

# TransBTS: Multimodal Brain Tumor Segmentation Using Transformer

Wenxuan Wang<sup>1</sup>, Chen Chen<sup>2</sup>, Meng Ding<sup>3</sup>, Hong Yu<sup>1</sup>, Sen Zha<sup>1</sup>, Jiangyun Li<sup>1,†</sup>

<sup>1</sup> School of Automation and Electrical Engineering, University of Science and Technology Beijing, China, [s20200579@xs.ustb.edu.cn](mailto:s20200579@xs.ustb.edu.cn), [g20198754@xs.ustb.edu.cn](mailto:g20198754@xs.ustb.edu.cn), [g20198675@xs.ustb.edu.cn](mailto:g20198675@xs.ustb.edu.cn), [leejy@ustb.edu.cn](mailto:leejy@ustb.edu.cn)  
<sup>2</sup> Center for Research in Computer Vision, University of Central Florida, USA, [chen.chen@ucf.edu](mailto:chen.chen@ucf.edu)

<sup>3</sup> Scoop Medical, Houston, TX, USA, [meng.ding@okstate.edu](mailto:meng.ding@okstate.edu)  
<sup>†</sup> Corresponding author: Jiangyun Li

**Abstract.** Transformer, which can benefit from global (long-range) information modeling using self-attention mechanisms, has been successful in natural language processing and 2D image classification recently. However, both local and global features are crucial for dense prediction tasks, especially for 3D medical image segmentation. In this paper, we for the first time exploit Transformer in 3D CNN for MRI Brain Tumor Segmentation and propose a novel network named TransBTS based on the encoder-decoder structure. To capture the local 3D context information, the encoder first utilizes 3D CNN to extract the volumetric spatial feature maps. Meanwhile, the feature maps are reformed elaborately for tokens that are fed into Transformer for global feature modeling. The decoder leverages the features embedded by Transformer and performs progressive upsampling to predict the detailed segmentation map. Extensive experimental results on both BraTS 2019 and 2020 datasets show that TransBTS achieves comparable or higher results than previous state-of-the-art 3D methods for brain tumor segmentation on 3D MRI scans. The source code is available at <https://github.com/Wenxuan-1119/TransBTS>.

**Keywords:** Segmentation · Brain Tumor · MRI · Transformer · 3D CNN

## 1 Introduction

Gliomas are the most common malignant brain tumors with different levels of aggressiveness. Automated and accurate segmentation of these malignancies on magnetic resonance imaging (MRI) is of vital importance for clinical diagnosis.

Convolutional Neural Networks (CNN) have achieved great success in various vision tasks such as classification, segmentation and object detection. Fully Convolutional Networks (FCN) [10] realize end-to-end semantic segmentation for the first time with impressive results. U-Net [15] uses a symmetric encoder-decoder structure with skip-connections to improve detail retention, becoming



the mainstream architecture for medical image segmentation. Many U-Net variants such as U-Net++ [24] and Res-UNet [23] further improve the performance for image segmentation. Although CNN-based methods have excellent representation ability, it is difficult to build an explicit **long-distance** dependence due to limited receptive fields of convolution kernels. This limitation of convolution operation raises challenges to learn global semantic information which is critical for dense prediction tasks like segmentation.

Inspired by the attention mechanism [1] in natural language processing, existing research overcomes this limitation by fusing the attention mechanism with CNN models. Non-local neural networks [21] design a plug-and-play non-local operator based on the self-attention mechanism, which can capture the long-distance dependence in the feature map but suffers from the high memory and computation cost. Schlemper et al. [16] propose an attention gate model, which can be integrated into standard CNN models with minimal computational overhead while increasing the model sensitivity and prediction accuracy. On the other hand, Transformer [19] is designed to model long-range dependencies in sequence-to-sequence tasks and capture the relations between arbitrary positions in the sequence. This architecture is proposed based *solely on self-attention*, dispensing with convolutions entirely. Unlike previous CNN-based methods, Transformer is not only powerful in modeling global context, but also can achieve excellent results on downstream tasks in the case of large-scale pre-training.

