Uncertainty-aware Multi-dimensional Mutual Learning for Brain and Brain Tumor Segmentation

Junting Zhao, Zhaohu Xing, Zhihao Chen, Liang Wan, Tong Han, Huazhu Fu, and Lei Zhu

***Abstract*— Existing segmentation methods for brain MRI data usually leverage 3D CNNs on 3D volumes or employ 2D CNNs on 2D image slices. We discovered that while volume-based approaches well respect spatial relationships across slices, slice-based methods typically excel at capturing fine local features. Furthermore, there is a wealth of complementary information between their segmentation predictions. Inspired by this observation, we develop an Uncertainty-aware Multi-dimensional Mutual learning framework to learn different dimensional networks simultaneously, each of which provides useful soft labels as supervision to the others, thus effectively improving the generalization ability. Specifically, our framework builds upon a 2D-CNN, a 2.5D-CNN, and a 3D-CNN, while an uncertainty gating mechanism is leveraged to facilitate the selection of qualified soft labels, so as to ensure the reliability of shared information. The proposed method is a general framework and can be applied to varying backbones. The experimental results on three datasets demonstrate that our method can significantly enhance the performance of the backbone network by notable margins, achieving a Dice metric improvement of 2.8% on MeniSeg, 1.4% on IBSR, and 1.3% on BraTS2020.**

***Index Terms*— Brain and brain tumor segmentation, deep mutual learning, 2D/2.5D/3D network, uncertainty.**

I. Introduction

BRAIN disease is one of the most common major causes of the increase in world's mortality [1]. Brain and brain tumor segmentation together with subsequent quantitative assessments provide critical information in the study of neuropathology [2], essential for the planning of treatment strategies, monitoring of disease progression, and prediction of patient outcomes [3]–[5]. However, manual segmentation is tedious, time-consuming, and easily leads to human biases and mistakes [6]. Computer-Aided Detection (CAD) can assist radiologists in interpreting medical images with dedicated automatic algorithms. In the diagnosis of brain pathologies, magnetic resonance imaging (MRI) plays an important role by providing typical volumetric medical image data.

With the rapid development of convolutional neural networks (CNNs) during past decades, deep learning based methods [7], [8] have been developed for addressing brain tissue and brain tumor segmentation tasks. In general, segmentation methods can be divided into two main categories. The first is 2D network that segments 2D image slices and concatenates along a certain axis to obtain the 3D segmentation results [9], [10]. 2D-based methods can capture rich information within one image plane, but do not fully exploit the spatial correlation between slices. The second is 3D network that uses 3D convolutions to directly process volumetric information [11]. Note that 3D models may suffer from overfitting risks on a small image dataset due to their large number of parameters [12]. To alleviate this issue, some researchers introduced 2.5D networks to utilize limited spatial context information [13], [14]. For example, adjacent slices are packed to feed into a 3D network for the segmentation predication of the middle slices, which are further packed to form the final 3D segmentation results [14]. As 2D, 2.5D, and 3D networks (denoted as 2D-CNN, 2.5D-CNN, and 3D-CNN) train models on different spatial dimensions, we found that one of them may perform better than the others in different cases. As illustrated in Fig. 1, 2D-CNN, 2.5D-CNN, and 3D-CNN, with their backbones set as U-Net, produce the best results for case 1, case 2, and case 3, respectively. This indicates that different dimensional models can provide complementary information for each other.

To make full use of such complementary information, we propose an Uncertainty-aware Multi-dimensional Mutual learning framework (UMM for short). Unlike deep feature fusion methods such as H-DenseUNet [15], which concatenates features from different dimensional models, our framework utilizes their soft labels, avoiding feature conflicts and resulting in strong generalization capabilities. More specifically, the framework is built upon the three dimensional models, i.e., 2D-CNN, 2.5D-CNN, and 3D-CNN, and train them simultaneously under both hard supervision concerning the ground truth and weak supervision concerning soft labels from each other. We further leverage an uncertainty gating mechanism to select soft labels, in order to prevent one model from teaching other models with unreliable sharing information. Experimental results show that the proposed framework can

J. Zhao, Z. Xing, Z. Chen are with Tianjin University, P.R. China (e-mail:zhaojt@tju.edu.cn, 920232796@qq.com, zh_chen@tju.edu.cn).

Liang Wan (Corresponding author) is with College of Intelligence and Computing, Medical College, Tianjin University, P.R. China (e-mail: lwan@tju.edu.cn).

T. Han is with Tianjin Huanhu Hospital. (e-mail: mrbold@163.com)

H. Fu is with the Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore 138632. (e-mail: hzfu@ieee.org)

L. Zhu is with the ROAS Thrust, Hong Kong University of Science and Technology (Guangzhou), China, and Department of Electronic and Computer Engineering, Hong Kong SAR, China. (e-mail: leizhu@ust.hk)

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2023.3274255

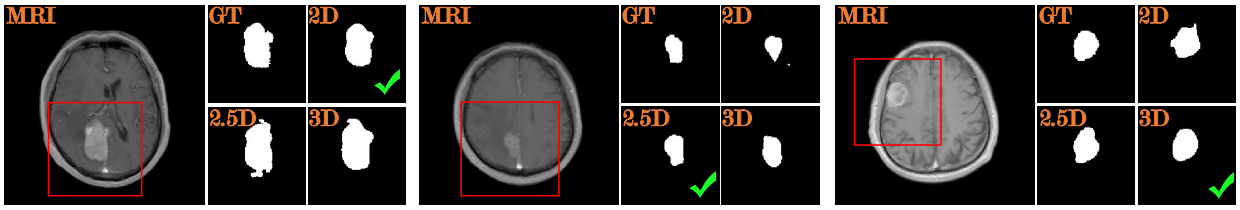


Fig. 1: Meningioma segmentation results for three segmentation networks, including 2D-CNN, 2.5D-CNN, and 3D-CNN, with their backbones set as UNet. **GT** presents ground truth.

clearly improve the model generalization ability, compared with using one single-dimensional model solely. In addition, the learned models can be used separately according to the specific needs of actual medical application scenarios. In summary, this work has the following contributions:

- We design an uncertainty-aware multi-dimensional mutual learning (UMM) framework with three segmentation models (2D-CNN, 2.5D-CNN, 3D-CNN) for brain and brain tumor segmentation. Each model provides additional soft supervision to the other two models.
- We leverage an uncertainty gating mechanism to filter out the uncertain regions in mutual learning for improving the reliability of shared information.
- Taking U-Net as the backbone, our framework outperforms the state-of-the-art (SOTA) segmentation methods, on the public IBSR brain tissue segmentation dataset, the public brain tumor segmentation BraTS2020 dataset and a private 3D meningioma brain tumor segmentation dataset that we collected from Tianjin Huanhu Hospital.

