of input and output images. Fig. 3 shows the brain extraction results of multiple slices incrementally covering a whole brain of a single subject. In the figure, we start from the top of the head and then progressively move down. From left to right, the first column is the input head image, $\boldsymbol{h}$, the second column is the ground truth brain image, $\boldsymbol{b}$, the third column is the mean image, $\bar{\boldsymbol{b}}$ (pixel-wise average of samples generated by generator $\boldsymbol{g}^*$ before thresholding and filtering), the fourth column is the output brain or the mean image after post-processing $\hat{\boldsymbol{b}}$, and the fifth column is the pixel-wise standard deviation image, $\boldsymbol{b}'$. These images are generated using a subject from CC359 dataset.

In all slices, we observe that the output mean image is remarkably close to the target image. Further, the model is robust enough to accurately extract relatively atypical brain parts, such as slices with varying thickness of skull or meninges (especially the topmost part of the brain shown in

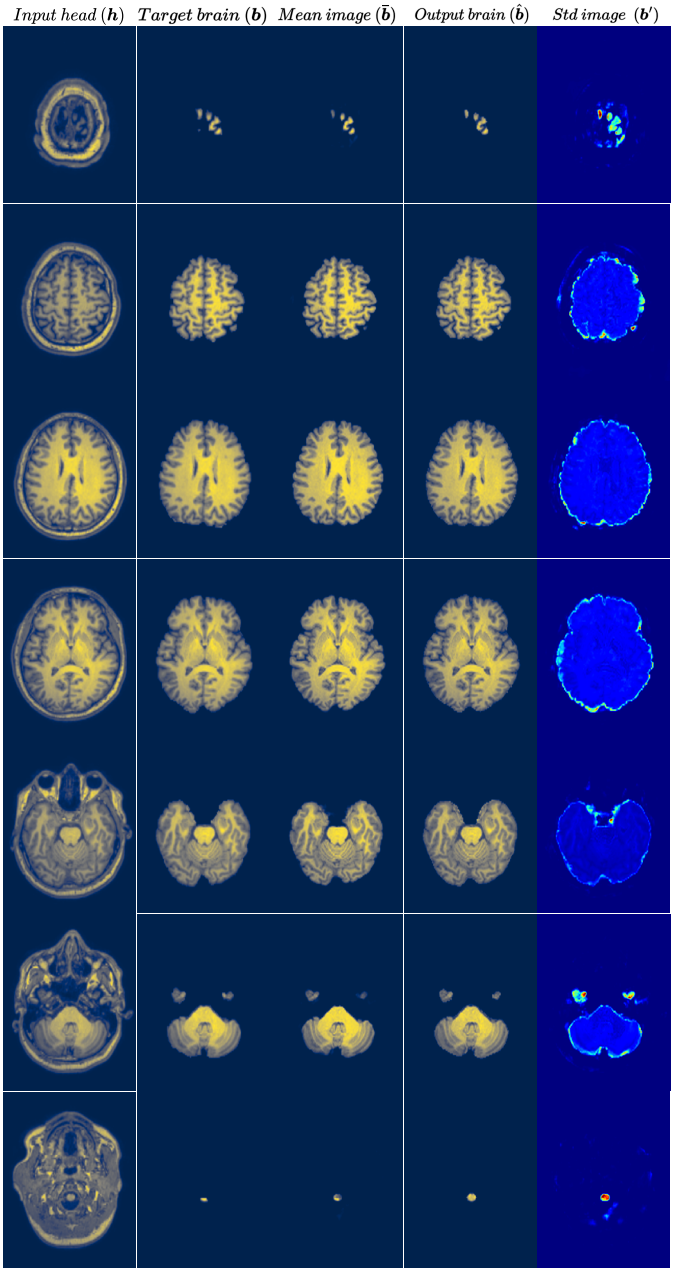

Fig. 3: Brain extraction results for multiple slices of a subject. Left to right: input head image, target brain image, predicted mean image, mean image after post-processing, and the standard deviation image.

the first row), cerebellum (lower part of the sixth row), multi-domain slices where the cerebellum and cerebrum appear in discontinuous locations (inferior temporal gyri shown in the middle of the sixth row), and the brain stem (shown in the last row). In the fifth column, we observe that the standard deviation, and therefore uncertainty, is peaked mainly along a thin 2-3 pixel region interface between the brain and the remainder of the image. Uncertainty quantification is discussed in Section IV-C in more detail.

B. Comparison and Quantitative Analysis

We compare the performance of our method and the algorithms discussed in Section III-H using the NFBS, CC359, LPBA, IBSR, and EXT datasets, based on the metrics presented in Section III-F. We train the cGAN, DCNN, CompNet, and DDPM models on EXT dataset (aggregate of all training subjects from the other datasets) and evaluate them on test subjects of each of the datasets (3D MR images that were kept aside for testing). We note that HD-BET method is evaluated based on the available model that is pre-trained on EORTC-26101 dataset [62]. Also, no training is involved for BET and BET-O since they are non-DL methods.

As discussed in Section III, the cGAN model performs brain extraction on axial slices and then stacks images to provide a 3D volume. Then a post-processing step produces the 3D binary mask image $\boldsymbol{M}$. We evaluate the accuracy of

TABLE I: Mean and standard deviation (in parenthesis) of evaluation metrics for different methods evaluated on five datasets. The best method for each metric within each dataset is shown in bold font.

NFBS	DSC	PPV	Se	VR
BET	77.48 (7.43)	73.54 (6.02)	73.55 (12.80)	0.886 (0.179)
DDPM	91.31 (15.98)	96.26 (0.68)	89.95 (20.14)	0.934 (0.208)
BET-O	91.81 (2.17)	87.58 (4.20)	96.60 (0.72)	1.106 (0.059)
CompNet	95.28 (3.80)	92.61 (6.60)	98.39 (0.57)	1.069 (0.094)
DCNN	96.49 (0.15)	**97.43** (0.22)	95.58 (0.27)	0.981 (0.004)
HD-BET	97.23 (0.28)	95.35 (0.66)	**99.19** (0.27)	1.040 (0.009)
cGAN	**97.84** (0.43)	96.93 (1.17)	98.77 (0.58)	**1.019** (0.017)

CC359	DSC	PPV	Se	VR
BET	87.27 (7.82)	80.04 (11.47)	97.11 (2.70)	1.246 (0.235)
DDPM	95.97 (2.79)	96.35 (0.86)	95.75 (4.92)	0.994 (0.052)
BET-O	96.58 (0.61)	95.18 (1.36)	98.05 (1.45)	1.310 (0.027)
CompNet	96.52 (4.28)	95.56 (7.02)	97.84 (0.63)	1.033 (0.123)
DCNN	96.54 (1.26)	**98.68** (1.03)	94.55 (2.50)	0.958 (0.031)
HD-BET	97.11 (0.50)	94.72 (1.01)	**99.64** (0.15)	1.052 (0.012)
cGAN	**97.19** (0.39)	96.43 (0.94)	97.99 (0.65)	**1.016** (0.015)

LPBA	DSC	PPV	Se	VR
BET	94.90 (2.71)	91.85 (4.91)	98.32 (0.52)	1.074 (0.067)
DDPM	97.21 (0.26)	96.83 (0.64)	97.59 (0.47)	1.008 (0.010)
BET-O	97.24 (0.29)	96.47 (0.73)	98.03 (0.50)	1.020 (0.012)
CompNet	97.57 (0.22)	97.27 (0.53)	97.87 (0.50)	**1.006** (0.010)
DCNN	96.22 (0.50)	93.05 (0.99)	**99.62** (0.93)	1.070 (0.012)
HD-BET	97.51 (0.01)	98.21 (0.47)	98.61 (1.06)	1.014 (0.011)
cGAN	**97.57** (0.24)	**98.31** (0.52)	96.84 (0.56)	98.51 (0.010)

IBSR	DSC	PPV	Se	VR
BET	87.07 (5.02)	97.01 (2.25)	79.50 (8.38)	0.821 (0.098)
DDPM	86.33 (9.27)	81.86 (13.29)	92.27 (5.56)	1.153 (0.172)
BET-O	87.20 (5.19)	**98.43** (0.48)	78.69 (8.08)	0.800 (0.085)
CompNet	81.98 (8.09)	75.61 (13.80)	91.78 (6.72)	1.26 (0.276)
DCNN	96.29 (0.70)	98.23 (0.56)	94.44 (1.34)	0.961 (0.016)
HD-BET	97.37 (1.02)	96.80 (2.74)	**98.02** (1.09)	1.014 (0.040)
cGAN	**97.43** (0.50)	97.42 (1.10)	97.47 (1.48)	**1.001** (0.025)

EXT (Extended)	DSC	PPV	Se	VR
BET	85.47 (8.89)	82.42 (10.67)	90.86 (12.41)	1.130 (0.266)
DDPM	94.58 (8.74)	95.80 (3.93)	94.37 (10.98)	0.987 (0.123)
BET-O	95.14 (2.97)	93.54 (4.17)	97.03 (4.00)	1.040 (0.069)
CompNet	95.58 (0.93)	94.00 (1.92)	97.71 (0.48)	1.052 (0.027)
DCNN	96.50 (3.40)	**98.01** (1.62)	95.94 (2.60)	1.035 (0.024)
HD-BET	97.17 (0.49)	95.05 (1.21)	**99.38** (0.54)	1.046 (0.018)
cGAN	**97.38** (0.48)	96.70 (1.10)	98.10 (0.83)	**1.015** (0.018)

$\boldsymbol{M}$, and the mask created by other methods, against the target mask image of the test subjects, $\boldsymbol{T}$. The quantitative metrics DSC, PPV, Se, and VR for the algorithms are compared in Table I, where we report the mean and the standard deviation (in parenthesis) of the metrics for each method. These statistics are obtained by applying each method to test subjects from the five datasets.