Recently, Transformer-based frameworks have also reached state-of-the-art performance on various computer vision tasks. Vision Transformer (ViT) [7] splits the image into patches and models the correlation between these patches as sequences with Transformer, achieving satisfactory results on image classification. DeiT [17] further introduces a knowledge distillation method for training Transformer. DETR [4] treats object detection as a set prediction task with the help of Transformer. TransUNet [5] is a concurrent work which employs ViT for medical image segmentation. We will elaborate the differences between our approach and TransUNet in Sec. 2.4.



**Research Motivation.** The success of Transformer has been witnessed mostly on image classification. For dense prediction tasks such as segmentation, both local and global (or long-range) information is important. However, as pointed out by [22], local structures are ignored when directly splitting images into patches as tokens for Transformer. Moreover, for medical volumetric data (e.g. 3D MRI scans) which is **beyond 2D**, local feature modeling among continuous slices (i.e. depth dimension) is also critical for volumetric segmentation. We are therefore inspired to ask: *How to design a neural network that can effectively model local and global features in spatial and depth dimensions of volumetric data by leveraging the highly expressive Transformer?*

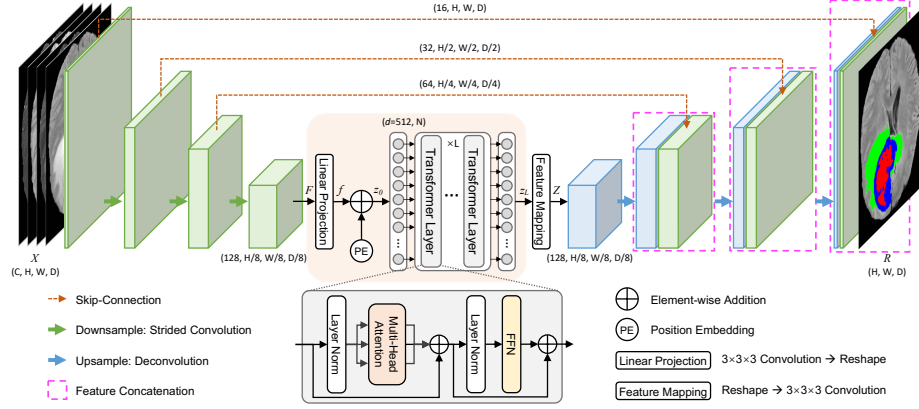
In this paper, we present the *first attempt* to exploit **Transformer** in 3D CNN for 3D MRI **Brain Tumor Segmentation** (TransBTS). The proposed TransBTS builds upon the encoder-decoder structure. The network encoder first utilizes 3D CNN to extract the volumetric spatial features and downsample the input 3D images at the same time, resulting in compact volumetric feature maps that

effectively captures the local 3D context information. Then each volume is reshaped into a vector (i.e. token) and fed into Transformer for global feature modeling. The 3D CNN decoder takes the feature embedding from Transformer and performs progressive upsampling to predict the full resolution segmentation map. Experiments on BraTS 2019 and 2020 datasets show that TransBTS achieves comparable or higher results than previous state-of-the-art 3D methods for brain tumor segmentation on 3D MRI scans. We also conduct comprehensive ablation study to shed light on architecture engineering of incorporating Transformer in 3D CNN to unleash the power of both architectures. We hope TransBTS can serve as a strong 3D baseline to facilitate future research on volumetric segmentation.

## 2 Method

### 2.1 Overall Architecture of TransBTS

An overview of the proposed TransBTS is presented in Fig. 1. Concretely, given an input MRI scan  $X \in \mathbb{R}^{C \times H \times W \times D}$  with a spatial resolution of  $H \times W$ , depth dimension of  $D$  (# of slices) and  $C$  channels (# of modalities), we first utilize 3D CNN to generate compact feature maps capturing spatial and depth information, and then leverage the Transformer encoder to model the long distance dependency in a global space. After that, we repeatedly stack the upsampling and convolutional layers to gradually produce a high-resolution segmentation result. The network details of TransBTS are provided in the **Appendix**. Next, we will describe the components of TransBTS in detail.



