Exploring graph-based neural networks for automatic brain tumor segmentation

Camillo Saueressig[1,2], Adam Berkley[1], Elliot Kang[1], Reshma Munbodh[3★], and Ritambhara Singh[1,2,∗]

[1] Department of Computer Science, Brown University
[2] Center for Computational Molecular Biology, Brown University
[3] Department of Radiation Oncology, Brown Alpert Medical School
{reshma_munbodh,ritambhara}@brown.edu

Abstract. Manual evaluation of medical images, such as MRI scans of brain tumors, requires years of training, is time-consuming, and is often subject to inter-annotator variation. The automatic segmentation of medical images is a long-standing challenge that seeks to alleviate these issues, with great potential benefits for physicians and patients. In the past few years, variations of Convolutional Neural Networks (CNNs) have established themselves as the state-of-the-art methodology for this task. Recently, Graph-based Neural Networks (GNNs) have gained considerable attention in the deep learning community. GNNs exploit the structural information present in graph data by aggregating information over connected nodes, allowing them to effectively capture relation information between data elements. In this project, we propose a GNN-based approach to brain tumor segmentation. We represent 3D MRI scans of the brain as a graph, where different regions in the images are represented by nodes and edges connect adjacent regions. We apply several variations of GNNs for the automatic segmentation of brain tumors from MRI scans. Our results show GNNs give reasonable performance on the task and allow for realistic modeling of the data. Furthermore, they are far less computationally expensive and time-consuming to train than state-of-the-art segmentation models. Lastly, we assign Shapley value-based contribution scores to input MRI features to learn what features are relevant for a particular segmentation, generating interesting insights into explaining the predictions of the proposed model.

Keywords: graph neural networks · brain tumor segmentation · deep learning

1 Introduction

Over 87,000 people are expected to be diagnosed with brain tumors in 2020 [19]. With a low survival rate for malignant tumors, timely detection and diagnosis of brain tumors are crucial for developing effective treatment plans for the patients. Neuroimaging using multimodal magnetic resonance imaging (MRI) is integral in the diagnosis and management of brain tumors, including for surgical and radiation treatment planning, longitudinal tumor monitoring, treatment response evaluation, and predictive analysis.

[★] Corresponding Authors

These require accurate delineation of the tumor boundary on the MRI images to characterize the tumors.

Automatic tumor segmentation methods seek to address the time and inter-observer variability limitations posed by manual segmentation. Furthermore, they underlie advances in quantitative tumor analysis and clinical workflow automation. The development of such automatic segmentation methods is challenging due to several intrinsic and extrinsic factors, such as the heterogeneity in appearance and shape of different tumor types on MRI, a lack of standardized imaging protocols, variability in equipment, and the presence of imaging noise and artifacts. Furthermore, advances in neuroimaging and the clinical management of brain tumors have increased the desired complexity of the segmentation, with an emphasis on a compartmentalized segmentation of the tumor into sub-regions describing necrosis, enhancing and non-enhancing tumor and vasogenic edema.

The use of deep learning methods for brain tumor segmentation has progressed rapidly in the past few years [4,16]. As opposed to conventional segmentation models that rely on the extraction of pre-defined features from the images [10,14,20], deep learning models automatically learn relevant features to perform accurate segmentation. However, current deep learning segmentation methods [9,8,34,18] are computationally intensive, require the division of the images into local patches, and do not explicitly account for brain connectivity information. They fail to capture the global structure of the 3D images adequately or relational dependencies between different regions in the tumor. We hypothesize that these properties are important for accurate and robust brain tumor segmentation.

We propose using Graph-based Neural Networks (GNNs) to segment brain tumors from multimodal 3D MRI. Unlike previous methods, GNNs allow for the processing of the entire brain simultaneously, while explicitly incorporating both local and global connectivity into their predictions by aggregating information across neighboring nodes in the graph. As such, GNNs effectively capture relational information between the data elements. Our framework, summarized in Fig. 1, first represents the 3D MRI scans of the entire brain as a graph, where nodes represent different regions in the images and edges connect adjacent regions. Next, a GNN classifies each node of the graph into healthy tissue, enhancing tumor, necrotic tissue and non-enhancing tumor, or edema. The node predictions are subsequently mapped back to their respective supervoxels on the MRI. We explore different GNN models for brain tumor segmentation from MRI scans on the BraTS 2019 challenge [16,4,3]. The best performing model achieves good performance that is comparable to other recent work. We also show that our approach is between 5 and 15 times faster than such computationally intensive methods. Finally, we provide explanations for the predictions of the deep learning GNN models in terms of the relative contributions of the inputted MRI modalities. We generate these explanations via Shapley values, a game-theoretic approach for fairly attributing contributions to an overall outcome among the game participants. Such interpretations are vital for applications of these models in the health domain.

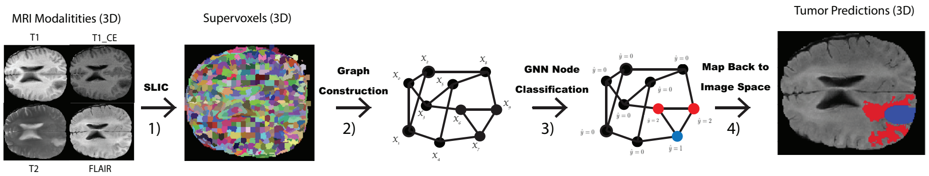

Fig. 1: **Model Overview.** MRI Modalities are first stacked to create one 3D Image with 4 channels. 1) Combined modalities are clustered into supervoxels. 2) Supervoxels are converted to a graph structure such that each supervoxel becomes one graph node. 3) Graph is fed through a Graph Neural Network, which predicts a label for each node. 4) Node predictions are overlaid back onto the supervoxels.