We also show the distribution of the results as boxplots in Fig. 4 for each metric, method, and dataset. The horizontal line in the middle of a boxplot indicates the average value of the metric over all test subjects, and the whiskers denote the range of the values excluding outliers. The color boxes show the interquartile range (IQR). The outliers, i.e., values that are more than 1.5 IQR below 25^{th} percentile and 1.5 IQR above 75^{th} percentile, are shown as cross markers. Each subplot in Fig. 4 reports results for a metric, and within each subplot, results are first arranged as per the datasets and then as per the method used. Results for EXT dataset are aggregates of the other datasets and represent the overall performance of the methods on all test subjects.

We regard DSC as the primary metric because, unlike other metrics, it yields a value of one only in the case of a perfect match between $\boldsymbol{M}$ and $\boldsymbol{T}$. We also consider EXT as the main dataset for image segmentation evaluation as it contains test subjects from all other datasets, making it the largest and most heterogeneous dataset. By observing the mean DSC values for all methods on the EXT dataset, we conclude that cGAN is the most accurate method, followed by HD-BET, DCNN, CompNet, BET-O, and then DDPM and BET. An examination of the size of the whiskers in the boxplots, or the variance reported in the table, reveals that the cGAN method has the smallest inter-subject variation, demonstrating its robustness. This improved performance may be attributed to the adversarial learning component in the cGAN-based methods through which they learn the underlying data distribution for better generalization. Moreover, we note that HD-BET performs slightly better than DCNN (around 0.7%). This may be attributed to the fact that the HD-BET model benefits from being trained on a significantly larger dataset of approximately 1,600 subjects ($\sim$2:1 ratio of training to testing from 2,401 T1-w images [62]) compared to 322 training subjects in the EXT dataset.

The cGAN method is also the most accurate in achieving values of volume ratio (VR) that are very close to unity with very small spread. This is particularly useful when image segmentation is followed by the estimation of the brain volume in tasks like brain atrophy analysis. The DCNN method produces the best PPV values, outperforming the cGAN, which is second best by 1.3%. HD-BET performs marginally better than cGAN in Se (1.2%). This can be explained by recognizing that HD-BET uses a larger smoothing filter with a more inclusive strategy, leading to relatively larger brains. Consequently, the generated mask $\boldsymbol{M}$ has a higher probability of including the target $\boldsymbol{T}$, leading $\boldsymbol{M} \cap \boldsymbol{T}$ to be closer to $\boldsymbol{T}$, which in turn pushes Se towards unity (see Section III-F for definitions). However, this strategy results in an increase in the predicted volume VR error in the HD-BET method by 4.6%.

We further compare the performance of the cGAN method when trained on different datasets in order to investigate the impact of dataset size on results. The results are shown in Table II, where for each dataset (NFBS, CC359, LPBA, and IBSR) accuracy metrics are presented for the model trained on the EXT dataset (labeled cGAN) and for the model trained on each dataset (labeled cGAN-dataset-name) using test subjects from that dataset. Based on these results, we conclude that when the dataset is sufficiently large (NFBS and CC359) the algorithm trained on that dataset can successfully generalize to unseen intra-dataset test subjects. Therefore, it performs slightly better than the model trained on the extended dataset (around 0.2% for NFBS and 0.8% for CC359). Conversely, when the dataset is small (LPBA and IBSR with 26 and 10 training images, respectively), the algorithm trained on the extended dataset generalizes better and its accuracy is higher (around 0.8% for LPBA, and 0.4% for IBSR).

TABLE II: Statistics of evaluation metrics, mean and standard deviation in parenthesis, for the cGAN model trained on EXT dataset (cGAN), as the combination of training subjects, in contrast to cGAN models trained on each of the four individual datasets (cGAN-dataset-name). Evaluations are done on intra-dataset test subjects. The best model for each metric within each dataset is shown in bold font.

	NFBS			
	DSC	PPV	Se	VR
cGAN-NFBS	**98.06** (0.75)	**97.80** (1.45)	98.34 (0.50)	**1.006** (0.018)
cGAN	97.84 (0.43)	96.93 (1.17)	**98.77** (0.58)	1.019 (0.017)
	CC359			
	DSC	PPV	Se	VR
cGAN-CC359	**98.03** (0.48)	**98.64** (1.03)	97.43 (0.87)	**0.988** (0.017)
cGAN	97.19 (0.39)	96.43 (0.94)	**97.99** (0.65)	1.016 (0.015)
	LPBA			
	DSC	PPV	Se	VR
cGAN-LPBA	96.80 (0.32)	96.41 (0.90)	**97.19** (0.47)	**1.008** (0.014)
cGAN	**97.57** (0.24)	**98.31** (0.52)	96.84 (0.56)	98.51 (0.010)
	IBSR			
	DSC	PPV	Se	VR
cGAN-IBSR	97.02 (0.59)	**97.57** (1.15)	96.50 (1.44)	0.989 (0.024)
cGAN	**97.43** (0.50)	97.42 (1.10)	**97.47** (1.48)	**1.001** (0.025)

Additionally, to provide intuition about how the prediction from different methods can differ, the masks generated by these methods are shown in Fig. 5 in the sagittal view. The input head and the target brain images are shown in the first column. In columns 2-8, the results generated by each method are displayed, and in column 9 the ground truth is displayed. The first row contains the brain masks where green pixels indicate a true negative prediction, gray pixels indicate a true positive prediction, orange pixels indicate a false positive prediction and pink pixels indicate a false negative prediction. The second row shows the boundary of the predicted brain mask overlaid on the head image so as to compare it with the corresponding brain tissue in the input image. The last column is the target image and therefore comprises only gray and green pixels. The sample 3D image is selected from the CC359 dataset.

Based on our observations, there is a significant difference

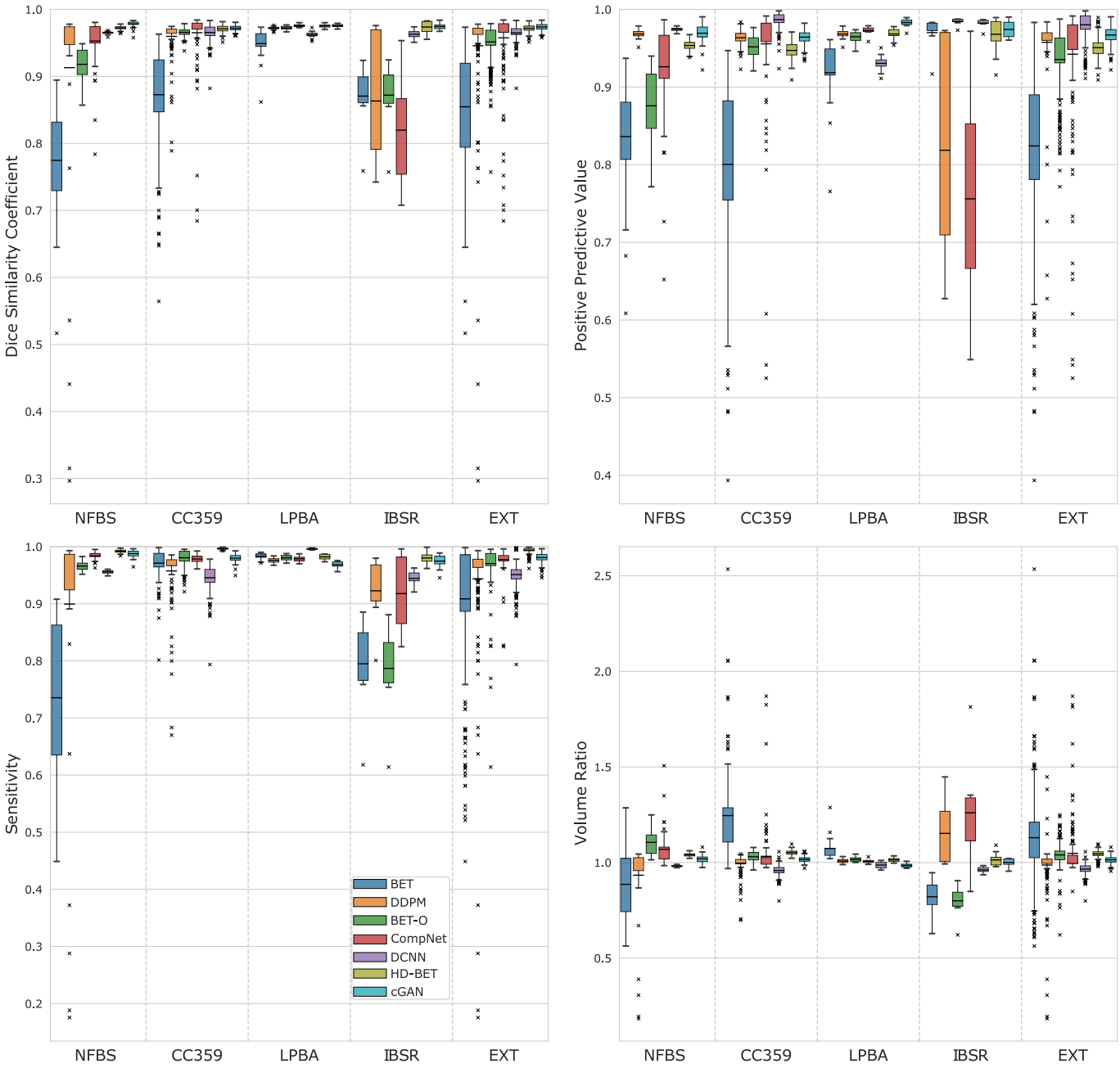

Fig. 4: Boxplots of evaluation metrics (DSC, PPV, SE, and VR) for NFBS, CC359, LPBA, IBSR, and EXT datasets. Within each dataset, boxplots are shown for BET, DDPM, BET robust (BET-O), CompNet, DCNN, HD-BET, and our method (cGAN) by distinct colors shown in the legend.