Our code and the trained models will be made publicly available upon the publication of this work.

II. Related Work

A. Brain and Brain Tumor Segmentation Methods

In past decades, hand-crafted feature based methods for brain and brain tumor segmentation have been proposed, including threshold-based methods [16], region-based methods [17], model-based methods [18], and clustering-based methods [19]. However, the hand-crafted features are not always satisfactory, thereby may degrade the segmentation performance [20].

Later, many methods based on convolutional neural networks (CNNs) are developed. The widely-used approaches are 2D-CNNs [7], [9] and 3D-CNNs [8], [11]. For instance, Ronneberger et al. [7] introduced skip connection to a standard encoder-decoder structure, and this U-shape structure has been widely utilized for addressing 2D medical image segmentation. Çiçek et al. [11] extended the 2D U-like structure to handling the 3D volume segmentation. Afterwards, a series of UNet-based models are proposed for brain tumor segmentation [9], [10]. They are mainly focused on increasing paths of dense connections or adding extra attention modules on skip connections. Apart from that, transformers, which were designed for sequence-to-sequence predictions, attract much attention for brain tumor segmentation due to their capabilities of explicitly modeling long-range relations [21], [22]. In work [21], the authors first extracted volumetric spatial feature maps and fed maps into transformers for global feature modeling, then used a decoder to predict the detailed segmentation map. In general, 2D-CNNs have small network size and low computational cost, but usually ignore inter-slice information of input 3D MRI data. On the other hand, 3D-CNNs consider inter-slice information, but may suffer from over-fitting risk on small 3D training datasets due to their large amount of parameters.

Considering the advantages of 2D-CNNs and 3D-CNNs, some researchers tried to integrate them. Li et al. [15] proposed a hybrid densely connected UNet (H-DenseUNet) for the liver and liver tumor segmentation. It first performs a 2D-based dense-UNet segmentation, and utilizes a 3D-based CNN to correct the spatial continuity of regions of interest. Zhou et al. [23] extracted 3D and 2D features on consecutive slices, then used a squeeze-and-excitation block to fuse the two kinds of features. What's more, some researchers introduced 2.5D networks to utilize limited spatial context information. For instance, Zhu et al. [24] designed a 2.5D recursive network, using several continuous 2D slices to explore the inter-slice context information. Cui et al. [25] introduced a slice radius to convolve adjacent slice information.

In this paper, we develop a mutual learning framework to investigate the potential of employing 2D, 2.5D, and 3D CNNs collaboratively, via supervising each other's learning process. During the testing phase, multi-dimensional models can be utilized individually or concurrently through a uncertainty-based fusion scheme.

B. Deep Mutual Learning

In [26], Zhang et al. developed a deep mutual learning (DML) model for the classification task. Unlike the one-way transfer between a teacher model and a student model in the knowledge distillation, an ensemble of students in DML learn collaboratively and teach each other by providing their predictions as soft labels. DML is also introduced to alleviate the cross-domain learning gap [27], [28]. Maximilian et al. [27] explored a multi-modality DML by controlling the information exchange between 2D images and 3D point clouds for 3D semantic segmentation. He et al. [28] exploited a graph pyramid DML to address the cross-dataset human parsing problem. They defined two levels of coarse-granularity categories to maintain two independent DML branches, which can prevent the learning process from introducing more noise that exists across datasets.

Here, we exploit the DML to alleviate the gap between multi-dimensional models (i.e., 2D-CNN, 2.5D-CNN, and 3D-CNN). Without increasing the amount of annotation data, we aim to allow multi-dimensional models to provide richer

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2023.3274255

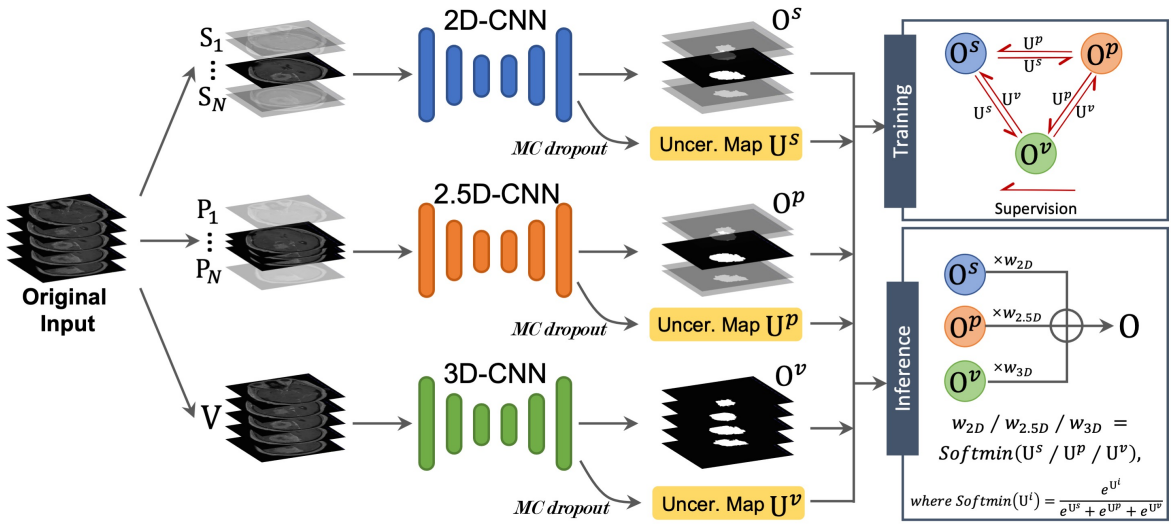


Fig. 2: The schematic illustration of the proposed framework. See Sec.III for details.

soft labels by respecting the prediction reliability, thereby improving the final volumetric segmentation performance.