2 Related Work

2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) have, so far, been the most successful models for fully automatic brain tumor segmentation. They excel at object classification and segmentation tasks by classifying pixels based on surrounding image content through 2D or 3D convolutional filters. These convolutional filters are translation invariant and can detect image edges and combine them into higher-level image features, making them well suited for image processing. The three best performing models of the recent 2018 BraTS Challenge [16,3,4] all consisted of CNN-based architectures. The BraTS challenge is a brain tumor segmentation competition where teams submit their models for testing on a multi-institutional database of MRI scans. The best performing model by Myronenko *et al.* [18] used an autoencoder-based regularization with a 3D-CNN to achieve state-of-the-art segmentation results. The next best-performing work by Isensee *et al.* [9] proposed that a well trained baseline 3D U-net could outperform other models with various architectural modifications. Finally, McKinley *et al.* [15] used a CNN with contextual and attentive information and tied for third place with Zhou *et al.* [34] who used a U-net with a novel loss function that modeled noise and uncertainty.

These CNN-based architectures take an extended amount of time to train, and many have harsh GPU-requirements. The best performing model requires 34GB of VRAM [18] and most require anywhere from 8 to 12 GB of VRAM. Combined with the training times greater than a week, this constitutes a resource bottleneck on training and evaluating models on new datasets. Furthermore, these models generally require the division of the images into local patches for training and segmentation and, hence, fail to capture the global information of the entire MRI scan.

2.2 Graph Neural Networks

The computational burden of segmentation with CNNs can be circumvented by summarizing the MRI images as a graph representation. This approximation reduces image complexity by two orders of magnitude, from millions of voxels down to several thousand nodes, while preserving most image information. A recently popularized form of deep learning, Graph Neural Networks, is specifically designed to learn over such graph structures. The theoretic underpinnings of learning on graphs have been established for close to a decade [7,22], but GNNs have only recently seen widespread use following, among others, Kipf and Wellings's [12,5,35] introduction of the graph convolutional network (GCN). Their work refined the convolution operation on graph-structured data and established a layerwise approach to learning over graphs, thus aligning it more closely to existing deep learning paradigms.

Subsequently, Hamilton *et al.* [6] developed GraphSAGE, which extends GCN [12] by generalizing graph learning as a series of alternating sampling and aggregation steps to share information across a graph. In a GraphSAGE layer, for each node, a predefined number of neighbors are sampled. Their information is aggregated by combining their features and applying a learnable transformation, the output of which becomes the node's features in the next layer. Notably, GraphSAGE allows GNNs to be extended to the inductive setting, generalizing to previously unseen graphs.

The Graph Attention Network (GAT) developed by Velickovic *et al.* [29] introduced the self-attention mechanism to graph learning. Self-attention is an operation which allows each input feature to assign weights, or "attend", differently to the other input features, and has shown the state-of-the-art performance on natural language processing (NLP) and other tasks (Vaswani *et al.* 2017) [28]. In the GAT formulation, attention is instead computed between each graph node and its neighbors. Like GraphSAGE, GAT readily allows for inductive learning and gives the state-of-the-art performance on an inductive protein-protein interaction (PPI) task.

GNNs have previously been applied to medical image segmentation tasks. Yan *et al.* 2019 [32] successfully applied a GCN variant, ChebNet, to segment brain tissue (gray matter, white matter, cerebro-spinal-fluid). They first used the SLIC algorithm [1] to cluster MRIs into supervoxels, and then predicted the tissue type of each supervoxel. The present work is partially inspired by their approach and follows a similar workflow. Juarez *et al.* [11] proposed a joint U-Net-GNN model for airway segmentation from CT scans and matched the state-of-the-art performance. They replaced the last two layers of a U-Net with a sequence of graph convolution layers, which allowed the model to aggregate information globally across the entire CT scan while maintaining the pattern-recognition capabilities of the early convolutional layers. However, GNN-based methods have not previously been attempted for brain tumor segmentation, and thus, we here explore the applicability and performance of several GNN variants on the same.

2.3 Explanation of Deep Learning models

Many interpretation methods for deep learning fall under the umbrella of *saliency maps* [23,27,26]. These methods utilize the gradients computed by a model with respect to the input to highlight regions of interest, i.e., those where the output changes greatly in response to small input changes. Saliency maps are especially useful in image processing, as they allow for easy visualization of pixel saliency and visual interpretation of results. However, one shortcoming of saliency maps is that they are often driven by the input image and largely agnostic to the model. In particular, it has been shown that the saliency outputs for a model trained on random labels can closely resemble those of a legitimate model, indicating that the saliency map is less a reflection of the model than of the input [2].

An interpretability method explicitly developed for GNNs is GNNExplainer [33]. GNNExplainer learns a mask on both the edges and features of an input graph to build a subgraph that seeks to summarize the connections and features that lead to the prediction on a node of interest. Unlike more general methods, GNNExplainer allows for interpreting how graph connectivity factors into a GNN prediction. A drawback of GNNExplainer is that it is difficult to optimize for larger subgraphs. We find that information from nodes far away from the target node often contributes to a prediction for tumor segmentation. Consequently, GNNExplainer was unfortunately unable to build meaningful subgraphs.

In this work, we interpret our results using SHAP values [13]. SHAP values are a computational approximation of Shapley values, a method for assigning payouts to players in a cooperative game, or in this case, contribution values to features in a prediction task. SHAP values maintain many of the theoretical properties of Shapley values, such as additivity and consistency, which make them attractive as a interpretative tool. Section 3.7 presents the details of the SHAP values.