between the performance of our method and the BET results. The BET method with default parameters (second column) generates large regions of false-negative and false-positive predictions. When it is used with optimal parameters (third column), the performance is improved. However, large false negatives in superior and frontal regions still remain. These findings correlate with the results observed in Table I. It is noteworthy that the performance of DDPM, BET-O, and CompNet are noticeably dataset-dependent, e.g., DSC of DDPM's is 97.21% for LPBA and 86.33% for IBSR, BET-O's DSC is 97.57% for LPBA and 87.20% for IBSR, and CompNet's DSC is 97.24% for LPBA and 81.98% for IBSR. The DCNN performance is generally acceptable. However, in atypical input slices, its performance drops. These include slices containing varying skull and meninges thicknesses or disconnected brain segments. The performance of our model (cGAN) and HD-BET are generally high, stable, and comparable, with slightly different characteristics. The HD-BET mask is more inclusive in some areas, such as regions anterior to the midbrain and inferior to the hypothalamus (pituitary gland), and less inclusive in the brain stem (caudal medulla). The cGAN model produces results that better follow the

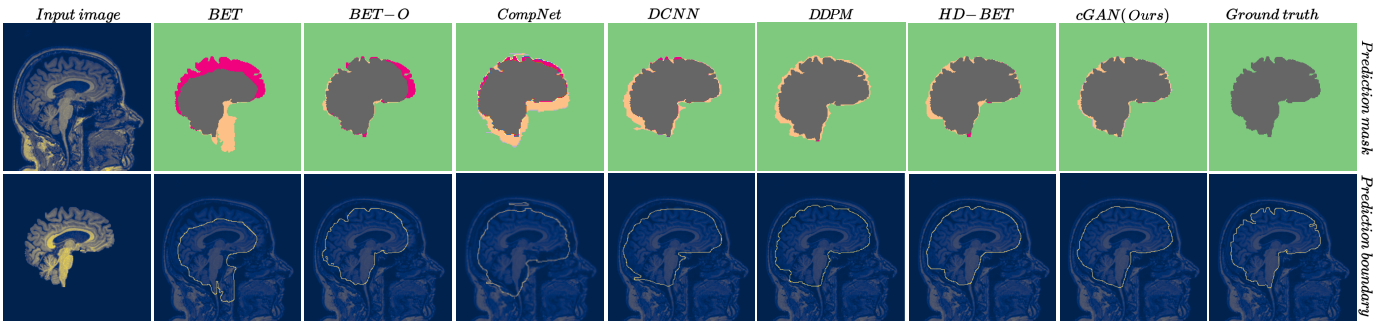

Fig. 5: From left, the first column is a sample input image and its brain. From the second column, the generated masks and the masks' boundaries are shown for the indicated methods. The last column is the ground truth. In the mask images, green indicates true negative, gray true positive, orange false positive, and pink false negative predictions.

geometry of the brain's gyri (ridges) and sulci (grooves) while staying accurate by not including irrelevant areas. It should also be noted that the small false positive areas indicated with orange pixels in the cGAN mask (eighth column in Fig. 5) are not necessarily detrimental for the subsequent tasks. This is because most neuroimaging tasks can handle brain images with smoothed sulci. This is not true for false negatives since any "lost brain tissue" is not recoverable in the subsequent steps. Further, we observe that DDPM results in occasional severe brain extraction failures. In other words, the DDPM performance is generally high in the majority of cases and as low as 29.66% in a few cases. Consequently, the mean performance of DDPM stands lower than other DL-based methods (94.58%). Also, based on our experiments, the inaccuracy in DDPM outputs is often in the form of false negatives, meaning that parts of the brain are excluded from the final prediction (unrecoverable).

C. Role of Uncertainty Quantification

In this section, we discuss how estimates of uncertainty, as given by the pixel-wise standard deviation of extracted brain samples generated by the model, can be utilized. In the first two columns (from the left) of Fig. 6, we have plotted the input head and the target brain images, and then two sets of columns generated by the DDPM and cGAN models. The fifth, sixth, and seventh columns of this figure illustrate three samples of the brain generated by the cGAN model ($\boldsymbol{b}^g_{1\ldots n}$). The pixel-wise mean of all of the generated samples, the pixel-wise mean after post-processing, and the pixel-wise standard deviation images are shown in the eighth, ninth, and tenth columns, respectively. The third and fourth columns show the pixel-wise mean and standard deviation images of the ensemble of images generated by the DDPM model. The DDPM and cGAN models use an equal number of samples (20) to generate the ensemble.

In the first row, a typical representative head slice (Original 1) is used. We observe that the standard deviation is higher at a narrow interface between the brain and the rest of the tissue in both models. Accordingly, the three shown samples generated by the cGAN generator are almost identical. In other words, the model is certain about its prediction. In the second row, we introduce a typical aliasing artifact in the original head image by altering the k-space representation of the head MRI image (setting every fifth column of the real component in phase encoding direction to zero). We observe that the mean images are robust to this perturbation. The standard deviation image of cGAN now captures a higher level of uncertainty, especially in a thin region on the left side of the image. Conversely, this artifact has a minimal impact on the standard deviation image generated by the DDPM model. In the third row, the artifact is more severe as every third column in the k-space is obscured. The standard deviation in cGAN responds to this by indicating higher levels of uncertainty, especially on the left side of the image. When comparing the mean and the target images in this region, we recognize that this is precisely where the algorithm has incorrectly labeled some pixels as brain (false positives). This example demonstrates how regions of high pixel-wise standard deviation can alert the end-user to where the prediction may be incorrect or ambiguous. We also note that this error is successfully eliminated in the post-processing step. In the DDPM output, the standard deviation increases slightly in some areas but generally remains minimal. The fourth row shows the original input image severely distorted by a blurring filter. As a result, the generated samples of cGAN are significantly different, and the standard deviation peaks in several regions. This is an indication that the input image is out of distribution (OOD) compared to the images used to train the model. Typically, machine learning algorithms are prone to producing erroneous outputs from OOD inputs without any warning. In contrast, the method introduced in this paper provides an estimate of uncertainty, which in turn can be inferred as a measure of confidence in the prediction. Again, the output of DDPM shows minimal increase in the standard deviation, which is a cause for concern given that this is an OOD sample. Row five presents an example of an input image altered by adding a random Gaussian noise filter. The noisy image values are clipped to preserve the range of input data between zero and one. The cGAN model prediction is robust to the added noise and produces a higher level of uncertainty. In contrast, DDPM gives the impression of being sensitive to added noise in this example, as its mean image shows severe failure in the form of a large false negative area. Accordingly, it produces a large standard deviation. However, even then it fails to cover the entire area where its prediction is