**Fig. 1.** Overall architecture of the proposed TransBTS.

## 2.2 Network Encoder

As the computational complexity of Transformer is quadratic with respect to the number of tokens (i.e. sequence length), directly flattening the input image to a sequence as the Transformer input is impractical. Therefore, ViT [7] splits an image into fixed-size ( $16 \times 16$ ) patches and then reshapes each patch into a token, reducing the sequence length to  $16^2$ . For 3D volumetric data, the straightforward tokenization, following ViT, would be splitting the data into 3D patches. However, this simple strategy makes Transformer unable to model the image *local context information across spatial and depth dimensions* for volumetric segmentation. To address this challenge, our solution is to stack the  $3 \times 3 \times 3$  convolution blocks with downsampling (strided convolution with stride=2) to gradually encode input images into low-resolution/high-level feature representation  $F \in \mathbb{R}^{K \times \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}}$  ( $K = 128$ ), which is  $1/8$  of input dimensions of  $H, W$  and  $D$  (overall stride (OS)=8). In this way, rich local 3D context features are effectively embedded in  $F$ . Then,  $F$  is fed into the Transformer encoder to further learn long-range correlations with a global receptive field.

**Feature Embedding of Transformer Encoder.** Given the feature map  $F$ , to ensure a comprehensive representation of each volume, a linear projection (a  $3 \times 3 \times 3$  convolutional layer) is used to increase the channel dimension from  $K = 128$  to  $d = 512$ . The Transformer layer expects a sequence as input. Therefore, we collapse the spatial and depth dimensions into one dimension, resulting in a  $d \times N$  ( $N = \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$ ) feature map  $f$ , which can be also regarded as  $N$   $d$ -dimensional tokens. To encode the location information which is vital in segmentation task, we introduce the learnable position embeddings and fuse them with the feature map  $f$  by direct addition, creating the feature embeddings as follows:

$$z_0 = f + PE = W \times F + PE \quad (1)$$

where  $W$  is the linear projection operation,  $PE \in \mathbb{R}^{d \times N}$  denotes the position embeddings, and  $z_0 \in \mathbb{R}^{d \times N}$  refers to the feature embeddings.

**Transformer Layers.** The Transformer encoder is composed of  $L$  Transformer layers, each of them has a standard architecture, which consists of a Multi-Head Attention (MHA) block and a Feed Forward Network (FFN). The output of the  $\ell$ -th ( $\ell \in [1, 2, \dots, L]$ ) Transformer layer can be calculated by:

$$z'_\ell = MHA(LN(z_{\ell-1})) + z_{\ell-1} \quad (2)$$

$$z_\ell = FFN(LN(z'_\ell)) + z'_\ell \quad (3)$$

where  $LN(*)$  denotes the layer normalization and  $z_\ell$  is the output of  $\ell$ -th Transformer layer.

## 2.3 Network Decoder

In order to generate the segmentation results in the original 3D image space ( $H \times W \times D$ ), we introduce a 3D CNN decoder to perform feature upsampling and pixel-level segmentation (see the right part of Fig. 1).

**Feature Mapping.** To fit the input dimension of 3D CNN decoder, we first design a feature mapping module to project the sequence data back to a standard 4D feature map. Specifically, the output sequence of Transformer  $z_L \in \mathbb{R}^{d \times N}$  is first reshaped to  $d \times \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}$ . In order to reduce the computational complexity of decoder, a convolution block is employed to reduce the channel dimension from  $d$  to  $K$ . Through these operations, the feature map  $Z \in \mathbb{R}^{K \times \frac{H}{8} \times \frac{W}{8} \times \frac{D}{8}}$ , which has the same dimension as  $F$  in the feature encoding part, is obtained.