III. Our Method

A. Network Architecture

As shown in Fig. 2 Our UMM framework contains three dimensional segmentation models (i.e., a slice-based 2D-CNN, a volumetric-based 3D-CNN, and a 2.5D-CNN). To leverage the complementary knowledge of the three models, we incorporate the uncertainty information into the mutual learning framework. The uncertainty information can help to penalize the missed information between slices for the 2D model and suppress the ambiguity of the information in the 3D model when the slice layer is thick (see Section III-B). In our framework, we use the same segmentation backbone to build up the three segmentation models. Here, the widely used segmentation model U-Net [7] is taken as the underlying backbone. Let $\mathbf{V} \in \mathcal{R}^{W\times H\times N}$ denote the input 3D MRI volume, where $[W, H, N]$ are $\mathbf{V}$'s three dimensions.

(1) 2D-CNN. We transform the input $\mathbf{V}$ into a set of 2D image slices $\{\mathbf{S}_k\}_{k=1}^{N} \in \mathcal{R}^{W\times H}$ along the transverse axis, and pass each slice to the 2D-CNN, then stack the outputs to obtain the 3D segmentation result, denoted as $\mathbf{O}^s \in \mathcal{R}^{W\times H\times N}$. By doing so, the 2D-CNN can extract the intra-slice information of the input volume for segmentation prediction.

(2) 3D-CNN. We take the 3D volume $\mathbf{V}$ as the input without any slicing operation, and the predicted 3D segmentation result is denoted as $\mathbf{O}^v \in \mathcal{R}^{W\times H\times N}$. Since the whole 3D volume has richer transverse axial information, the 3D-CNN can extract inter-slice information along the transverse axis.

(3) 2.5D-CNN. As Cui et al. [25] stacked adjacent slices in their 2.5D network, we pack every three adjacent slices along the transverse axis as a group, and take it as the input of 2.5D-CNN. For example, the k-th group $\mathbf{P}_k \in \mathcal{R}^{W\times H}$ consists of $(k-1)$-th slice, k-th slice, and $(k+1)$-th slice. Then it is fed into the 2.5D-CNN to predict the segmentation result of the k-th slice. The segmentation results from each group are packed to yield the 3D segmentation result $\mathbf{O}^p \in \mathcal{R}^{W\times H\times N}$.

Given the segmentation ground truth $\mathbf{G}$, we can compute the supervised losses of 2D-CNN, 3D-CNN and 2.5D-CNN, denoted as $\mathcal{L}_s^s$, $\mathcal{L}_s^v$, and $\mathcal{L}_s^p$ as follows:

$$\begin{aligned}\mathcal{L}_s^s &= \Phi_{CE}(\mathbf{O}^s, \mathbf{G}),\\ \mathcal{L}_s^v &= \Phi_{CE}(\mathbf{O}^v, \mathbf{G}),\\ \mathcal{L}_s^p &= \Phi_{CE}(\mathbf{O}^p, \mathbf{G}),\end{aligned} \tag{1}$$

where $\Phi_{CE}(\cdot)$ represents the weighted cross-entropy loss [29].

B. Uncertainty-aware Mutual Learning

To make 2D-CNN, 3D-CNN and 2.5D-CNN learn from each other, we resort to the recent mutual learning framework [26], and design three consistency losses (denoted as $\mathcal{D}_c^s$, $\mathcal{D}_c^v$, and $\mathcal{D}_c^p$) among three predications $\mathbf{O}^s$, $\mathbf{O}^p$ and $\mathbf{O}^v$:

$$\begin{aligned}\mathcal{D}_c^s &= \Phi_{MSE}(\mathbf{O}^v, \mathbf{O}^s) + \Phi_{MSE}(\mathbf{O}^p, \mathbf{O}^s),\\ \mathcal{D}_c^v &= \Phi_{MSE}(\mathbf{O}^s, \mathbf{O}^v) + \Phi_{MSE}(\mathbf{O}^p, \mathbf{O}^v),\\ \mathcal{D}_c^p &= \Phi_{MSE}(\mathbf{O}^s, \mathbf{O}^p) + \Phi_{MSE}(\mathbf{O}^v, \mathbf{O}^p),\end{aligned} \tag{2}$$

where $\Phi_{MSE}(\cdot)$ denotes the mean square error (MSE) loss. Compared to the supervised loss that offers hard labels in Eq. (1), the consistency loss provides soft labels for each voxel to learn important data distribution knowledge for enhancing the segmentation performance. However, apart from exchanging useful information, the original mutual learning may introduce noises inevitably. For instance, if the 2D-CNN has wrong predictions in some regions, the mutual learning will pass these errors to 3D-CNN and 2.5D-CNN, which eventually leads to unstable training.

To alleviate this issue, we leverage more reliable segmentation regions of training samples [30], [31] to assist mutual learning. In this work, we estimate three uncertainty maps (denoted as $\mathbf{U}^s$, $\mathbf{U}^v$, $\mathbf{U}^p$) for 2D-CNN, 3D-CNN and, 2.5D-CNN by using the Monte Carlo Dropout [32] (see MC dropout of Fig. 2), respectively. Specifically, we perform T stochastic forward passes under a random dropout, and empirically set T = 8 [33]. Then a voxel has a large uncertainty score if the model tends to generate a wrong segmentation prediction. In

detail, the definitions of $\mathbf{U}^s$, $\mathbf{U}^v$ and $\mathbf{U}^p$ are given by:

$$\begin{aligned}
\mathbf{U}^s &= -\sum_c \mathbf{A}^s \log \mathbf{A}^s \ , \ \text{where } \mathbf{A}^s = \frac{1}{\mathrm{T}} \sum_{t=1}^{T} \mathbf{O}_t^s, \\
\mathbf{U}^v &= -\sum_c \mathbf{A}^v \log \mathbf{A}^v \ , \ \text{where } \mathbf{A}^v = \frac{1}{\mathrm{T}} \sum_{t=1}^{T} \mathbf{O}_t^v, \\
\mathbf{U}^p &= -\sum_c \mathbf{A}^p \log \mathbf{A}^p \ , \ \text{where } \mathbf{A}^p = \frac{1}{\mathrm{T}} \sum_{t=1}^{T} \mathbf{O}_t^p,
\end{aligned} \tag{3}$$