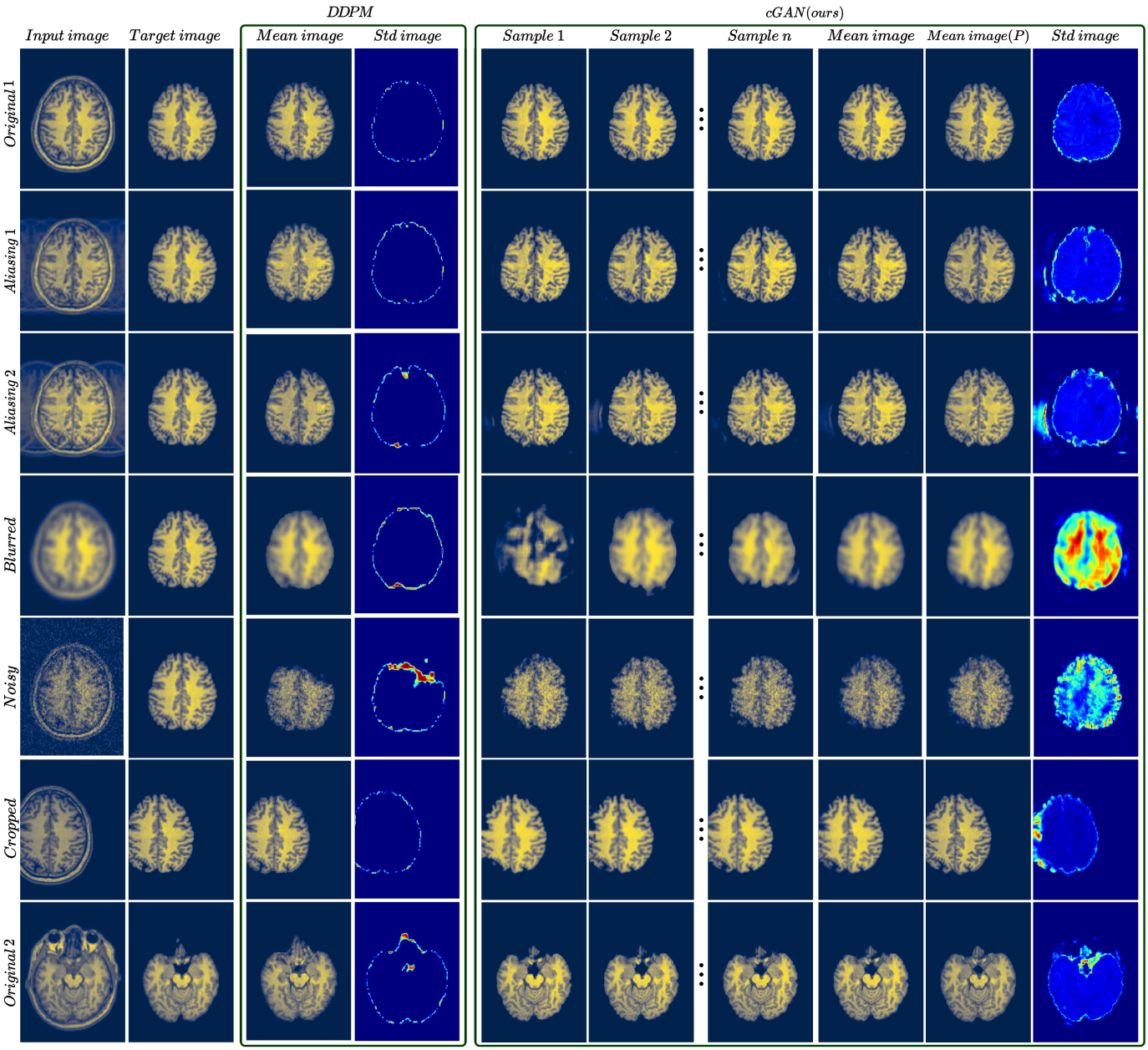

Fig. 6: From left to right, the input image, target brain, and the mean and standard deviation images generated by the DDPM method (diffusion model) are shown. The remaining columns belong to the cGAN method, where three generated sample brain images, the mean image, the mean image after post-processing, and standard deviation images are depicted. Each row shows the results using different input images. The first row is a typical head slice (Original 1). In the second row aliasing artifact is synthetically induced to the original 1 image. In the third row, the artifact is more severe. In the fourth row, the input image is blurred. Row five shows the input image altered by adding noise. The sixth row has an improper cropping. In the last row (Original 2) the geometry of the brain is relatively complex with visual similarities with other tissues. The main observation is that in atypical input images, the samples generated by cGAN are more diverse and the standard deviation is higher, reflecting the model's uncertainty in prediction.

erroneous. In the sixth row, an image with improper cropping (a common issue in clinical context) is shown where a part of the brain is missing. As can be seen, the brain extraction outputs of both methods are reasonably accurate. However, the cGAN algorithm produces a peak in the standard deviation in the affected area, while the DDPM appears to be insensitive to this artifact. The last row (Original 2) in Fig. 6 is an MRI slice intersecting the brain tangentially through the gyrus rectus and orbital gyri (the part of the brain between eye globes and above the nasal cavity). This area is visually similar to the optic chiasm, a cross-shaped tissue that does not belong to the brain and is found in a nearby location. Due to this ambiguity, it is reasonable to expect that in some outputs, this region is treated as part of the brain, and in others, it is not. The proposed

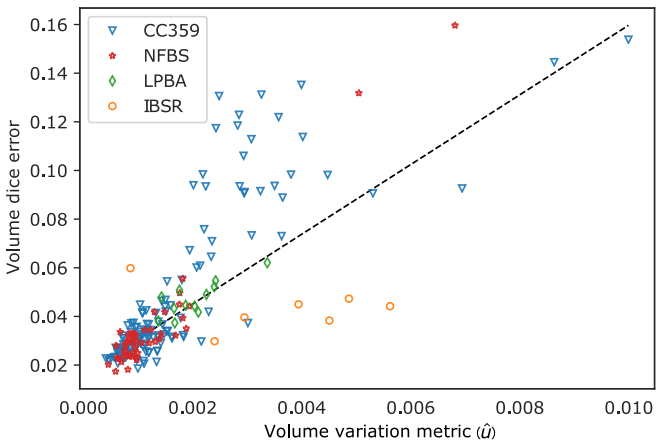

Fig. 7: Plot of dice error $(1 - DSC)$ and aggregate standard deviation measure for the test subjects with a correlation coefficient of 0.788.

method reproduces this uncertainty and reports large standard deviation values in this region. The standard deviation for the DDPM also shows slight increase in this area.

In summary, the standard deviation reported by the cGAN method may be interpreted as the uncertainty in its prediction and can arise from a variety of factors, including out-of-distribution inputs, artifacts, and input ambiguity. Its spatial distribution can also be utilized to locate regions with potential errors. When compared to the DDPM, the cGAN generates more diverse images that are more effective in pointing to uncertainties, covering the potential inaccuracies, and are sensitive to a wider range of ambiguities. It is noteworthy that a deterministic method can, at best, produce an accurate segmentation output similar to the ninth column in Fig. 6. However, this is not sufficient in some cases. For instance, in the cropped image, even if the segmented brain is accurate, a subsequent algorithm that measures the volume of the brain tissue will yield inaccurate results as a significant portion of the brain will be missing. In this scenario, the proposed method warns the end-user by producing higher uncertainty values.

The standard deviation can also be used to provide a consolidated estimate of the reliability of the model output in extracting the brain from a whole head 3D MR image. We demonstrate this by computing an aggregate value of the standard deviation defined as the sum of the standard deviation of voxels with a value greater than a hyper-parameter set to be 0.05. We then normalize this value by dividing it by the number of voxels in the brain mass and denote this value as $\hat{u}$. In Fig. 7, for each test subject of the training datasets, we plot the dice error $\bar{e} = 1 - DSC$ and this aggregate value. We note that these two measures are correlated, and thus, the aggregate standard deviation value $\hat{u}$ can be used by an end-user as a measure of confidence in the generated mask.

This measure is especially useful when the proposed algorithm is used in clinical applications. In these instances, the target brain image and, hence, the dice error are unknown. However, the measure $\hat{u}$ is still calculable through our algorithm. A small value of this measure will provide the end user with higher confidence that the resulting brain image is sufficiently accurate. On the other hand, larger values can warn the end-user for the need for quality control (QC), i.e., to investigate the brain extraction output manually. To demonstrate this, we conducted brain extraction and calculated the uncertainty measure $\hat{u}$ on an independent internal dataset that is not seen by the model during the training (see Section III-G for details).

The histogram of the calculated values for $\hat{u}$ of all subjects is shown in blue in Fig. 8-a. We note that, for better presentation, the values are min-max normalized, i.e., the minimum $(1.1991e^{-3})$, and maximum $(1.2105e^{-2})$ values are set to be zero and one respectively, and the rest of values are linearly scaled In the next step, we employ a Gaussian mixture (GM) model to categorize $\hat{u}$ values into two primary classes, indicated by two Gaussian probability distributions, using maximum likelihood estimation. Fig. 8-b demonstrates the probability of the values belonging to the second class. As can be seen, the values of $\hat{u}$ close to zero, i.e., those with little uncertainty in their prediction, belong to the first group of images indicated as *Accept*. As the uncertainty increases, the probability of images belonging to the second group increases. This corresponds to the group for which quality control is required (denoted by *QC*). This yields the decision boundary indicated by a dashed vertical line in Fig. 8 ($\hat{u}_{QC} = 3.9845e^{-3}$). Subjects with uncertainty higher than this value should be recommended for a manual QC check.