**Progressive Feature Upsampling.** After the feature mapping, cascaded up-sampling operations and convolution blocks are applied to  $Z$  to gradually recover a full resolution segmentation result  $R \in \mathbb{R}^{H \times W \times D}$ . Moreover, skip-connections are employed to fuse the encoder features with the decoder counterparts by concatenation for finer segmentation masks with richer spatial details.

## 2.4 Discussion

A very recent work TransUNet [5] also employs Transformer for medical image segmentation. Here we want to highlight a few key distinctions between our TransBTS and TransUNet. (1) TransUNet is a 2D network that processes each 3D medical image in a **slice-by-slice** manner. However, our TransBTS is based on 3D CNN and processes all the image slices at a time, allowing the exploitation of better representations of continuous information between slices. In other words, TransUNet only focuses on the spatial correlation between tokenized image patches, but our method can model the long-range dependencies in both slice/depth dimension and spatial dimension simultaneously for volumetric segmentation. (2) As TransUNet adopts the ViT structure, it relies on pre-trained ViT models on large-scale image datasets. In contrast, our TransBTS has a flexible network design and is trained from scratch on task-specific dataset without the dependence on pre-trained weights.

## 3 Experiments

**Data and Evaluation Metric.** The first 3D MRI dataset used in the experiments is provided by the Brain Tumor Segmentation (BraTS) 2019 challenge [11,2,3]. It contains 335 cases of patients for training and 125 cases for validation. Each sample is composed of four modalities of brain MRI scans, namely native T1-weighted (T1), post-contrast T1-weighted (T1ce), T2-weighted (T2) and Fluid Attenuated Inversion Recovery (FLAIR). Each modality has a volume of  $240 \times 240 \times 155$  which has been aligned into the same space. The labels contain 4 classes: background (label 0), necrotic and non-enhancing tumor (label 1), peritumoral edema (label 2) and GD-enhancing tumor (label 4). The segmentation accuracy is measured by the Dice score and the Hausdorff distance (95%) metrics for enhancing tumor region (ET, label 1), regions of the tumor core (TC, labels 1 and 4), and the whole tumor region (WT, labels 1,2 and 4). The second 3D MRI dataset is provided by the Brain Tumor Segmentation Challenge (BraTS) 2020 [11,2,3]. It consists of 369 cases for training, 125 cases for validation and

166 cases for testing. Except for the number of samples in the dataset, the other information about these two datasets are the same.

**Implementation Details.** The proposed TransBTS is implemented in Pytorch and trained with 8 NVIDIA Titan RTX GPUs (each has 24GB memory) for 8000 epochs from scratch using a batch size of 16. We adopt the Adam optimizer to train the model. The initial learning rate is set to 0.0004 with a poly learning rate strategy, in which the initial rate decays by each iteration with power 0.9.

The following data augmentation techniques are applied: (1) random cropping the data from  $240 \times 240 \times 155$  to  $128 \times 128 \times 128$  voxels; (2) random mirror flipping across the axial, coronal and sagittal planes by a probability of 0.5; (3) random intensity shift between  $[-0.1, 0.1]$  and scale between  $[0.9, 1.1]$ . The softmax Dice loss is employed to train the network and  $L2$  Norm is also applied for model regularization with a weight decay rate of  $10^{-5}$ . In the testing phase, we utilize Test Time Augmentation (TTA) to further improve the performance of our proposed TransBTS.