3 Methods

In this section, we first introduce the dataset we use and associated pre-processing followed by a description of transforming patient images into a graph structure. Subsequently, we present in greater detail our experimental setup. Finally, we describe our use of SHAP values to help interpret the results of the proposed model.

3.1 Imaging Data

The imaging data used in this study, including ground truth annotations, were obtained from the training data of the BraTS 2019 challenge [3,4,16]. The dataset consists of 76 low-grade glioma and 259 high-grade glioma MRIs from 19 contributing institutions. Each sample is composed of four imaging modalities obtained from the same patient: T2-weighted fluid attenuated inversion recovery (Flair), T1-weighted (T1), T1-weighted contrast-enhanced (T1CE), and T2-weighted (T2), which provide complementary information about the tumor. All provided imaging data has been skull-stripped, normalized to a resolution of $1\,\mathrm{mm}^3$, and spatially aligned to the other modalities for the same patient [4]. Domain experts manually segmented the provided ground truth annotations following a standardized annotation protocol, and they were further reviewed for consistency and accuracy by additional neuro-radiologists. The ground truth annotation labels were as follows:

Label 0 Normal brain tissue

Label 1 Volume comprising the necrotic core and non-enhancing gross tumor abnormality

Label 2 Vasogenic edema
Label 4 Active core or enhancing region within the gross tumor abnormality

Label 3 (non-enhancing tumor) was removed from the competition as a distinct region. Instead, it was combined with Label 1 (necrotic tumor) because the BraTS organizers found that it can be subject to significant inter-annotator variance and therefore introduce a bias into the ground truth segmentation based on the annotating institution [4].

For this paper's purposes, one set of MRIs (all four modalities) from the same patient is referred to as a patient sample.

3.2 Data Preprocessing

Before segmentation, each MRI is cropped to the tightest possible bounding box of the brain tissue. This step is accomplished by excluding all image planes where all voxels have zero intensity. Next, we standardize each modality separately to a mean of zero and a standard deviation of one. Bias correction of the MRIs did not improve performance, so we report our final results without bias correction (two-sided t-test, $p \approx 1$).

3.3 Graph Construction

In order for the patient samples to be used as training examples for a GNN, they must first be converted to graph representations (Fig. 1 Step 2). To create the graph nodes, all four MRI modalities are concatenated to create one 3D image with four channels. The combined image is then fed through the Simple Linear Iterative Clustering (SLIC) algorithm [1] to generate a set of k supervoxels, where k is a tunable parameter of the SLIC algorithm. SLIC uses a K-means approach to cluster voxels that are similar in both intensity values and physical location in the brain (Eq. 1). In the concatenated MRI images, the spatial distance between two voxels is simply the 3D Euclidean distance between their coordinates. The intensity distance is the Euclidean distance calculated across all four intensity channels. A compactness parameter, m, controls the trade-off between intensity and spatial information.

The distance, D, calculation between two voxels i and j used for the supervoxel clustering thus becomes

$$D = \sqrt{{d_I}^2 + \left(\frac{d_s}{S}\right)^2 m^2} \tag{1}$$

$$d_I = \sqrt{(I_{T1,i} - I_{T1,j})^2 + (I_{T1CE,i} - I_{T1CE,j})^2 + (I_{T2,i} - I_{T2,j})^2 + (I_{FLAIR,i} - I_{FLAIR,j})^2}$$

$$d_s = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

where x,y,z are the spatial position of a voxel in image coordinates, I is the intensity value of a modality at that pixel, and S is the expected spacing between supervoxels.

After clustering, supervoxels outside of the brain mass, that is, those with zero intensity, are filtered out, typically reducing the number of supervoxels by a factor of 2. Each remaining supervoxel is then assigned a feature vector consisting of the 10th, 25th, 50th, 75th, 90th percentiles of its constituent voxels' intensity values across all four modalities. This formulation results in a feature vector of length 20 for each supervoxel. We chose to use quantiles as it empirically performed better than only the mean intensity. Each supervoxel is also assigned a label, which is determined by finding the most common label (mode) of all of its constituent voxels in the ground truth labeling.

To determine the appropriate values of k and m used in constructing the graphs, we calculated the achievable segmentation accuracy (ASA) of several different combinations of values on a subset of the patient samples. The ASA quantifies how well the SLIC supervoxels recover the ground truth segmentation. This metric is equivalent to our model's accuracy at the voxel level when it predicts every supervoxel correctly. Because of the class imbalance skewing towards healthy tissue, we only consider the tumorous region when computing ASA. These results are presented in Section 4.1.

Once the supervoxels are generated for a patient sample, they are used to construct a regular graph. The graph takes the form $\{N, E\}$, where N is the set of vertices (referred to here as nodes), and E is the set of edges between them. Each node in the graph corresponds to exactly one generated supervoxel and is represented by its feature vector and its label (during training). The edge set E captures proximity information between nodes and is composed of undirected and unweighted edges constructed between each supervoxel and the r supervoxels spatially closest to it in the patient sample, where r represents the desired degree of the graph. We define the distance between two supervoxels as the Euclidean distance between the centroids of their constituent voxels' x-y-z coordinates.

3.4 GNN Details

We evaluated several standard GNN models on their ability to segment the tumors: GCN [12], GAT [29], and the gcn, mean, and pool variants of GraphSAGE [6]. In broad terms, each model is composed of individual layers that share information across adjacent nodes. That is, each layer updates each node's feature vector as a transformed combination of its own features and those of its neighbors. As in a standard neural network, an arbitrary number of these layers can then be stacked sequentially. As the number of layers increases, the nodes eventually indirectly receive information from nodes outside of their immediate neighborhood. The mathematical formulations of each of these graph learning layers are shown in equations 2 through 5.