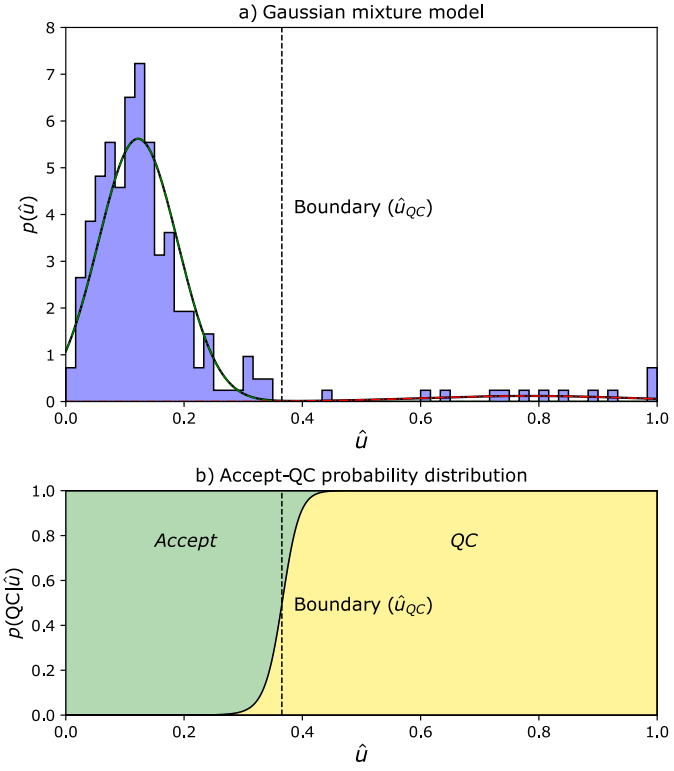

Fig. 8: a) Histogram of normalized values of variation metric $\hat{u}$ and the fitted Gaussian mixture (GM) model. b) The probability of each subject belonging to *Accept* and *QC* groups based on the GM probabilities.

To evaluate the effectiveness of the calculated decision boundary, we conducted an exhaustive QC on the dataset. Out of 249 MR images, 24 yielded $\hat{u}$ greater than $\hat{u}_{QC}$. Out of these, 13 demonstrated significant issues with brain extraction. Of these, 11 instances were scanned with a wrong sequence (are not standard MRI images) due to human error, one had a hyperintense noise signal, manifested as a strip of voxels with spiked values, and one had a minor inappropriate cropping error. All these erroneous images were detected by the algorithm. Among the 225 images that were included in the *Accept* group, no significant issues were observed.

V. Conclusions

In this manuscript, we have developed, implemented, and tested a novel algorithm for brain extraction that is based on Deep Bayesian inference. It utilizes a conditional GAN formulation, where the generator is in the form of a U-Net, and the uncertainty is introduced by the latent variables at multiple scales and for multiple features through conditional instance normalization. The key features of this approach are:

1) Accuracy: The method described in this manuscript yields accuracy metrics that are significantly better than a widely-used brain extraction tool and compare favorably with the best values achieved by the state-of-the-art methods in the literature.
2) Uncertainty quantification: This brain extraction method can generate estimates of uncertainty in its prediction. We also demonstrate how these results can be used to detect regions of likely error within an image and to assess the overall performance of the algorithm.
3) Robustness: We aimed to maximize the heterogeneity of our dataset to evaluate the robustness of our method by combining four datasets. The datasets contain MRI images (a) from healthy subjects and patients with psychiatric symptoms, (b) with different methods of defacing, and (c) obtained from different manufacturers, sequences, and varying contrast and magnetic field strength. We also evaluated the application of uncertainty quantification on a clinical dataset.
4) Speed: The fully trained model can be deployed on a reasonable desktop (with Nvidia GeForce RTX 2080 GPU) and can generate 40 samples in less than a minute (on average, 46 seconds over MR images of size $256 \times 192 \times 170$). The BET-O process takes about nine minutes on the same computer. The diffusion model (DDPM) takes about 16 seconds to generate one sample per slice. This amounts to about 20 hours for 20 samples for a 3D image, which is substantially slower that the cGAN method.

However, despite the above-mentioned features, the cGAN method requires more training time than other methods that utilize U-Nets (approximately 150 hours compared to 45 hours for DCNN and 84 hours for DDPM, using the same hardware). This is because the adversarial loss function (including the gradient penalty term) and having two neural networks make it more expensive and memory-demanding to train. Also, during inference, the model has to generate multiple samples, which needs longer computation time compared to a deterministic model with the same neural network size. It should be noted that the model only needs to be trained once, so this extra cost of training will be amortized over the multiple runs of the network.

We would also like to note that there are certain limitations of the current study that could be addressed in future work and extensions. These include: (a) When comparing with other DL-based brain extraction methods, we did our best to re-train the previous models with the data used to train our model. However, in one instance (HD-BET), the source code was unavailable, and we used the pre-trained model. Also, in another case [61], the input pipeline provided by the authors had to be modified significantly to accommodate heterogeneous data. These difficulties point to the importance of setting up a public database and challenges for brain extraction that could be used by researchers to test their methods against the state-of-the-art in a systematic way. Similar efforts in other areas of imaging are already quite mature; (b) As mentioned earlier, brain extraction is usually the first step in a neurological pipeline that includes downstream tasks like computing the volume of grey and white matter and segmenting brain lesions. It will be of interest to see how the performance of the methods considered in this study translates to performance in these downstream tasks.

Appendix
A: Architecture of Neural networks

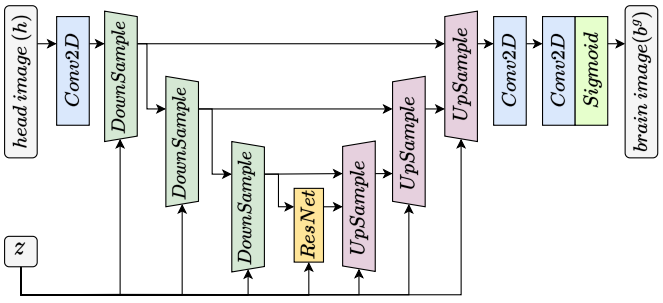

Fig. 9: The U-net based architecture of the generator $\boldsymbol{g}$ in the proposed cGAN. It consists of three convolution layers, three levels of down-sampling and up-sampling, and skip connections. There is a ResNet block that accepts the latent variable $\boldsymbol{z}$ inside each down-sampling and up-sampling block and at the middle of the U-net. The tensor size of the data is also shown. Please refer to Section III-B.2 and Appendix for more details.

The architecture of the generator and critic neural networks discussed in III-B.2 are illustrated in Figures 9 and 10. As shown, they comprise various network blocks. We describe the key components in the following.

a) Down-sampling block: This block is used to reduce the spatial resolution while increasing the number of channels. As depicted in Fig. 11, each down-sampling block consists of a convolution layer with output channels twice the number of the channels of the input, a 2D average pooling layer that reduces the spatial dimensions by a factor of two, and a ResNet block.

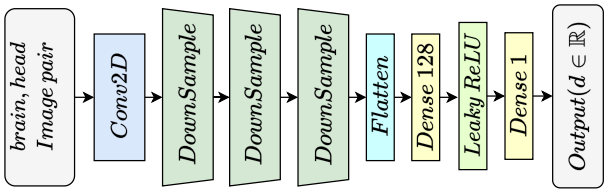

Fig. 10: The critic d neural network architecture, consisting of a convolutional layer, three layers of down-sampling, and two dense layers. d receives a pair of always-real head and a brain image. It produces larger values for real brains and smaller values for generated brains.

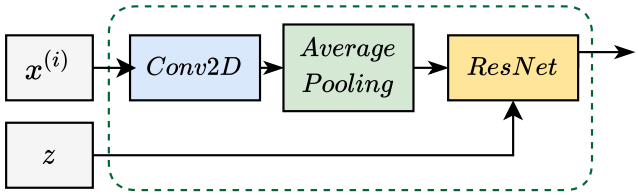

Fig. 11: The down-sampling block.

b) Up-sampling block: Each up-sampling block, shown in Fig. 12, receives the output of the previous block and an output of a down-sampling block of the same spatial size through a skip connection. These tensors are concatenated in the channel dimension. The up-sampling block then performs a convolution that halves the channel size, a 2D up-sampling that increases the spatial dimension by a factor of two, and finally passes the signal through a ResNet block.

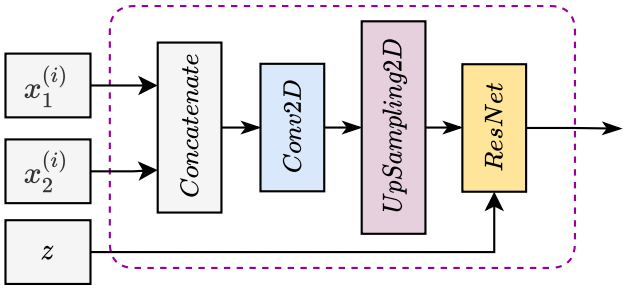

Fig. 12: the up-sampling block.

c) Conditional instance normalization (CIN): The CIN block, depicted in Fig. 13, is used to inject the latent variable $\boldsymbol{z}$ into different levels of the generator's U-Net architecture. It accepts as input $\boldsymbol{z}$ and an intermediate tensor $\boldsymbol{x}^{(i)}$ of size $h \times w \times c$. The CIN block first performs a channel-wise normalization $(Norm)$ of $\boldsymbol{x}^{(i)}$,

$$Norm(\boldsymbol{x}^{(i)})_j = \left(\frac{\boldsymbol{x}_j^{(i)} - \mu\left(\boldsymbol{x}_j^{(i)}\right)}{\sigma\left(\boldsymbol{x}_j^{(i)}\right)} \right), \; j = 1 \cdots c \qquad (12)$$

where $\mu(.)$ and $\sigma(.)$ compute the mean and standard deviation along the spatial directions for a given channel j. Next, the latent vector $\boldsymbol{z}$ of size $1 \times 1 \times N_Z$ is passed through two separate 2D convolution layers, $\boldsymbol{\alpha}(\boldsymbol{z})$ and $\boldsymbol{\beta}(\boldsymbol{z})$, each transforming $\boldsymbol{z}$ to a tensor of shape $1 \times 1 \times c$. The final output of the CIN block is given by the following re-normalization

$$CIN(\boldsymbol{x}^{(i)}, \boldsymbol{z})_j = \boldsymbol{\alpha}(\boldsymbol{z})_j \otimes Norm(\boldsymbol{x}^{(i)})_j \oplus \boldsymbol{\beta}(\boldsymbol{z})_j \,, \qquad (13)$$

where $\otimes$ and $\oplus$ represent element-wise multiplication and summation in the channel direction for $j = 1 \cdots c$, respectively. In other words, CIN redefines the channel-wise mean and standard deviation of an intermediate tensor to new values depending (non-linearly) on the latent signal. Note that an advantage of injecting the latent information in this manner is that the dimension N_Z can be chosen independently of the spatial resolution of the MR image.