**Table 1.** Comparison on BraTS 2019 validation set.

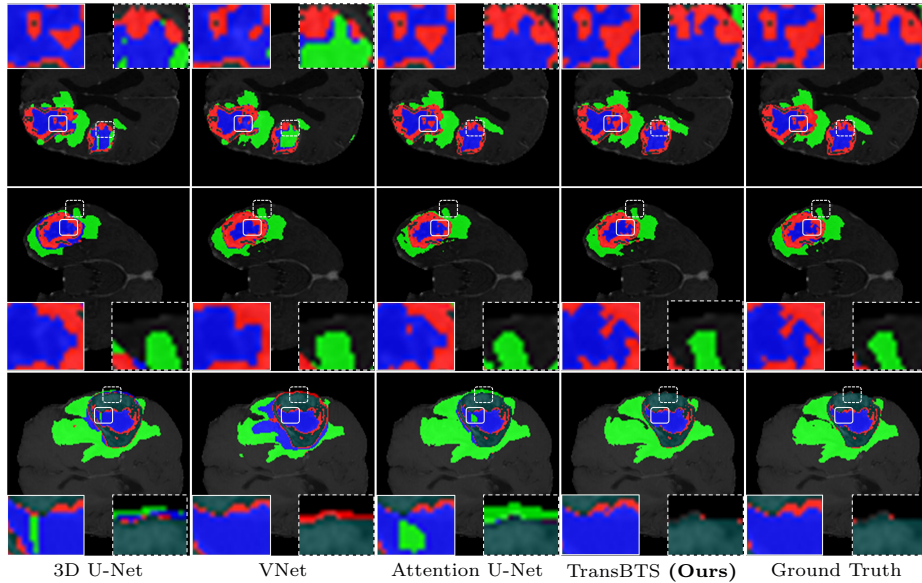
Method	Dice Score (%) $\uparrow$			Hausdorff Dist. (mm) $\downarrow$		
	ET	WT	TC	ET	WT	TC
3D U-Net [6]	70.86	87.38	72.48	5.062	9.432	8.719
V-Net [12]	73.89	88.73	76.56	6.131	6.256	8.705
KiU-Net [18]	73.21	87.60	73.92	6.323	8.942	9.893
Attention U-Net [14]	75.96	88.81	77.20	5.202	7.756	8.258
Wang et al. [20]	73.70	89.40	80.70	5.994	5.677	7.357
Li et al. [9]	77.10	88.60	81.30	6.033	6.232	7.409
Frey et al. [8]	78.7	89.6	80.0	6.005	8.171	8.241
Myronenko et al. [13]	<b>80.0</b>	89.4	<b>83.4</b>	3.921	5.89	6.562
<b>TransBTS w/o TTA</b>	78.36	88.89	81.41	5.908	7.599	7.584
<b>TransBTS w/ TTA</b>	78.93	<b>90.00</b>	81.94	<b>3.736</b>	<b>5.644</b>	<b>6.049</b>

**Table 2.** Comparison on BraTS 2020 validation set.

Method	Dice Score (%) $\uparrow$			Hausdorff Dist. (mm) $\downarrow$		
	ET	WT	TC	ET	WT	TC
3D U-Net [6]	68.76	84.11	79.06	50.983	13.366	13.607
Basic V-Net [12]	61.79	84.63	75.26	47.702	20.407	12.175
Deeper V-Net [12]	68.97	86.11	77.90	43.518	14.499	16.153
Residual 3D U-Net	71.63	82.46	76.47	37.422	12.337	13.105
<b>TransBTS w/o TTA</b>	78.50	89.00	81.36	<b>16.716</b>	6.469	10.468
<b>TransBTS w/ TTA</b>	<b>78.73</b>	<b>90.09</b>	<b>81.73</b>	17.947	<b>4.964</b>	<b>9.769</b>

### 3.1 Main Results

**BraTS 2019.** We first conduct five-fold cross-validation evaluation on the training set – a conventional setting followed by many existing works. Our TransBTS achieves average Dice scores of 78.69%, 90.98%, 82.85% respectively for ET, WT and TC. We also conduct experiments on the BraTS 2019 **validation** set and