where $\mathbf{A}^s$, $\mathbf{A}^v$, $\mathbf{A}^p$ denote the average of 2D-CNN predictions $\{\mathbf{O}_t^s | \ 1 \leq t \leq \mathrm{T}\}$, 3D-CNN predictions $\{\mathbf{O}_t^v\}$ and 2.5D-CNN predictions $\{\mathbf{O}_t^p\}$; $\log(\cdot)$ is the logarithmic function; c represents the class number of the segmented regions. Then, we incorporate the uncertainty maps into the computation of the three consistency losses ($\mathcal{L}_c^s$, $\mathcal{L}_c^v$, $\mathcal{L}_c^p$), given by:

$$\begin{aligned}
\mathcal{L}_c^s &= \Phi_{UG}(\mathbf{O}_1^v, \mathbf{O}_1^s) + \Phi_{UG}(\mathbf{O}_1^p, \mathbf{O}_1^s), \\
\mathcal{L}_c^v &= \Phi_{UG}(\mathbf{O}_1^s, \mathbf{O}_1^v) + \Phi_{UG}(\mathbf{O}_1^p, \mathbf{O}_1^v), \\
\mathcal{L}_c^p &= \Phi_{UG}(\mathbf{O}_1^s, \mathbf{O}_1^p) + \Phi_{UG}(\mathbf{O}_1^v, \mathbf{O}_1^p),
\end{aligned} \tag{4}$$

where the uncertainty-guided loss Φ_{UG} is defined as:

$$\begin{aligned}
\Phi_{UG}(\mathbf{O}_1^i, \mathbf{O}_1^j) &= \frac{\sum_v \left(\mathbb{I}(\mathbf{U}_{(v)}^i < z^i) \cdot \| \mathbf{O}_{1(v)}^j - \mathbf{O}_{1(v)}^i \|^2 \right)}{\sum_v \mathbb{I}(\mathbf{U}_{(v)}^i < z^i)}, \\
\text{where } \mathbb{I}(\mathbf{U}_{(v)}^i < z^i) &= \begin{cases} 1, & \textit{if } \mathbf{U}_{(v)}^i < z^i \ , \\ 0, & \textit{otherwise} \ . \end{cases}
\end{aligned} \tag{5}$$

In Eq. (5), for each sub-network, the regions with low uncertainty are reserved and used as the supervision for the other networks [33]. $\mathbf{O}_1^i$ and $\mathbf{O}_1^j$ are from the set $\{\mathbf{O}_1^v, \mathbf{O}_1^s, \mathbf{O}_1^p\}$. $\mathbf{O}_{1(v)}^i$/$\mathbf{O}_{1(v)}^j$ is the segmentation prediction $\mathbf{O}_1^i$/$\mathbf{O}_1^j$ at the $v-$th voxel. $\mathbf{U}_{(v)}^i$ is the uncertainty score at the $v-$th voxel for predicting $\mathbf{O}_1^i$. z^i is the threshold scalar on the uncertainty map, and its value is from the set $\{z^s, z^v, z^p\}$. We compute z^s, z^v, and z^p as: $z^s = \frac{1}{3}\max(\mathbf{U}^s)$, $z^v = \frac{1}{3}\max(\mathbf{U}^v)$, and $z^p = \frac{1}{3}\max(\mathbf{U}^p)$, where $\max(\mathbf{U})$ means the maximal value in this uncertainty map. '$(\cdot)$' denotes an element-wise product.

Training loss of our network. Hence, the total loss $\mathcal{L}_{total}^s$ of the 2D-CNN, the total loss $\mathcal{L}_{total}^v$ of the 3D-CNN and the total loss $\mathcal{L}_{total}^p$ of the 2.5D-CNN are computed as:

$$\begin{aligned}
\mathcal{L}_{total}^s &= \mathcal{L}_s^s + w\mathcal{L}_c^s, \\
\mathcal{L}_{total}^v &= \mathcal{L}_s^v + w\mathcal{L}_c^v, \\
\mathcal{L}_{total}^p &= \mathcal{L}_s^p + w\mathcal{L}_c^p,
\end{aligned} \tag{6}$$

where the weight w balances the supervised loss and consistency loss. Since the models often have poor-quality predictions at the initial training process, we set w using an adaptive weighting scheme, $w = exp(-5(1 - \frac{m}{\Gamma})^2)$, where m is the training epoch, $\exp(\cdot)$ denotes exponential function, and Γ is the total number of epochs. Note that w is small at the beginning of the training process, and gradually increases against the training number. This results in an enhanced ability to generalize to the input data. The training stage uses three total losses of Eq. (6) to train the 2D-CNN, the 3D-CNN, and the 2.5D-CNN, simultaneously.

C. Inference of Our Network

As shown in Fig. 2, the 2D-CNN, 2.5D-CNN and 3D-CNN of our method can produce three segmentation results. To obtain the final output of UMM-Net, we consider ensembling the three outputs. However, directly stacking the results from all three models is unreliable and prone to introduce bad predictions. Instead, we leverage the uncertainty map to guide the fusion process. It is evident that branches with lower uncertainty should exert a stronger influence on the final outcomes, whereas branches with higher uncertainty should contribute less significantly. Hence, we calculate weights $\mathbf{W}^s$, $\mathbf{W}^p$, $\mathbf{W}^v$ for 2D/2.5D/3D-CNN to balance the predictions. The final fusion prediction $\mathcal{P}$ is computed by:

$$\begin{aligned}
\mathcal{P} &= \mathbf{O}^s \cdot \mathbf{W}^s + \mathbf{O}^v \cdot \mathbf{W}^v + \mathbf{O}^p \cdot \mathbf{W}^p \ , \\
\text{where } \mathbf{W}^i &= \mathrm{Softmin}(\mathbf{U}^i) = \frac{e^{\mathbf{U}^i}}{e^{\mathbf{U}^s} + e^{\mathbf{U}^p} + e^{\mathbf{U}^v}},
\end{aligned} \tag{7}$$

$\mathrm{Softmin}(\cdot)$ denotes the Softmin function, which is used to assign higher weights to branches with low uncertainty.

IV. Experimental Results

A. Datasets and Experiment Settings

Datasets. We evaluate the effectiveness of our framework on three datasets, including a self-gathered meningioma segmentation dataset (denoted as **MeniSeg**), a public brain segmentation repository (**IBSR**) dataset[1], and a public brain tumor segmentation (**BraTS2020**) dataset[2].