In each case, $h_u^{(l)}$ is the features of node u at layer l, σ is a differentiable, non-linear activation function, $W^{(l)}$ is a layer specific trainable weight matrix, $||$ is the concatenation operator, and $V(u)$ is the subset of nodes which are are connected to u via the edge set E, also known as the neighborhood of u.

GCN/GS-gcn:

$$h_u^{(1+1)} = \sigma(\frac{1}{q} W^{(l)} \cdot (h_u^{(l)} + \sum_{v}^{V(u)} h_v^{(l)})) \tag{2}$$

where q is a normalization constant that differs between formulations from Kipf *et al.* [12] and Hamilton *et al.* [6]. In the case of a regular graph as considered here, however, q is equal to r, the graph degree, for both.

GS-mean:

$$h_u^{(l+1)} = \sigma(W^{(l)} \cdot (h_u^{(l)} \mid\mid mean(h_v^{(l)} \; \forall \; v \in V)) \tag{3}$$

GS-pool:

$$h_u^{(l+1)} = \sigma(W^{(l)} \cdot (h_u^{(l)} \mid\mid max(\sigma(W_{pool} \cdot h_v^{(l)}) \; \forall \; v \in V(u))) \tag{4}$$

where W_{pool} is a global trainable weight matrix.

GAT:

$$h_u^{(l+1)} = \Big\|_{b}^{B} \sigma(\sum_{v \in V(u)} a_{uv}^{b} W_b^{(l)} h_u) \tag{5}$$

where B are multiple attention heads per layer, which each compute their own pairwise self-attention (a_{uv}^{b}) between each pair of neighboring nodes u and v. Here, we use ReLU as the non-linear activation function for all models.

3.5 Training and Evaluation Metrics

Prior to training, each patient sample is converted to a graph as described in section 3.3. We split the dataset into training (60%), validation (20%), and test sets (20%).

The input to the GNN is defined formally as a graph of the form $\{N, E\}$, and a feature matrix $H \in \mathbb{R}^{n \times f}$, where n is the number of nodes, and f is the number of features per node. $f = 20$ for all experiments, as described in section 3.3. The output is of size $n \times c$, where for each graph node, the model returns the probability of that node belonging to each of the four classes (c) defined in Section 3.1

To determine the best hyperparameters for each of the GNN variants, we perform a random hyperparameter search on the validation set. We sweep over regularly spaced intervals of learning rate from 0.00001 to 0.001, feature dropout between 0 and 0.5, model depth from 3 to 9, and hidden layer size between 64 and 256. For GAT models, we additionally examine attention dropout between 0 and 0.5 and attention heads between 3 and 10 for each layer.

Each model is trained to minimize node-wise multi-label cross-entropy loss (Eq. 6) on the validation set using the Adam optimizer. The class weights are adjusted to be inversely proportional to their prevalence in the test set to address the class imbalance.

$$\text{Loss} = \sum_{c=0}^{C} (\mathbf{1}_{c=y}) w_c log(\hat{p}_y) \tag{6}$$

where C are the possible classes, w_c is the class weight, y is the true label, $\mathbf{1}_{c=y}$ is an indicator function, and $\hat{p}_y$ is the predicted probability of that label.

Upon convergence, each model is evaluated on the average Dice scores of its predictions, as shown in Eq. 7.

$$\text{Dice} = \frac{2TP}{2TP + FP + FN}. \tag{7}$$

where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively. True positive voxels are defined as those correctly assigned as belonging to a specific tumor compartment.

Specifically, we calculate the Dice score for the following tumor subregions: Whole Tumor (WT: union of labels 1,2,4), Core Tumor (CT: 1,4), and Active Tumor (AT: 4). These metrics provide insight into the ability of the model to assess tumor shape correctly as well as to differentiate between the different tumor subregions. To allow for direct comparison to published models in the literature, we report voxelwise Dice scores, rather than the Dice score on node (supervoxel) classification.

After the best hyperparameters have been selected for each GNN model, we train a final model on the combined training and validation sets and evaluate it on the test set. All models were implemented in PyTorch using Deep Graph Library (DGL) [31].

3.6 Baseline method

We use the popular U-net model as a baseline to which to compare the results obtained with the GNN models. The top-performing 3D-CNN model [18] of the BraTS2018 [16,3,4] competition uses state-of-the-art GPUs with 34GB of VRAM that were not easily accessible. Therefore, we selected the second-best model, nnU-net [9], since it requires only 11GB VRAM and is easily trainable through an included Python module and available code. Both GNN and CNN models were trained using the same train and test data sets.

3.7 Model Interpretation

In addition to accurately segmenting brain tumors, it is vital that we understand how and why our models make their predictions. Model interpretation allows us to 1) ensure that a model learned robust and generalizable features by cross-referencing important features with known predictive ones, and 2) identify novel features that aid in tumor segmentation. One method for assigning the contribution scores of the input features for a model is to compute Shapley values. The concept of Shapley values is borrowed from Game Theory. It corresponds to a fair payout to all the players in a cooperative game, given the outcome of the game. In the case of a predictive model, Shapley values can be interpreted as the contribution of each input feature towards the prediction of the model. Formally, they are defined as the average marginal contribution of a feature to a given prediction when added to a subset of other features, over all possible subsets [17]. Since the complexity of computing exact Shapley values is combinatorial in the number of features, we instead use the DeepSHAP model [13] to approximate them. This method takes in a background feature distribution and a query prediction it seeks to explain, and assigns each feature a score representing its contribution to the model output. The method calculates the difference in model output when given the true features versus background features. Next, it backpropagates this difference back to each of the input features in a way that satisfies the properties of additivity, consistency, and local accuracy [25]. The backpropagated value at each feature can then be considered the part of the difference it is ‘responsible’ for.