**Fig. 2.** The visual comparison of MRI brain tumor segmentation results.

compare TransBTS with state-of-the-art (SOTA) 3D approaches. The quantitative results are presented in Table 1. TransBTS achieves the Dice scores of 78.93%, 90.00%, 81.94% on ET, WT, TC, respectively, which are comparable or higher results than previous SOTA 3D methods presented in Table 1. In terms of Hausdorff distance metric, a considerable improvement has also been achieved for segmentation. Compared with 3D U-Net[6], TransBTS shows great superiority in both metrics with significant improvements. This clearly reveals the benefit of leveraging Transformer for modeling the global relationships. For **qualitative analysis**, we also show a visual comparison of the brain tumor segmentation results of various methods including 3D U-Net[6], V-Net[12], Attention U-Net[14] and our TransBTS in Fig. 2. Since the ground truth for the validation set is not available, we conduct five-fold cross-validation evaluation on the training set for all methods. It is evident from Fig. 2 that TransBTS can describe brain tumors more accurately and generate much better segmentation masks by modeling long-range dependencies between each volume.

**BraTS 2020.** We also evaluate TransBTS on BraTS 2020 validation set and the results are reported in Table 2. We directly adopt the hyperparameters on BraTS19 for model training, our TransBTS achieves Dice scores of 78.73%, 90.09%, 81.73% and HD of 17.947mm, 4.964mm, 9.769mm on ET, WT, TC. Compared with 3D U-Net[6], V-Net[12] and Residual 3D U-Net, our TransBTS shows great superiority in both metrics with significant improvements. This clearly reveals the benefit of leveraging Transformer for modeling the global relationships.



### 3.2 Model Complexity

TransBTS has 32.99M parameters and 333G FLOPs which is a moderate size model. Besides, by reducing the number of stacked Transformer layers from 4 to 1 and halving the hidden dimension of the FFN, we reach a lightweight TransBTS which only has 15.14M parameters and 208G FLOPs while achieving Dice scores of 78.94%, 90.36%, 81.76% and HD of 4.552mm, 6.004mm, 6.173mm on ET, WT, TC on BraTS2019 validation set. In other words, by reducing the layers in Transformer as a simple and straightforward way to reduce complexity (54.11% reduction in parameters and 37.54% reduction in FLOPs of our lightweight TransBTS), the performance only drops marginally. Compared with 3D U-Net[6] which has 16.21M parameters and 1670G FLOPs, our lightweight TransBTS shows great superiority in terms of model complexity. Note that efficient Transformer variants can be used in our framework to replace the vanilla Transformer to further reduce the memory and computation complexity while maintaining the accuracy. But this is beyond the scope of this work.

### 3.3 Ablation Study

We conduct extensive ablation experiments to verify the effectiveness of TransBTS and justify the rationale of its design choices based on five-fold cross-validation evaluations on the BraTS 2019 training set. (1) We investigate the impact of the sequence length ( $N$ ) of tokens for Transformer, which is controlled by the overall stride (OS) of 3D CNN in the network encoder. (2) We explore Transformer at various model scales (i.e. depth ( $L$ ) and embedding dimension ( $d$ )). (3) We also analyze the impact of different positions of skip-connections.

**Sequence length  $N$ .** Table 3 presents the ablation study of various sequence lengths for Transformer. The first row (OS=16) and the second row (OS=8) both reshape each volume of the feature map to a feature vector after downsampling. It is noticeable that increasing the length of tokens, by adjusting the OS from 16 to 8, leads to a significant improvement on performance. Specifically, 1.66% and 2.41% have been attained for the Dice score of ET and WT respectively. Due to the memory constraint, after setting the OS to 4, we can not directly reshape each volume to a feature vector. So we make a slight modification to keep the sequence length to 4096, which is unfolding each  $2 \times 2 \times 2$  patch into a feature vector before passing to the Transformer. We find that although the OS drops from 8 to 4, without the essential increase of sequence length, the performance does not improve or even gets worse.