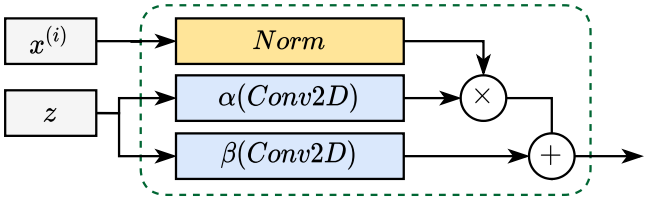

Fig. 13: The conditional instance normalization (CIN) block.

d) ResNet block: Motivated by the network architecture in [11], we implemented a customized ResNet block depicted in Fig. 14. When appearing in the generator, it takes as input $\boldsymbol{z}$ and an intermediate tensor $\boldsymbol{x}^{(i)}$. The inputs pass through two parallel passways, whose results are summed together to give an output tensor that retains the same shape as $\boldsymbol{x}^{(i)}$. Note that when the ResNet block is used in the critic and the U-net in DCNN, it takes as input only $\boldsymbol{x}^{(i)}$ with CIN replaced by layer normalization.

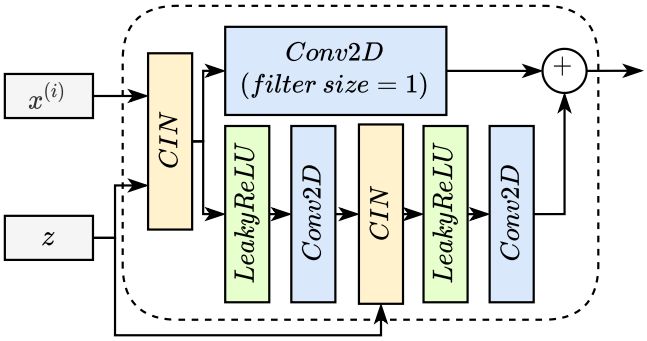

Fig. 14: The customized residual network (ResNet) block.

B: Implementation Notes

We used the TensorFlow [96] library for training and testing our models. All training experiments are done with a batch size of 16 using NVIDIA Geforce RTX 2080 with 8 GB of GPU memory and 64 GB of computer RAM. An early stopping scheme is utilized based on dice similarity coefficient (DSC) calculation on the validation data after each epoch. Our best model was obtained at epoch 883 after about 150 hours of computing time. Furthermore, we used an iterative algorithm for training where the generator weights

were updated after every four updates of the critic weights. We used Adam's [97] amsgrad variant [98] as the optimizer for the training of our model. We also set $\beta_1 = 0.2$ and $\beta_2 = 0.7$ and an initial learning rate of $1.0e-4$ as optimizer hyper-parameters. Additional information and instructions are available at: `https://github.com/bmri/bmri`

C: Comparing Methods' Details

C-I. Deep convolutional neural network (DCNN) model

We implemented a deep convolutional neural network (DCNN) logistic regression model. One can deem a DCNN model as a function approximator, $\boldsymbol{f} : \Omega_H \longmapsto \Omega_B$, that provides the probability that each pixel in the head image belongs to the brain. This is done by training a CNN-based model that is trained in a supervised fashion. That is, unlike the adversarial loss in the cGAN, a binary cross entropy loss function is used to directly measure the difference between the target binary mask brain $\boldsymbol{t}$ and the model output for the input head image, $\boldsymbol{f}(\boldsymbol{h})$, in a pixel-wise binary classification setup:

$$\mathcal{L}(\boldsymbol{t}, \boldsymbol{f}(\boldsymbol{h})) = -\frac{1}{N_1 \times N_2} \sum_{i=1}^{N_1 \times N_2} [\boldsymbol{t}_i \log(\boldsymbol{f}(\boldsymbol{h})_i) + (1 - \boldsymbol{t}_i) \log(1 - f(\boldsymbol{h})_i)], \quad (14)$$

where $\boldsymbol{t}_i$ denotes the pixels of the ground truth binary mask, and $\boldsymbol{f}(\boldsymbol{h})_i$ are pixels of the model output. This loss function penalizes the difference between the corresponding pixels in the ground truth (what should be predicted) and the model's output (what is predicted). The optimized DCNN model is given by

$$\boldsymbol{f}^* = arg\min_{\boldsymbol{f}} \mathcal{L}(\boldsymbol{t}, \boldsymbol{f}(\boldsymbol{h})). \quad (15)$$

That is the model that produces images that are closer to the target mask based on the binary cross entropy measure. Finally, a thresholding filter is applied to the output image to generate the output mask. One should note that the ground truth brain images used to train the cGAN are not converted to binary mask images. The neural network architecture of the DCNN closely resembles the U-net based generator, $\boldsymbol{g}$, from the cGAN. However, it does not include a latent variable, and there is no conditional instance normalization (CIN) in the ResNet blocks. Further, most of the hyperparameters, such as batch size and learning rate, were kept the same as the cGAN model.

C-II. Denoising diffusion probabilistic model (DDPM)

During the training phase of the unconditional version of diffusion models, a datapoint is drawn from the given distribution as $x_0 \sim q(x_0)$. The forward noising process is then defined as producing latent images $x_t, t = 1 \dots T$ through T steps of adding Gaussian noise to the image of the previous step [27] as follows:

$$q(x_t|x_{t-1}) := \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \boldsymbol{I}\right) \quad (16)$$

where $\boldsymbol{I}$ denotes the identity matrix and $\beta_t \in (0,1), t = 1 \dots T$ represent the variances of the added noise in different steps, such that $\beta 0 < \beta 1 < \cdots < \beta t$. Additionally, it is demonstrated in [95] that by introducing $\alpha_t := 1 - \beta_t$ and $\alpha_t := \prod_{s=1}^{t} \alpha_s$, we can obtain the latent image through the direct application of noise to the initial (clear) image x_0 as:

$$q(x_t|x_0) := \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\boldsymbol{I}\right) \quad (17)$$

This further results in the definition of a function to provide x_t as:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \boldsymbol{I}). \quad (18)$$

The noisy images are used to learn the backward denoising process as follows:

$$p_\theta(x_{t-1}|x_t) \sim \mathcal{N}\left(x_t - 1; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right), \quad (19)$$

which in turn is used to define the following denoising formulation:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t \boldsymbol{z}. \quad (20)$$

$\epsilon_\theta(x_t, t)$ is the output of a U-net, parameterized by θ that is trained based on x_t generated from Equation (18) for varying $t \in [1 \dots T]$. $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$ is a random noise, and σ_t is the standard variation of the noise that is learned by the model in this version (See Equations (15) and (16) of Reference [95] for more details). In practice, Equation (20) receives an initial random noise and iteratively reduces the noise to generate a new point sample from the dataset q.

The implemented benchmark used in this work concerns the conditional version of the presented DDPM. Accordingly, the brain mask image $\boldsymbol{m}$ undergoes the forward noising and backward denoising process, while the head image $\boldsymbol{h}$ is used as the condition in the form of an additionally concatenated image channel. Accordingly, the noising step constitutes:

$$\boldsymbol{m}_{\boldsymbol{h},t} = \sqrt{\bar{\alpha}}\boldsymbol{m}_{\boldsymbol{h},0} + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \boldsymbol{I}), \quad (21)$$

and the denoising process takes place as follows:

$$\boldsymbol{m}_{\boldsymbol{h},t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{m}_{\boldsymbol{h},t} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{m}_{\boldsymbol{h},t}, t)\right) + \sigma_t \boldsymbol{z}. \quad (22)$$

In this paper, we used the same data used for training the cGAN model for the DDPM model. Also, we tried to keep the default values of the implementation whenever possible. Accordingly, $T = 1000$ diffusion steps with batch size 16 and learning rate $1e-4$ is used for training. 64 channels were used for the first U-Net level with two res-block units. As mentioned, σ_t is learned by the model, and the attention model's resolution is set to 16.

Acknowledgment

The authors would like to thank the reviewers for their insightful feedback and constructive comments.

REFERENCES

[1] B. Fischl, "Freesurfer," *Neuroimage*, vol. 62, no. 2, pp. 774–781, 2012.