The background feature distribution is obtained by randomly sampling 500 nodes across the entire dataset of input graphs such that the relative proportions of node labels remain consistent. Since predictions on nodes cannot be made in isolation (i.e., they rely on the graph structure and surrounding nodes), SHAP values are computed for each node in a graph simultaneously.

4 Results

4.1 Supervoxel generation affects achievable accuracy

The graph construction step involves two parameters, the choice of the number, k, and compactness, m, for the supervoxel generation via SLIC. We find that $k = 15000, 20000$ and $m = 0.1$ led to the highest ASA (Supplementary Fig. S1). We choose $k = 15000$ for all subsequent experiments as $k = 20000$ required longer to train with no noticeable improvement in performance.

Of note, even the best SLIC parameters result in an ASA of only 0.9, on average (Supplementary Fig. S1). The diminished accuracy is caused by SLIC-generated supervoxels, which encompass voxels of multiple different labels. A drawback of clustering into supervoxels is that it approximates the brain as a collection of homogeneous regions, while each supervoxel may be somewhat heterogeneous. This effect is especially pronounced along the borders between tumor subtypes and regions with low contrast. Here, the transition in intensity across the different modalities and the ground truth labels may not be well aligned, or the intensity differences are gradual, while the shift in labels is abrupt. In these cases, supervoxels are created with a mixture of labels, yet can only be labeled as one of them.

The partial volume effects introduced by supervoxel creation adversely affect the performance of our model. As shown in Supplementary Fig. S2, the voxel-wise Dice score achieved by our model are significantly lower than the supervoxel-wise Dice score across all tumor regions for both the training and testing data.

4.2 Brain tumor segmentation performance of different GNN models

We summarize the segmentation results of the different GNN models on the test set in Table 1. The best performing GNN is a GraphSAGE-pool network with 5 hidden layers of 256 units each, which is trained until convergence at a learning rate of 0.0001. The mean aggregator function performs slightly worse than the pooling operator. The worst performing models by a substantial margin are the GCN models. We hypothesize that this is because they lack the implicit skip connection built into the mean and pooling aggregators via the concatenation step. These results are consistent with those reported by both Velickovic *et al.* [29] and Hamilton *et al.* [6] for the performance trend on protein-protein interaction (PPI) dataset. Surprisingly, GAT performs much worse on this task than GraphSAGE-pool, despite demonstrating improved performance on other inductive tasks. Several factors could account for this discrepancy, including a larger average graph size, less expressive node feature vectors, the different label classification scheme, or simply because attention may be less suited for brain segmentation.

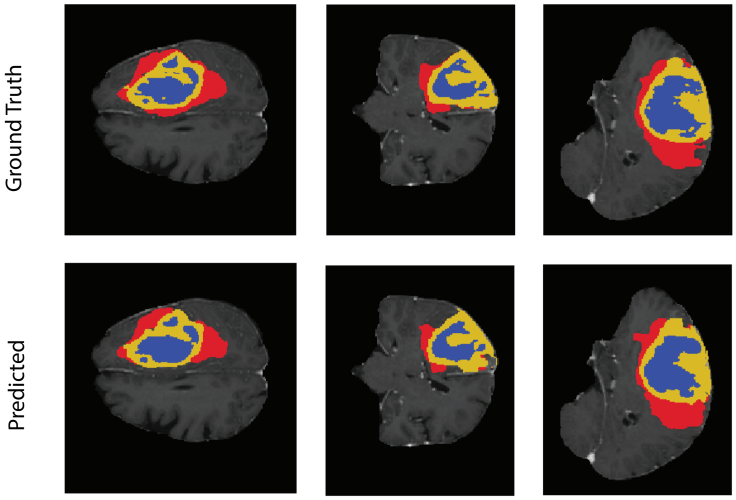

Fig. 2: An example segmentation produced by the best-performing GNN model vs. the ground truth segmentation. Shown are an example horizontal, coronal, and sagittal slice of the same MRI. The colors correspond to the different tumor subtypes: blue = NET/necrosis, yellow = ET, red = Edema. Tumor predictions are overlaid onto the T1-Contrast Enhanced Image. There is a close correspondence between the predicted tumor and the ground truth.

We note that our best performing model is deeper than those reported in previous works [12,29,6], with 5 hidden layers, rather than 2 or 3. We hypothesize that aggregating information from further away is more important for tumor segmentation than other graph learning applications, such as social networks or PPI.

4.3 Performance and runtime results for varying neighborhood sizes

For the best performing model, GraphSAGE-pool, we compare model performance on datasets with varying graph degrees. We create three different sets of graphs from the raw MRIs, with identical node features but either 10, 20, or 30 neighbors. These results are reported in Table 2. While increasing graph degree has no noticeable effect on model performance on the training set, a higher degree does seem to allow the model to generalize better to the unseen data in the test set. However, this comes at the cost of increased training time, with the degree 30 dataset requiring about twice as long to finish training as the degree 10 dataset.

4.4 Comparison of GNN model with Other Recent Models

Next, we compare the GraphSAGE pool model trained on graphs of degree 30 to nnU-Net, the second place model in the BraTS 2018 competition [9]. Both models are trained and evaluated on the same train and test splits. These results are presented in Table 3.

Table 1: Average Dice coefficients across different GNN models for whole tumor (WT), enhancing tumor (ET), and tumor core (TC) trained and evaluated on same train-test split from the training set of the BraTS 2019 data set [16].