**Transformer Scale.** Two hyper-parameters, the feature embedding dimension ( $d$ ) and the number of Transformer layers (depth  $L$ ), mainly determines the scale of Transformer. We conduct ablation study to verify the impact of Transformer scale on the segmentation performance. For efficiency, we only train each model configuration for 1000 epochs. As shown in Table 4, the network with  $d = 512$  and  $L = 4$  achieves the best scores of ET and WT. Increasing the embedding dimension ( $d$ ) may not necessarily lead to improved performance ( $L = 4$ ,  $d$ : 512



**Table 3.** Ablation study on sequence length ( $N$ ).

OS	Sequence length( $N$ )	Dice score(%)		
		ET	WT	TC
16	512	73.30	87.59	<b>81.36</b>
8	4096	<b>74.96</b>	<b>90.00</b>	79.96
4	4096	74.86	87.10	77.46

**Table 4.** Ablation study on Transformer.

Depth ( $L$ )	Embedding dim ( $d$ )	Dice score(%)		
		ET	WT	TC
4	384	68.95	83.31	66.89
4	512	<b>73.72</b>	<b>88.02</b>	73.14
4	768	69.38	83.54	<b>74.16</b>
1	512	70.11	85.84	70.95
8	512	66.48	79.16	67.22

**Table 5.** Ablation study on the positions of skip-connections (SC).

Number of SC	Position of SC	Dice score(%)		
		ET	WT	TC
3	Transformer layer	74.96	90.00	79.96
3	3D Conv (Fig. 1)	<b>78.92</b>	<b>90.23</b>	<b>81.19</b>

vs. 768) yet brings extra computational cost. We also observe that  $L = 4$  is a “sweet spot” for the Transformer in terms of performance and complexity.

**Positions of Skip-connections (SC).** To improve the representation ability of the model, we further investigate the positions for skip-connections (orange dash lines “-→” in Fig. 1). The ablation results are listed in Table 5. If skip-connections are attached to the first three Transformer layers, it is more alike to feature aggregation from adjacent layers without the compensation for loss of spatial details. Following the traditional design of skip-connections from U-Net (i.e. attach to the 3D Conv layers as shown in Fig. 1), considerable gains (3.96% and 1.23%) have been achieved for the important ET and TC, thanks to the recovery of low-level spatial detail information.

## 4 Conclusion

We present a novel segmentation framework that effectively incorporates Transformer in 3D CNN for multimodal brain tumor segmentation in MRI. The resulting architecture, TransBTS, not only inherits the advantage of 3D CNN for modeling local context information, but also leverages Transformer on learning global semantic correlations. Experimental results on two datasets (BraTS 2019 and 2020) validate the effectiveness of the proposed TransBTS. In future work, we will explore computational and memory efficient attention mechanisms in Transformer to develop efficiency-focused models for volumetric segmentation.

## References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. 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)

3. 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)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
5. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)
6. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. pp. 424–432. Springer (2016)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Frey, M., Nau, M.: Memory efficient brain tumor segmentation using an autoencoder-regularized u-net. In: International MICCAI Brainlesion Workshop. pp. 388–396. Springer (2019)
9. Li, X., Luo, G., Wang, K.: Multi-step cascaded networks for brain tumor segmentation. In: International MICCAI Brainlesion Workshop. pp. 163–173. Springer (2019)
10. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
11. 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)
12. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). pp. 565–571. IEEE (2016)
13. Myronenko, A., Hatamizadeh, A.: Robust semantic segmentation of brain tumor regions from 3d mris. In: International MICCAI Brainlesion Workshop. pp. 82–89. Springer (2019)
14. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al.: Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999 (2018)
15. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
16. Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., Rueckert, D.: Attention gated networks: Learning to leverage salient regions in medical images. Medical image analysis **53**, 197–207 (2019)
17. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. arXiv preprint arXiv:2012.12877 (2020)