1) MeniSeg Dataset: Meningiomas are the most common primary intracranial tumors in adults, comprising 38.3% of central nervous system tumors. We collected the Meningioma dataset (MeniSeg) in cooperation with Tianjin Huanhu Hospital. The dataset contains 155 MRI T1Gd volumes from meningioma patients, who had undergone tumor resection between March 2016 and March 2021. The scans were performed using four 3.0T MRI scanners (Skyra, Trio, Avanto, and Prisma from Siemens). The scanning parameters were adjusted to generate similar MRI appearances. Three radiologists marked meningioma tumors. The resolution of MRI slices ranges from 256 $\times$ 204 to 320 $\times$ 320, and in-plane spacing varies from 0.71 mm $\times$ 0.71 mm to 0.89 mm $\times$ 0.89 mm. Regarding all the data, the number of slices takes a range of [17, 23] and the thickness is within [6.5mm, 7.8mm]. In our experiments, we resample the 3D volumes to 256 $\times$ 256 $\times$ 32. Moreover, two-fold cross-validation is adopted for MeniSeg dataset.

2) IBSR Dataset: Brain segmentation repository (IBSR) dataset has 18 MRI T1 scans, and all the volumes have the same spatial dimension (i.e., 256 $\times$ 256 $\times$ 128). The volume spacing varies from 0.83 mm $\times$ 0.83 mm $\times$ 1.5 mm to 1.0 mm $\times$ 1.0 mm $\times$ 1.5 mm. For each volume, three brain tissues, i.e. cerebrospinal fluid (CSF), gray matter (GM), and white matter (WM), are manually annotated by clinicians. In our experiments, all these volumes have been skull-stripped and resampled to a size of 256 $\times$ 256 $\times$ 128. Due to the small size of the IBSR dataset, we adopt six-fold cross-validation for IBSR in the evaluation.

[1]IBSR dataset link: http://www.cma.mgh.harvard.edu/ibsr/
[2]BraTS 2020 dataset link: https://www.med.upenn.edu/cbica/brats2020/

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2023.3274255

3) BraTS2020 Dataset: Brain tumor segmentation (BraTS) 2020 dataset contains 369 aligned four-modal MRI scans (i.e., T1, T1Gd, T2, and T2-FLAIR). Each volume are already resampled and co-registered with the same dimension (i.e., 240 $\times$ 240 $\times$ 155). All the images are annotated by experienced neuro-radiologists with GD-enhancing tumor (ET), peritumoral edema (ED), and the necrotic and non-enhancing tumor core (NCR/NET). The segmentation task aims to segment the different sub-regions containing the enhancing tumor (ET), the tumor core (TC), and the whole tumor (WT). We randomly divide the dataset into three subsets with a split ratio of 7:1:2 for training/validation/testing.

Implementation details. Our proposed framework is implemented with PyTorch. The network parameters are initialized with a random uniform distribution (Xavier initialization). We use an Adam optimizer with a weight decay of 10^{-5}, and a learning rate of 10^{-4}. For MeniSeg and IBSR dataset, we feed the whole volume into 3D network for training and inference. For BraTS2020 dataset, considering memory limit, we randomly sample the volume as patches of size 96 $\times$ 96 $\times$ 96; in the inference, the sliding window overlap rate is set as 0.5. Experiments on all the datasets adopt data augmentation with random flips, rotations, intensity scaling, and shifts. Our network requires 12.5 hours on an NVIDIA GTX 3090 GPU for the training on the MeniSeg, 5 hours for the IBSR dataset, and needs 8 hours on an NVIDIA A100 GPU for the BraTS2020 dataset.

Evaluation Metrics. We use four evaluation metrics. (1) Dice similarity coefficient [34] computes the region-based similarity between the predicted result X and the ground truth Y as Eq. (8), where $|\cdot|$ represents the operation of cardinality computation. (2) The 95% Hausdorff Distance (HD95) [35] measures the maximum 95% surface distances between X and Y as Eq. (9). (3) the Recall [36] measure how much the model correctly identifies True Positives as Eq. (10), where TP, FP, and FN denote the true positive, false positive and false negative counts, respectively. (4) We design an average precision at 70% Volume-based Recall (VR70) metric, inspired from AP50 [37], which is commonly used in object detection and instance segmentation tasks. VR70 measures the average precision of the volume predictions with a Recall of at least 70% between the predictions and the ground truth.

$$Dice(X,Y) = \frac{2\times \mid X \cap Y \mid}{\mid X \mid + \mid Y \mid} \tag{8}$$

$$HD95 = max_{k95\%}[d(X,Y), d[Y,X]] \tag{9}$$

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

To evaluate the statistical significance, we also perform paired-t tests with p-value < 0.05 on Dice, HD95, and Recall.

B. Comparison with SOTA Methods

We compare our method against several SOTA segmentation methods, including **(1)** 2D slice-based segmentation methods, i.e. U-Net [7], UNet++ [9], Attention U-Net [10] (Att. U-Net for short), Swin-Unet [22], **(2)** 3D volume-based segmentation methods, i.e. 3D U-Net [11], V-Net [34], nnU-Net [38], SegResNet [8], TransBTS [21], UNERT [6], Swin UNETR [35] (Swin UR for short), **(3)** a 2D-3D fusion method H-DenseUNet [15], (H-Den for short), and **(4)** a 2.5D U-Net. For a fair comparison, we utilize public implementations of comparative methods and re-trained them respectively. In our experiments, our UMM framework utilizes U-Net as the backbone, which is denoted as UMM-Net.

TABLE I: The comparison results on **MeniSeg**. † indicates our UMM-Net achieves statistically significant results (p-value $<$ 0.05) in comparison with SOTA. Red and blue values mean the best and the second best among these methods.