Model	WT Dice	TC Dice	ET Dice
GraphSAGE-pool	0.841	0.737	0.671
GraphSAGE-mean	0.804	0.720	0.70
GraphSAGE-gcn	0.536	0.483	0.302
GCN	0.564	0.455	0.341
GAT	0.742	0.687	0.588

Table 2: Average Whole Tumor Dice on training and test sets, along with training time in hours, for GraphSAGE pool models trained and evaluated on graphs of varying degrees.

Model	Train WT Dice	Test WT Dice	Time to Train (hours)
GSpool-10	0.917	0.819	8.7
GSpool-20	0.912	0.832	10.2
GSpool-30	0.915	0.841	15.5

While our GNN model fails to match the state of the art performance of the nnU-Net, the results nonetheless show that GNNs can successfully perform the segmentation task, despite the approximations made in graph construction and the relative novelty of inductive graph-learning techniques. In particular, for the segmentation of the whole tumor, our model achieves a median Dice score that is quite close to nnU-Net. This result indicates that 1) our model is better at outlining the gross tumor than at identifying tumor subregions, and 2) while on most patient samples, GNN models are quite effective, it fails to generalize for some, adversely affecting the mean more than the median.

Our GNN-based approach compares favorably to many other experimental techniques submitted to the BraTS challenge in recent years. Serrano-Rubio *et al.* [24] also attempt a supervoxel-based technique, coupled with Extremely Randomized Trees, to achieve Dice scores of 0.80, 0.63, and 0.57 on the official 2018 validation dataset [4] for whole tumor, core tumor, and enhancing tumor, respectively. Another group, Rezaei *et al.* [21], presents a novel Generative Adversarial Network (GAN) termed voxel-GAN, which seeks to address the label imbalance present in tumor segmentation. This model achieves mean Dice scores of 0.84, 0.79, and 0.63 on the BraTS 2018 validation set. Like ours, these models may not achieve state-of-the-art performance, but identify an important issue in tumor segmentation and attempt to solve it using a novel approach.

Moreover, GNNs' running requirements are relatively modest. Each GNN model was trained on 6 GB of GPU memory with a batch size of 4 brains within hours (Table 2). By contrast, [18] and [9] require 32GB and 12GB of RAM, take days to weeks to train to completion, and are limited to a batch size of 1 and 2 image *patches*, respectively. The eased computational burden could be an important consideration when developing online segmentation models that are regularly updated with new MRIs.

Table 3: Results on our test set (a partition of the BraTS2019 training set). We report both mean and median Dice scores for the whole tumor, tumor core, and enhancing tumor.

Test Set Results						
Statistic	Median			Mean		
Tumor Compartment	WT	TC	ET	WT	TC	ET
nnU-Net [9]	0.929	0.919	0.857	0.906	0.827	0.745
GSpool-30	0.892	0.841	0.783	0.841	0.737	0.672

4.5 Explaining GNN predictions using SHAP values

Finally, we compute the SHAP values for a subset of representative patient samples. We stratify the computed SHAP values by modality, label, and whether the corresponding feature value was high intensity (bright) or low intensity (dark) (Fig. 3). Bright intensities are defined as the top 15% of intensity values within a given modality, while dark intensities are those in the bottom 15%.

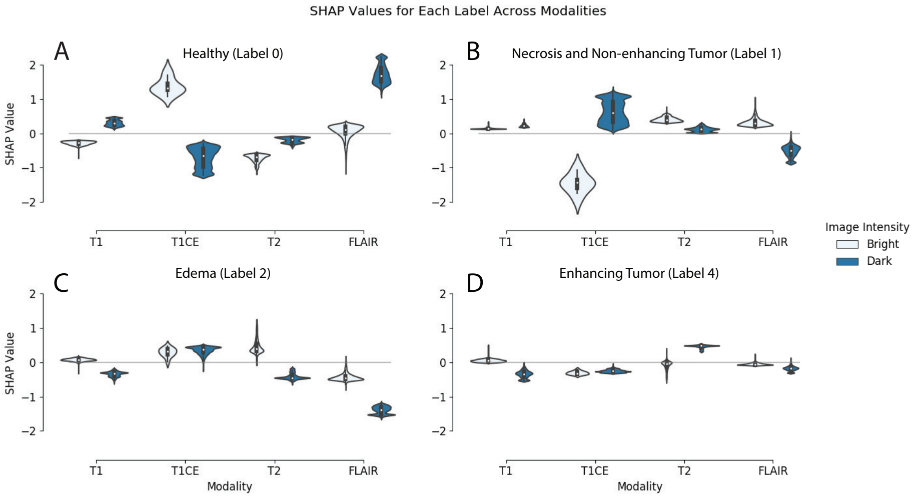

Fig. 3: SHAP values distribution grouped by label and stratified by modality. Dark Violin plots correspond to dark image regions in a particular modality, while lighter plots correspond to bright regions in the corresponding modality. Positive SHAP values indicate that the modality contributes to the prediction of a particular label, while negative SHAP values indicate that a modality contributes negatively to predicting that label. Panels A-D represent the SHAP values computed for different tissue labels.

We identify several trends for each modality's contribution to different labels in Fig. 3. Bright T1CE regions and dark FLAIR regions drive the prediction of healthy tissue (Fig. 3A), while the inverse is predictive of the necrotic and non-enhancing tumor core (Fig. 3B). Edematous tissue is defined by bright T2 regions and the lack of dark FLAIR regions (Fig. 3C). Lastly, in the tissue predicted to be enhancing tumor, dark T2 and dark T1 regions are assigned the highest and lowest SHAP values, respectively (Fig. 3D). For the enhancing tumor, we also observe that the absolute magnitudes of the SHAP values are substantially lower than those of the other 3 possible classifications. This observation indicates that predicting a node as an enhancing tumor is driven by a "process of elimination", not by intrinsic characteristics of the enhancing tumor. Rather than learning which features uniquely identify the enhancing tumor, the model instead relies on recognizing feature combinations that make the other label unlikely.