18. Valanarasu, J.M.J., Sindagi, V.A., Hacihaliloglu, I., Patel, V.M.: Kiu-net: Over-complete convolutional architectures for biomedical image and volumetric segmentation. arXiv preprint arXiv:2010.01663 (2020)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
20. Wang, F., Jiang, R., Zheng, L., Meng, C., Biswal, B.: 3d u-net based brain tumor segmentation and survival days prediction. In: International MICCAI Brainlesion Workshop. pp. 131–141. Springer (2019)
21. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
22. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. arXiv preprint arXiv:2101.11986 (2021)
23. Zhang, Z., Liu, Q., Wang, Y.: Road extraction by deep residual u-net. IEEE Geoscience and Remote Sensing Letters **15**(5), 749–753 (2018)
24. Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J.: Unet++: A nested u-net architecture for medical image segmentation. In: Deep learning in medical image analysis and multimodal learning for clinical decision support, pp. 3–11. Springer (2018)

## Appendix

In this Appendix, we provide the network details of TransBTS in Table 6.

**Table 6.** The design details of our proposed TransBTS network. Conv3 denotes a  $3 \times 3 \times 3$  convolutional layer; BN denotes Batch Normalization; DeConv denotes deconvolution layer. Each block of encoder and decoder is a residual block. Note that the size of input image is  $4 \times 128 \times 128 \times 128$ .

stage	block name	details	output size
3D CNN Encoder	InitConv	Conv3, Dropout	$16 \times 128 \times 128 \times 128$
	EnBlock1	$\begin{bmatrix} BN, ReLU, Conv3 \\ BN, ReLU, Conv3 \end{bmatrix} \times 1$	$16 \times 128 \times 128 \times 128$
	DownSample1	Conv3(stride2)	$32 \times 64 \times 64 \times 64$
	EnBlock2	$\begin{bmatrix} BN, ReLU, Conv3 \\ BN, ReLU, Conv3 \end{bmatrix} \times 2$	$32 \times 64 \times 64 \times 64$
	DownSample2	Conv3(stride2)	$64 \times 32 \times 32 \times 32$
	EnBlock3	$\begin{bmatrix} BN, ReLU, Conv3 \\ BN, ReLU, Conv3 \end{bmatrix} \times 2$	$64 \times 32 \times 32 \times 32$
	DownSample3	Conv3(stride2)	$128 \times 16 \times 16 \times 16$
	EnBlock4	$\begin{bmatrix} BN, ReLU, Conv3 \\ BN, ReLU, Conv3 \end{bmatrix} \times 2$	$128 \times 16 \times 16 \times 16$
Transformer Encoder	Linear Projection	Conv3, Reshape	$512 \times 4096 (d \times N)$
	Transformer	Transformer Layer $\times 4$	$512 \times 4096 (d \times N)$
3D CNN Decoder	Feature Mapping	Reshape, Conv3	$128 \times 16 \times 16 \times 16$
	DeBlock1	$\begin{bmatrix} BN, ReLU, Conv3 \\ BN, ReLU, Conv3 \end{bmatrix} \times 2$	$128 \times 16 \times 16 \times 16$
	UpSample1	Conv3, DeConv, Conv3	$64 \times 32 \times 32 \times 32$
	DeBlock2	$\begin{bmatrix} BN, ReLU, Conv3 \\ BN, ReLU, Conv3 \end{bmatrix} \times 1$	$64 \times 32 \times 32 \times 32$
	UpSample2	Conv3, DeConv, Conv3	$32 \times 64 \times 64 \times 64$
	DeBlock3	$\begin{bmatrix} BN, ReLU, Conv3 \\ BN, ReLU, Conv3 \end{bmatrix} \times 1$	$32 \times 64 \times 64 \times 64$
	UpSample3	Conv3, DeConv, Conv3	$16 \times 128 \times 128 \times 128$
	DeBlock4	$\begin{bmatrix} BN, ReLU, Conv3 \\ BN, ReLU, Conv3 \end{bmatrix} \times 1$	$16 \times 128 \times 128 \times 128$
	EndConv	Conv1, Softmax	$4 \times 128 \times 128 \times 128$