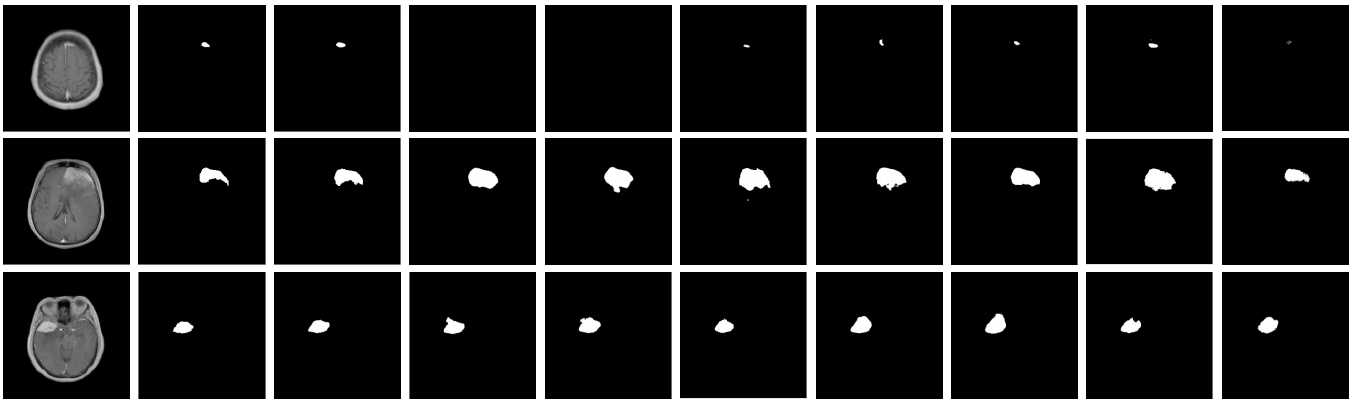
Method	Type	Dice ↑	HD95 ↓	Recall ↑	VR70 ↑
U-Net [7]	2D	0.712±0.003 †	9.717±0.195 †	0.718±0.014 †	0.664±0.007
UNet++ [9]	2D	0.713±0.008 †	6.302±3.296 †	0.760±0.037	0.710±0.030
Att. U-Net [10]	2D	0.697±0.010 †	6.874±1.200 †	0.681±0.020	0.652±0.033
Swin-Unet [22]	2D	0.752±0.008 †	4.339±1.040	0.801±0.081	0.755±0.034
3D U-Net [11]	3D	0.740±0.000 †	8.343±0.239 †	0.772±0.014	0.768±0.002
V-Net [34]	3D	0.648±0.032 †	4.548±0.752	0.731±0.037 †	0.658±0.006
nnU-Net [38]	3D	0.742±0.006 †	9.110±1.433 †	0.766±0.014 †	0.781±0.020
SegResNet [8]	3D	0.739±0.019 †	7.298±1.470 †	0.784±0.031 †	0.774±0.001
TransBTS [21]	3D	0.721±0.009 †	6.894±0.483 †	0.765±0.002	0.716±0.016
H-DenseUNet [15]	2D & 3D	0.696±0.000 †	9.851±0.553 †	0.704±0.001 †	0.710±0.043
2.5D UNet	2.5D	0.737±0.006 †	6.874±1.200 †	0.759±0.020 †	0.710±0.012
UMM-Net (Ours)	2D&2.5D&3D	0.761±0.009	3.068±0.914	0.790±0.006	0.813±0.029

1) MeniSeg Dataset:

Quantitative results on MeniSeg. As shown in Table I, we can find that 3D methods generally have better performance than 2D methods on MeniSeg. This is because that the meningioma regions have inhomogeneous appearances across slices, and 3D networks can better exploit the inter-slices spatial correlations. However, the 2D-based Swin-Unet achieves the best Recall score of 0.801±0.081, and works better than 3D segmentation methods, since it leverages the effective transformer blocks to capture long-range relations. Among the compared methods, our UMM-Net achieves the best Dice score of 0.761±0.009, the best HD95 score of 3.068±0.914, the second Recall score of 0.790±0.006, and the best VR70 of 0.813±0.029. Compared to its baseline in 2D (i.e., U-Net), it obtains significant improvements of 6.9% Dice, 68.4% HD95, and 10.0% Recall, as evidenced by all p-value$<$0.05. Compared to its baseline in 3D (i.e., 3D U-Net), it also has clear improvements upon these metrics of 2.8%, 63.2%, 2.3%, and 5.9%.

Note that H-DenseUNet, as a SOTA 2D-3D fusion method, does not achieve desired segmentation performance. This may be because H-DenseUNet has lots of training parameters, and tends to incur the over-fitted issue on our limited meningioma training samples. On the other hand, the three segmentation models in our method just exchange the label information for supervision rather than exchanging dense features, thereby largely reducing training parameters. Compared to H-DenseUNet, our UMM-Net obtains improvements of 9.3% Dice, 68.9% HD95, 12.2% Recall, and 14.5% VR70.

Visual comparisons on MeniSeg. Fig. 3 shows that our UMM-Net can more accurately segment meningioma regions than all competitors, and our results are more consistent with the ground truth (see Fig. 3(b)). More concretely, for small

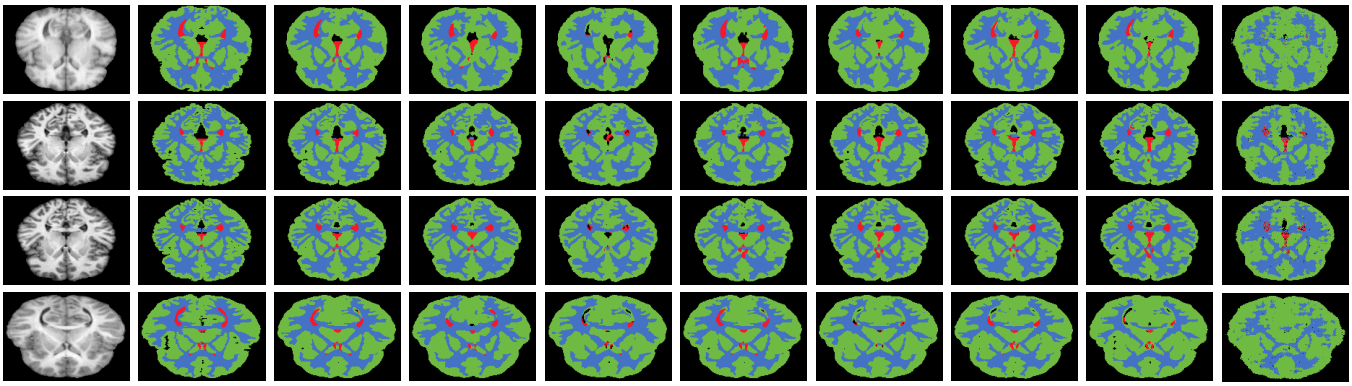


(a) Input (b) GT (c) **Ours** (d) U-Net (e) Att. U-Net (f) Swin-Unet (g) V-Net (h) SegResNet (i) TransBTS (j) H-Den.