Overall, the T1CE and FLAIR modalities are consistently assigned the most variable SHAP values, while the T1 modality remains relatively constant. The relative utility of each modality is consistent with that determined by the BraTS organizers, who also state that the T1CE and FLAIR modalities are the most useful for manual segmentation [4].

Many of our findings for individual tumor regions also conform to radiation oncology practices for manual segmentation of brain tumors. For example, both non-enhancing tumor and necrosis are typically delineated by dark T1-CE, bright T2, and bright FLAIR regions of the MRI. Our model's SHAP value analysis recovers all three of these trends for the combined NET/necrosis regions. Interestingly, however, it indicates that T1CE and FLAIR have a much more pronounced effect on the prediction of these regions than T2 does. (Fig. 3B). Vasogenic edema (Label 2) may be visually assessed by contrasting bright T2 and FLAIR regions with moderate intensity T1CE and T1. However, it is often difficult to distinguish from other tumorous labels (1 and 4), since these can all appear bright on the T2 and FLAIR images, depending on tumor grade. Our analysis shows that the model correctly recognizes the brightness trend in the T2 and FLAIR modalities, but learns a more nuanced classification scheme to circumvent this issue. Rather than using bright FLAIR intensities as a marker for edema, it instead learns that a brain region that *lacks* dark FLAIR intensities are unlikely to be healthy, and then relies on the other modalities to distinguish further between the tumor subregions. Lastly, enhancing tumor is traditionally defined as bright (enhanced) regions in the T1CE modality. Surprisingly, bright T1CE regions are not assigned high SHAP values for the enhancing tumor, indicating that they play little to no role in the model's predictions thereof (Fig. 3D). When coupled with the relative scarcity of enhancing tumor labels, this observation could explain the inferior performance of the model in predicting the enhancing tumor (Label 4).

The above analysis indicates agreement between the feature combinations used by the model and clinical practice. Furthermore, the analysis provides insight into how the model distinguishes between regions that are known to be difficult to differentiate on MRI. Insight into why the results might not be optimal for enhancing tumor will allow us to address this issue. Such interpretability analysis is key to ensuring the adoption of deep learning models in healthcare [30].

5 Discussion

The development of effective automatic segmentation techniques can improve timely treatment for thousands of brain tumor patients annually. Furthermore, integrating automatic segmentation into routine clinical workflows could save physicians thousands of hours of painstaking manual annotation and standardize segmentations otherwise subject to inter-annotator variation. In this work, we have presented the application of Graph Neural Networks to brain tumor segmentation from MRIs. With this work, we provide several important contributions to the field. Firstly, we compare several common GNN variants and determine that GraphSAGE with the pooling aggregator performs the best. Secondly, we show that, compared to CNNs designed for the same task, GNN is less resource expensive and time-consuming to train. Lastly, we provide an interpretation of our model's predictions using Shapley value-based contribution scores.

A logical extension to this work is to combine the graph construction (involving supervoxel generation) and graph prediction in an end-to-end model, similarly to [11]. While the use of supervoxels to represent the images improves computational efficiency, our current model performance is heavily gated by the discrepancy between the SLIC output and the true segmentation labels. The treatment of supervoxels which contain voxels with different labels is poorly defined and consequently results in misclassified voxels. Even a model that classifies every graph node correctly achieves a voxel-wise Dice Whole Tumor score of only about 0.93 (Supplementary Fig. S2). A task-specific, end-to-end approach has the potential to alleviate this concern and increase performance substantially. End-to-end training would allow graph nodes to be delineated in greater accordance with the underlying tumor subregions, limiting the number of supervoxels spanning multiple labels. Furthermore, it would allow the model to learn node descriptors, which would likely be more informative than hand-engineered summary statistics for each modality. Another direction for future improvement is training the model hierarchically, that is, first determining the outline of the tumorous region(s) as a whole, and then segmenting each tumor subtype within the tumorous region. Most brain tumor segmentation models are effective at outlining the gross tumor, but struggle to delineate tumor compartments [4]. Such a training scheme should allow for a more nuanced capacity to distinguish the regions.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence **34**(11), 2274–2282 (2012)
2. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: Advances in Neural Information Processing Systems. pp. 9505–9515 (2018)
3. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., Freymann, J.B., Farahani, K., Davatzikos, C.: Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. Scientific data **4**, 170117 (2017)
4. Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R.T., Berger, C., Ha, S.M., Rozycki, M., et al.: Identifying the best machine learning algorithms for brain

tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. arXiv preprint arXiv:1811.02629 (2018)

5. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. pp. 3844–3852 (2016)
6. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in neural information processing systems. pp. 1024–1034 (2017)
7. Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis **30**(2), 129–150 (2011)
8. Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M., Larochelle, H.: Brain tumor segmentation with deep neural networks. Medical image analysis **35**, 18–31 (2017)
9. Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., Maier-Hein, K.H.: Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge. In: International MICCAI Brainlesion Workshop. pp. 287–297. Springer (2017)
10. Islam, A., Reza, S.M.S., Iftekharuddin, K.M.: Multifractal Texture Estimation for Detection and Segmentation of Brain Tumors. IEEE Transactions on Biomedical Engineering **60**(11), 3204–3215 (2013). https://doi.org/10.1109/TBME.2013.2271383
11. Juarez, A.G.U., Selvan, R., Saghir, Z., de Bruijne, M.: A joint 3d unet-graph neural network-based method for airway segmentation from chest cts. In: International Workshop on Machine Learning in Medical Imaging. pp. 583–591. Springer (2019)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
13. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 4765–4774. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf
14. Ma, C., Luo, G., Wang, K.: Concatenated and Connected Random Forests With Multiscale Patch Driven Active Contour Model for Automated Brain Tumor Segmentation of MR Images. IEEE Transactions on Medical Imaging **37**(8), 1943–1954 (2018). https://doi.org/10.1109/TMI.2018.2805821
15. McKinley, R., Meier, R., Wiest, R.: Ensembles of densely-connected cnns with label-uncertainty for brain tumor segmentation. In: International MICCAI Brainlesion Workshop. pp. 456–465. Springer (2018)
16. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al.: The multimodal brain tumor image segmentation benchmark (brats). IEEE transactions on medical imaging **34**(10), 1993–2024 (2014)
17. Molnar, C.: Interpretable Machine Learning (2019), https://christophm.github.io/interpretable-ml-book/
18. Myronenko, A.: 3d mri brain tumor segmentation using autoencoder regularization. In: International MICCAI Brainlesion Workshop. pp. 311–320. Springer (2018)
19. Ostrom, Q.T., Cioffi, G., Gittleman, H., Patil, N., Waite, K., Kruchko, C., Barnholtz-Sloan, J.S.: CBTRUS Statistical Report: Primary Brain and Other Central Nervous System Tumors Diagnosed in the United States in 2012–2016. Neuro-Oncology **21** (2019)
20. Pei, L., Bakas, S., Vossough, A., Reza, S.M., Davatzikos, C., Iftekharuddin, K.M.: Longitudinal brain tumor segmentation prediction in MRI using feature and label fusion. Biomedical Signal Processing and Control **55**, 101648 (Jan 2020). https://doi.org/10.1016/j.bspc.2019.101648

21. Rezaei, M., Yang, H., Meinel, C.: voxel-gan: Adversarial framework for learning imbalanced brain tumor segmentation. In: International MICCAI Brainlesion Workshop. pp. 321–333. Springer (2018)
22. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**(1), 61–80 (2009)
23. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
24. Serrano-Rubio, J., Everson, R., Hutt, H.: Brain tumour segmentation method based on sparse feature vectors. In: International MICCAI Brainlesion Workshop. pp. 420–427. Springer (2018)
25. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. arXiv preprint arXiv:1704.02685 (2017)
26. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
27. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. arXiv preprint arXiv:1703.01365 (2017)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
29. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
30. Vellido, A.: The importance of interpretability and visualization in machine learning for applications in medicine and health care. Neural Computing and Applications (Jan 2019). https://doi.org/10.1007/s00521-019-04051-w
31. Wang, M., Yu, L., Zheng, D., Gan, Q., Gai, Y., Ye, Z., Li, M., Zhou, J., Huang, Q., Ma, C., et al.: Deep graph library: Towards efficient and scalable deep learning on graphs. arXiv preprint arXiv:1909.01315 (2019)
32. Yan, Z., Youyong, K., Jiasong, W., Coatrieux, G., Huazhong, S.: Brain tissue segmentation based on graph convolutional networks. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 1470–1474. IEEE (2019)
33. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. In: Advances in neural information processing systems. pp. 9244–9255 (2019)
34. Zhou, C., Chen, S., Ding, C., Tao, D.: Learning contextual and attentive information for brain tumor segmentation. In: International MICCAI Brainlesion Workshop. pp. 497–507. Springer (2018)
35. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434 (2018)

1 Supplementary Information

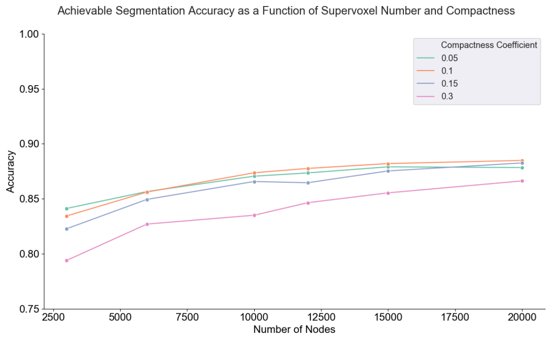

Fig. S1: The achievable segmentation accuracy as a function of supervoxel number and compactness. More supervoxels increase the achievable accuracy.

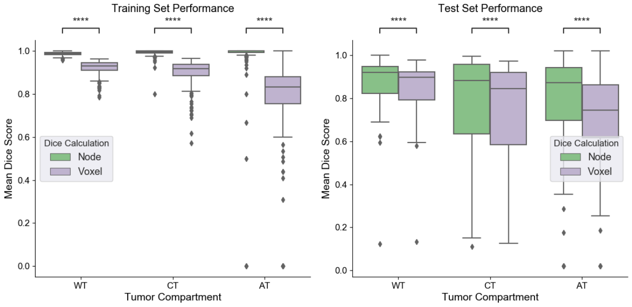

Fig. S2: Boxplot of Dice scores for the same brains computed by voxel vs. by supervoxel (node). Results shown for both test and train set. **** shows $p < 0.0001$ in paired t-test. Across every comparison, Dice scores calculated on voxels are significantly lower than when calculated by node. This effect is especially pronounced on the test set.

Table S1: Hausdorff Distances (95 percentile) calculated on test set for our model and nnUnet. Both median and mean scores are reported.

Test Set Results						
Statistic	Median			Mean		
Tumor Compartment	WT	TC	ET	WT	TC	ET
nnU-Net [9]	2.828	2.27	1.414	4.645	6.17	5.011
GSpool-30	4.359	5.10	3.317	7.60	10.30	5.45