[2] A. M. Dale, B. Fischl, and M. I. Sereno, "Cortical surface-based analysis: I. segmentation and surface reconstruction," *Neuroimage*, vol. 9, no. 2, pp. 179–194, 1999.

[3] M. Jenkinson, C. F. Beckmann, T. E. J. Behrens, M. W. Woolrich, and S. M. Smith, "Fsl," *Neuroimage*, vol. 62, no. 2, pp. 782–790, 2012.

[4] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm," *IEEE transactions on medical imaging*, vol. 20, no. 1, pp. 45–57, 2001.

[5] L. Storelli, M. A. Rocca, E. Pagani, W. Van Hecke, M. A. Horsfield, N. De Stefano, A. Rovira, J. Sastre-Garriga, J. Palace, D. Sima *et al.*, "Measurement of whole-brain and gray matter atrophy in multiple sclerosis: assessment with mr imaging," *Radiology*, vol. 288, no. 2, 2018.

[6] Y. Zhao, B. Ma, P. Jiang, D. Zeng, X. Wang, and S. Li, "Prediction of alzheimer's disease progression with multi-information generative adversarial network," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 3, pp. 711–719, 2020.

[7] S. M. Smith, Y. Zhang, M. Jenkinson, J. Chen, P. M. Matthews, A. Federico, and N. De Stefano, "Accurate, robust, and automated longitudinal and cross-sectional brain change analysis," *Neuroimage*, vol. 17, no. 1, pp. 479–489, 2002.

[8] J. Bernal, S. Valverde, K. Kushibar, M. Cabezas, A. Oliver, and X. Llado, "Generating longitudinal atrophy evaluation datasets on brain magnetic resonance images using convolutional neural networks and segmentation priors," *Neuroinformatics*, vol. 19, no. 3, pp. 477–492, 2021.

[9] O. Commowick, A. Istace, M. Kain, B. Laurent, F. Leray, M. Simon, S. C. Pop, P. Girard, R. Ameli, J.-C. Ferré *et al.*, "Objective evaluation of multiple sclerosis lesion segmentation using a data management and processing infrastructure," *Scientific reports*, vol. 8, no. 1, pp. 1–17, 2018.

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

[11] J. Adler and O. Öktem, "Deep bayesian inversion," *arXiv preprint arXiv:1811.05910*, 2018.

[12] D. Ray, H. Ramaswamy, D. V. Patel, and A. A. Oberai, "The efficacy and generalizability of conditional gans for posterior inference in physics-based inverse problems," *arXiv preprint arXiv:2202.07773*, 2022.

[13] X. Chen, X. Wang, K. Zhang, K.-M. Fung, T. C. Thai, K. Moore, R. S. Mannel, H. Liu, B. Zheng, and Y. Qiu, "Recent advances and clinical applications of deep learning in medical image analysis," *Medical Image Analysis*, p. 102444, 2022.

[14] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[15] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019. [Online]. Available: http://dx.doi.org/10.1561/2200000056

[16] J. Ehrhardt and M. Wilms, "Autoencoders and variational autoencoders in medical image analysis," in *Biomedical Image Synthesis and Simulation*. Elsevier, 2022, pp. 129–162.

[17] D. Moyer, G. Ver Steeg, C. M. Tax, and P. M. Thompson, "Scanner invariant representations for diffusion mri harmonization," *Magnetic resonance in medicine*, vol. 84, no. 4, pp. 2174–2189, 2020.

[18] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[19] K. Ding, M. Zhou, Z. Wang, Q. Liu, C. W. Arnold, S. Zhang, and D. N. Metaxas, "Graph convolutional networks for multi-modality medical imaging: Methods, architectures, and clinical applications," *arXiv preprint arXiv:2202.08916*, 2022.

[20] M. Ghorbani, A. Kazi, M. S. Baghshah, H. R. Rabiee, and N. Navab, "Ra-gcn: Graph convolutional network for disease prediction problems with imbalanced data," *Medical Image Analysis*, vol. 75, p. 102272, 2022.

[21] C. Saueressig, A. Berkley, E. Kang, R. Munbodh, and R. Singh, "Exploring graph-based neural networks for automatic brain tumor segmentation," in *From Data to Models and Back: 9th International Symposium, DataMod 2020, Virtual Event, October 20, 2020, Revised Selected Papers 9*. Springer, 2021, pp. 18–37.

[22] S. K. Zhou, H. N. Le, K. Luu, H. V. Nguyen, and N. Ayache, "Deep reinforcement learning in medical imaging: A literature review," *Medical image analysis*, vol. 73, p. 102193, 2021.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[25] F. Shamshad, S. Khan, S. W. Zamir, M. H. Khan, M. Hayat, F. S. Khan, and H. Fu, "Transformers in medical imaging: A survey," *arXiv preprint arXiv:2201.09873*, 2022.

[26] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[27] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[28] A. Kazerouni, E. K. Aghdam, M. Heidari, R. Azad, M. Fayyaz, I. Hacihaliloglu, and D. Merhof, "Diffusion models for medical image analysis: A comprehensive survey," *arXiv preprint arXiv:2211.07804*, 2022.

[29] J. Wolleb, R. Sandkühler, F. Bieder, P. Valmaggia, and P. C. Cattin, "Diffusion models for implicit image segmentation ensembles," in *International Conference on Medical Imaging with Deep Learning*. PMLR, 2022, pp. 1336–1348.

[30] S. U. Dar, Ş. Öztürk, Y. Korkmaz, G. Elmas, M. Özbey, A. Güngör, and T. Çukur, "Adaptive diffusion priors for accelerated mri reconstruction," *arXiv preprint arXiv:2207.05876*, 2022.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[32] D. Saxena and J. Cao, "Generative adversarial networks (gans) challenges, solutions, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–42, 2021.

[33] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," *Journal of Medical Imaging and Radiation Oncology*, vol. 65, no. 5, pp. 545–563, 2021.

[34] G. Kwon, C. Han, and D.-s. Kim, "Generation of 3d brain mri using auto-encoding generative adversarial networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 118–126.

[35] C. Bermudez, A. J. Plassard, L. T. Davis, A. T. Newton, S. M. Resnick, and B. A. Landman, "Learning implicit brain mri manifolds with deep learning," in *Medical Imaging 2018: Image Processing*, vol. 10574. SPIE, 2018, pp. 408–414.

[36] S. Kaji and S. Kida, "Overview of image-to-image translation by use of deep neural networks: denoising, super-resolution, modality conversion, and reconstruction in medical imaging," *Radiological physics and technology*, vol. 12, no. 3, pp. 235–248, 2019.

[37] M. Boulanger, J.-C. Nunes, H. Chourak, A. Largent, S. Tahri, O. Acosta, R. De Crevoisier, C. Lafond, and A. Barateau, "Deep learning methods to generate synthetic ct from mri in radiotherapy: A literature review," *Physica Medica*, vol. 89, pp. 265–281, 2021.

[38] A. Sharma and G. Hamarneh, "Missing mri pulse sequence synthesis using multi-modal generative adversarial network," *IEEE transactions on medical imaging*, vol. 39, no. 4, pp. 1170–1183, 2019.

[39] H. Guan and M. Liu, "Domain adaptation for medical image analysis: a survey," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 3, pp. 1173–1185, 2021.

[40] "Miccai workshop on domain adaptation and representation transfer: https://link.springer.com/conference/dart."

[41] C. Chen, Q. Dou, H. Chen, J. Qin, and P.-A. Heng, "Synergistic image and feature adaptation: Towards cross-modality domain adaptation for medical image segmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 865–872.

[42] F. Mahmood, R. Chen, and N. J. Durr, "Unsupervised reverse domain adaptation for synthetic medical images via adversarial training," *IEEE transactions on medical imaging*, vol. 37, no. 12, pp. 2572–2581, 2018.

[43] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang, "A review of deep learning based methods for medical image multi-organ segmentation," *Physica Medica*, vol. 85, pp. 107–122, 2021.

[44] M. Arabahmadi, R. Farahbakhsh, and J. Rezazadeh, "Deep learning for smart healthcare—a survey on brain tumor detection from medical imaging," *Sensors*, vol. 22, no. 5, p. 1960, 2022.

[45] E. Giacomello, D. Loiacono, and L. Mainardi, "Brain mri tumor segmentation with adversarial networks," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[46] "Brain tumor segmentation challenge (brats):" http://braintumorsegmentation.org/."

[47] R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng, and A. K. Nandi, "Medical image segmentation using deep learning: A survey," *IET Image Processing*, vol. 16, no. 5, pp. 1243–1267, 2022.

[48] S. Kazeminia, C. Baur, A. Kuijper, B. van Ginneken, N. Navab, S. Albarqouni, and A. Mukhopadhyay, "Gans for medical image analysis," *Artificial Intelligence in Medicine*, vol. 109, p. 101938, 2020.

[49] M. AlAmir and M. AlGhamdi, "The role of generative adversarial network in medical image analysis: An in-depth survey," *ACM Computing Surveys (CSUR)*, 2022.

[50] Y. Skandarani, P.-M. Jodoin, and A. Lalande, "Gans for medical image synthesis: An empirical study," *arXiv preprint arXiv:2105.05318*, 2021.