Fig. 3: Visual comparisons on **MeniSeg**. Apparently, our UMM-Net achieves best visual results.

TABLE II: The comparison results on **IBSR**. † indicates our UMM-Net achieves statistically significant results (p-value < 0.05) in comparison with SOTA. Red and blue values mean the best and the second best among these methods.

		CSF			GM			WM			Average			
Method	Type	Dice ↑	HD95 ↓	Recall ↑	Dice ↑	HD95 ↓	Recall ↑	Dice ↑	HD95 ↓	Recall ↑	mDice ↑	mHD95 ↓	mRecall ↑	mVR70 ↑
U-Net [7]	2D	0.744±0.049	4.606±1.195	0.706±0.093	0.883±0.012	1.000±0.000	0.915±0.019	0.863±0.016	1.046±0.113	0.867±0.014	0.830±0.026 †	2.217±0.436 †	0.829±0.042 †	0.889±0.070
UNet++ [9]	2D	0.839±0.018	2.467±0.986	0.832±0.034	0.913±0.012	1.000±0.000	0.935±0.023	0.905±0.023	1.000±0.000	0.899±0.061	0.886±0.018 †	1.489±0.329 †	0.889±0.039 †	0.981±0.045
Att. U-Net [10]	2D	0.684±0.039	2.953±2.191	0.637±0.052	0.872±0.016	1.000±0.000	0.910±0.012	0.846±0.022	1.147±0.361	0.837±0.028	0.801±0.026 †	1.700±0.851 †	0.795±0.031 †	0.963±0.057
Swin-Unet [22]	2D	0.853±0.014	1.157±0.150	0.836±0.033	0.908±0.016	1.000±0.000	0.924±0.015	0.899±0.008	1.000±0.000	0.906±0.027	0.887±0.013 †	1.052±0.050	0.889±0.025 †	1.000±0.000
3D U-Net [11]	3D	0.835±0.008	2.102±1.195	0.837±0.037	0.914±0.014	1.000±0.000	0.941±0.014	0.907±0.011	1.000±0.000	0.902±0.034	0.885±0.011 †	1.367±0.398 †	0.893±0.028	1.000±0.000
V-Net [34]	3D	0.758±0.122	3.980±2.521	0.690±0.191	0.907±0.014	1.000±0.000	0.947±0.021	0.909±0.020	1.046±0.113	0.905±0.050	0.858±0.052 †	2.009±0.878 †	0.847±0.087 †	0.889± 0.141
nnU-Net [38]	3D	0.847±0.025	1.916±1.335	0.843±0.048	0.910±0.020	1.000±0.000	0.951±0.018	0.917±0.010	1.000±0.000	0.912±0.017	0.891±0.018 †	1.305±0.445	0.902±0.028	1.000±0.000
SegResNet [8]	3D	0.826±0.026	1.400±0.685	0.826±0.083	0.914±0.020	1.000±0.000	0.950±0.012	0.911±0.012	1.000±0.000	0.905±0.030	0.884±0.019 †	1.133±0.228	0.894±0.042 †	1.000±0.000
TransBTS [21]	3D	0.819±0.025	2.359±1.033	0.804±0.058	0.917±0.009	1.000±0.000	0.943±0.014	0.911±0.011	1.000±0.000	0.906±0.036	0.882±0.015 †	1.453±0.344 †	0.884±0.036 †	1.000±0.000
H-DenseUNet [15]	2D&3D	0.584±0.204	9.051±1.548	0.511±0.188	0.774±0.028	1.764±1.872	0.810±0.026	0.688±0.018	1.000±0.000	0.671±0.031	0.682±0.083 †	3.939±1.140 †	0.664±0.082	0.870± 0.271
2.5D UNet	2.5D	0.721±0.049	4.704±1.602	0.669±0.098	0.904±0.013	1.739±1.676	0.936±0.014	0.897±0.012	1.000±0.000	0.898±0.018	0.841±0.025 †	2.481±1.093 †	0.834±0.043 †	0.944± 0.062
UMM-Net (Ours)	2D&2.5D&3D	0.859±0.017	1.343±0.370	0.839±0.016	0.921±0.009	1.000±0.000	0.940±0.009	0.912±0.011	1.000±0.000	0.906±0.014	0.897±0.012	1.114±0.123	0.895±0.013	1.000±0.000



(a) Input (b) GT (c) **Ours** (d) U-Net (e) Att. U-Net (f) Swin-Unet (g) V-Net (h) SegResNet (i) TransBTS (j) H-Den.

Fig. 4: Visual comparisons on the CSF, GM, and WM segmentation results on **IBSR**. Apparently, UMM-Net achieves best visual results. **Blue** represents WM. **Geen** represents GM. **Red** represents CSF.

meningioma lesions like the 1st-case, U-Net, Attention U-Net, and H-DenseUNet have degraded performance. In the 2nd-case, our UMM-Net exhibits good ability in processing edges of the lesions, as well as for irregularly shaped lesions. For relatively regular shape meningioma lesions like the 3rd-case, our UMM-Net and Swin-Unet reach the best performance.

2) IBSR Dataset: **Quantitative results on IBSR.** Table II reports the quantitative segmentation performances on IBSR in terms of three brain regions, including CSF, GM, and WM. We also compute the metric mean on the three regions, denoted as mDice, mHD95, mRecall, and mVR70. We can find that 2D methods achieve comparable performances to 3D methods for IBSR. The main reason is that brain organs have a relatively fixed position, small morphological changes, and obvious object characteristics, and hence the spatial contexts across 2D image slices have a limited impact on the segmentation results. Swin-Unet has superior segmentation results on CSFs, which are small and non-fixed regions, achieves the second best Dice value on CSF of 0.853±0.014, and the best HD95 on all ROIs. On GM, TransBTS gets the second best Dice score of 0.917±0.009, and a perfect HD95 score of 1.000±0.000. nnU-Net has outstanding performance on WM while achieving

the best Dice and Recall scores. It indicates that TransBTS and nnU-Net can better handle large brain regions.