[51] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *Medical image analysis*, vol. 58, p. 101552, 2019.

[52] H. Ali, R. Biswas, F. Ali, U. Shah, A. Alamgir, O. Mousa, and Z. Shah, "The role of generative adversarial networks in brain mri: a scoping review," *Insights into Imaging*, vol. 13, no. 1, pp. 1–15, 2022.

[53] A. Fatima, A. R. Shahid, B. Raza, T. M. Madni, and U. I. Janjua, "State-of-the-art traditional to the machine-and deep-learning-based skull stripping techniques, models, and algorithms," *Journal of Digital Imaging*, vol. 33, no. 6, pp. 1443–1464, 2020.

[54] H. Z. U. Rehman, H. Hwang, and S. Lee, "Conventional and Deep Learning Methods for Skull Stripping in Brain MRI," *Applied Sciences*, vol. 10, no. 5, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/5/1773

[55] M. Jenkinson, M. Pechaud, S. Smith, and Others, "BET2: MR-based estimation of brain, skull and scalp surfaces," in *Eleventh annual meeting of the organization for human brain mapping*, vol. 17. Toronto., 2005, p. 167.

[56] S. F. Eskildsen, P. Coupé, V. Fonov, J. V. Manjón, K. K. Leung, N. Guizard, S. N. Wassef, L. R. Østergaard, and D. L. Collins, "BEaST: Brain extraction based on nonlocal segmentation technique," *NeuroImage*, vol. 59, no. 3, pp. 2362–2373, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811911010573

[57] R. W. Cox, "AFNI: software for analysis and visualization of functional magnetic resonance neuroimages." *Computers and biomedical research, an international journal*, vol. 29, no. 3, pp. 162–173, jun 1996.

[58] J. Kleesiek, G. Urban, A. Hubert, D. Schwarz, K. Maier-Hein, M. Bendszus, and A. Biller, "Deep MRI brain extraction: A 3D convolutional neural network for skull stripping," *NeuroImage*, vol. 129, pp. 460–469, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811916000306

[59] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[60] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, 2021.

[61] S. S. Mohseni Salehi, D. Erdogmus, and A. Gholipour, "Auto-Context Convolutional Neural Network (Auto-Net) for Brain Extraction in Magnetic Resonance Imaging," *IEEE Transactions on Medical Imaging*, vol. 36, no. 11, pp. 2319–2330, 2017.

[62] F. Isensee, M. Schell, I. Pflueger, G. Brugnara, D. Bonekamp, U. Neuberger, A. Wick, H.-P. Schlemmer, S. Heiland, W. Wick, and Others, "Automated brain extraction of multisequence MRI using artificial neural networks," *Human brain mapping*, vol. 40, no. 17, pp. 4952–4964, 2019.

[63] R. Dey and Y. Hong, "CompNet: Complementary segmentation network for brain MRI extraction," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 628–636.

[64] A. Fatima, T. M. Madni, F. Anwar, U. I. Janjua, and N. Sultana, "Automated 2d slice-based skull stripping multi-view ensemble model on nfbs and ibsr datasets," *Journal of Digital Imaging*, vol. 35, no. 2, pp. 374–384, 2022.

[65] H. Hwang, H. Z. U. Rehman, and S. Lee, "3D U-Net for Skull Stripping in Brain MRI," *Applied Sciences*, vol. 9, no. 3, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/3/569

[66] A. Derin, A. F. Bayram, C. Gurkan, A. Budak, and H. KARATAŞ, "Automatic skull stripping and brain segmentation with u-net in mri database," *Avrupa Bilim ve Teknoloji Dergisi*, no. 40, pp. 75–81, 2022.

[67] A. Hoopes, J. S. Mora, A. V. Dalca, B. Fischl, and M. Hoffmann, "Synthstrip: skull-stripping for any brain image," *NeuroImage*, vol. 260, p. 119474, 2022.

[68] X. Wang, X.-H. Li, J. W. Cho, B. E. Russ, N. Rajamani, A. Omelchenko, L. Ai, A. Korchmaros, S. Sawiak, R. A. Benn *et al.*, "U-net model for brain extraction: Trained on humans for transfer to non-human primates," *Neuroimage*, vol. 235, p. 118001, 2021.

[69] L.-M. Hsu, S. Wang, P. Ranadive, W. Ban, T.-H. H. Chao, S. Song, D. H. Cerri, L. R. Walton, M. A. Broadwater, S.-H. Lee *et al.*, "Automatic skull stripping of rat and mouse brain mri data using u-net," *Frontiers in neuroscience*, vol. 14, p. 568614, 2020.

[70] G. Ruan, J. Liu, Z. An, K. Wu, C. Tong, Q. Liu, P. Liang, Z. Liang, W. Chen, X. Zhang *et al.*, "Automated skull stripping in mouse fmri analysis using 3d u-net," *bioRxiv*, pp. 2021–10, 2021.

[71] L. Pei, M. Ak, N. H. M. Tahon, S. Zenkin, S. Alkarawi, A. Kamal, M. Yilmaz, L. Chen, M. Er, N. Ak *et al.*, "A general skull stripping of multiparametric brain mris using 3d convolutional neural network," *Scientific Reports*, vol. 12, no. 1, p. 10826, 2022.

[72] S. Mayala, I. Herdlevær, J. B. Haugsøen, S. Anandan, N. Blaser, S. Gavasso, and M. Brun, "Gubs: Graph-based unsupervised brain segmentation in mri images," *Journal of Imaging*, vol. 8, no. 10, p. 262, 2022.

[73] D. Ray, J. Murgoitio-Esandi, A. Dasgupta, and A. A. Oberai, "Solution of physics-based inverse problems using conditional generative adversarial networks with full gradient penalty," *arXiv preprint arXiv:2306.04895*, 2023.

[74] K. Zou, Z. Chen, X. Yuan, X. Shen, M. Wang, and H. Fu, "A review of uncertainty estimation and its application in medical imaging," *arXiv preprint arXiv:2302.08119*, 2023.

[75] Y. Yang, X. Guo, Y. Pan, P. Shi, H. Lv, and T. Ma, "Uncertainty quantification in medical image segmentation with multi-decoder u-net," in *International MICCAI Brainlesion Workshop*. Springer, 2021, pp. 570–577.

[76] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. Eslami, D. Jimenez Rezende, and O. Ronneberger, "A probabilistic u-net for segmentation of ambiguous images," *Advances in neural information processing systems*, vol. 31, 2018.

[77] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann, "Conditional image generation with score-based diffusion models," *arXiv preprint arXiv:2111.13606*, 2021.

[78] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

[79] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in neural information processing systems*, vol. 32, 2019.

[80] A. Rahman, J. M. J. Valanarasu, I. Hacihaliloglu, and V. M. Patel, "Ambiguous medical image segmentation using diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 536–11 546.

[81] L. Zbinden, L. Doorenbos, T. Pissas, R. Sznitman, and P. Márquez-Neila, "Stochastic segmentation with conditional categorical diffusion models," *arXiv preprint arXiv:2303.08888*, 2023.

[82] Y. Song and S. Ermon, "Improved techniques for training score-based generative models," *Advances in neural information processing systems*, vol. 33, pp. 12 438–12 448, 2020.

[83] A. Almahairi, S. Rajeshwar, A. Sordoni, P. Bachman, and A. Courville, "Augmented cyclegan: Learning many-to-many mappings from unpaired data," in *International Conference on Machine Learning*. PMLR, 2018, pp. 195–204.

[84] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[85] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[86] C. Villani, *The Wasserstein distances*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 93–111. [Online]. Available: https://doi.org/10.1007/978-3-540-71050-9-6

[87] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," 2017.

[88] "https://scikit-image.org/docs/stable/api/skimage.filters."

[89] "https://scikit-image.org/docs/stable/api/skimage.morphology."

[90] B. Puccio, J. P. Pooley, J. S. Pellman, E. C. Taverna, and R. C. Craddock, "The preprocessed connectomes project repository of manually corrected skull-stripped T1-weighted anatomical MRI data," *GigaScience*, vol. 5, no. 1, 2016. [Online]. Available: https://doi.org/10.1186/s13742-016-0150-5

[91] R. Souza, O. Lucena, J. Garrafa, D. Gobbi, M. Saluzzi, S. Appenzeller, L. Rittner, R. Frayne, and R. Lotufo, "An open, multi-vendor, multi-field-strength brain mr dataset and analysis of publicly available skull stripping methods agreement," *NeuroImage*, vol. 170, pp. 482–494, 2018.

[92] D. W. Shattuck, M. Mirza, V. Adisetiyo, C. Hojatkashani, G. Salamon, K. L. Narr, R. A. Poldrack, R. M. Bilder, and A. W. Toga, "Construction of a 3d probabilistic atlas of human cortical structures," *Neuroimage*, vol. 39, no. 3, pp. 1064–1080, 2008.

[93] T. Rohlfing, "Image similarity and tissue overlaps as surrogates for image registration accuracy: widely used but unreliable," *IEEE transactions on medical imaging*, vol. 31, no. 2, pp. 153–163, 2011.

[94] "fsl bet user guide:" https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide."

[95] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8162–8171.

[96] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[97] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[98] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.