In comparison, our UMM-Net significantly outperforms most competitors on the average performance, getting the best mDice of 0.897±0.012, the second best mHD95 of 1.114±0.123, the second best mRecall of 0.895±0.013, and the best mVR70 of 1.000±0.000; also achieving the same ranking as the average metrics on the CSF. It is clear that our framework has better overall segmentation results on IBSR as well as on the small brain regions (i.e., CSF), which are further illustrated in the visual comparison (see Fig 4.) Compared to its baseline in 2D (i.e., U-Net), UMM-Net has a significant improvements upon mDice, mHD95, mRecall of 8.1%, 49.8%, 8.0%; and achieves 1.4%, 18.5%, 0.2% improvements compared to 3D U-Net.

Visual comparisons on IBSR. As shown in Fig. 4, regarding the large brain tissues (i.e., GM and WM), we can find that H-DenseUNet tends to introduce a lot of noise, while other compared methods can better segment them. More importantly, our method has the best segmentation performance on GM and WM. For the CSF, all the compared methods except Swin-Unet tend to produce wrong segmentation predictions, especially for narrow and slender CSFs (see the 1st-row and the 4th-row). On the contrary, our method can more accurately segment CSF.

3) BraTS2020 Dataset:

Quantitative results on BraTS2020. On BraTS2020, since we train the 3D model by randomly sampling the volume as patches (see Section IV-A for details), 3D methods can fully learn the information of adjacent pixels. Thus, as shown in Table III 3D methods have better results than 2D methods which learn image features slice by slice. Specifically, for ET, the smallest ROI, Swin UNETR achieves the second best Dice value of 0.749, and the best Recall value of 0.831, which suggests that Swin UNETR has better performance for tumor segmentation in small regions. For TC, with additional focus on the peritumoral edema and invaded tissue compared to ET, nnU-Net has satisfactory results, achieving the best Dice value of 0.847. For WT, the largest region among ROIs, SegResNet has competitive results, getting 0.917, 1.208, and 0.911 on Dice, HD95, and Recall, respectively. Moreover, SegResNet obtains a VR70 of 0.891, showing that it produces accurate segmentation results of WT for more than 89.1% volumes.

Compared to SOTA, our UMM-Net achieves the best on mDice, mHD95, and mVR70. For WT and ET, it obtains the best Dice (0.924 and 0.756, respectively) and HD95 (1.045 and 3.192, respectively). For TC, our method achieves the second best score on Dice, HD95, and Recall. It indicates that our method has better segmentation performance for tumor regions compared to the peritumoral edema and invaded tissue.

Visual comparisons on BraTS2020. As shown in Fig. 5, when the enhancing tumor area is small (e.g., the 1st-case), compared methods have missed detection (e.g., Swin-Unet), or misindentifed the edema around enhancing tumor as enhancing tumor (e.g, UNet++, Attention U-Net, SegResNet), or misindentifed the tumor core as multiple enhancing tumors. In comparison, our UMM-Net has better segmentation ability for small enhancing tumor. For the irregular WT such as the 2nd-case, our method and UNet++ have more consistent segmentation results with GT. The segmentation regions of other methods are too regular and coherent. For cases with relatively small WT area in the 3rd-case, SOTA methods segment WT into normal brain tissue, such as UNet++. SegResNet has respectable segmentation for WT and TC, however it predicts part of the ET as normal tissues.

C. Ablation Study

1) Module Performance Analysis: We conduct ablation studies on MeniSeg and IBSR in Fig. 6. We denote each segmentation network (i.e., 2D/2.5D/3D-CNN) as **Pure**. **ML** represents that we adopt the original mutual learning strategy of Eq. (2) to boost each sub-network. We denote **ML+U** as the mutual learning with an uncertainty-aware gating mechanism (see Eq. (5)). The final performance calculated by Eq. (7) is denoted as **Fusion**.

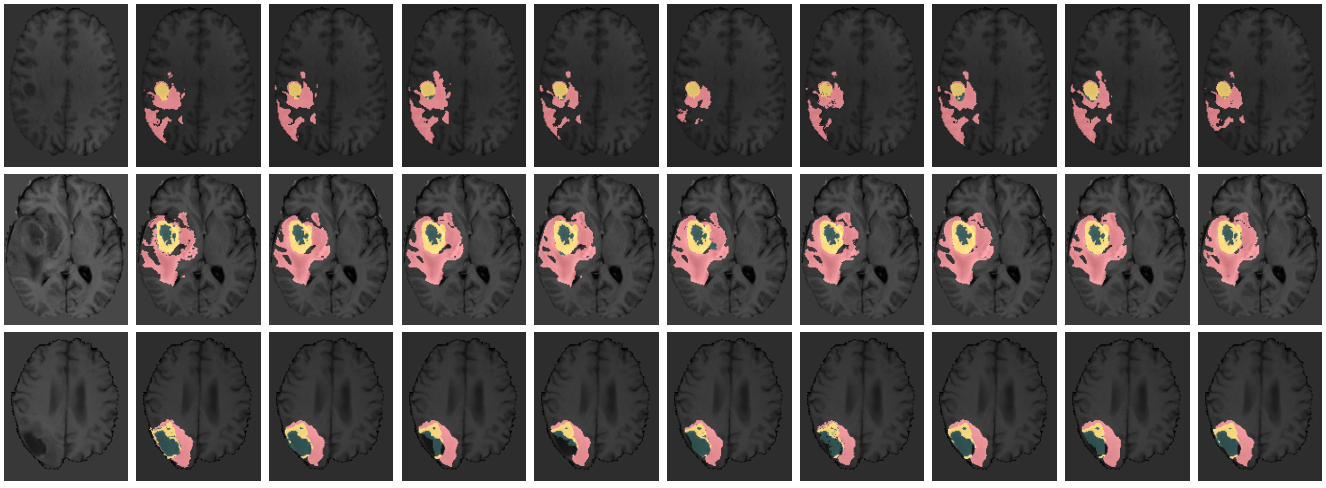
In Fig. 6(A), we can find that **ML** improves the segmentation performance of 2.5D-CNN and 3D-CNN, with an increase of 0.4% Dice for 2.5D-CNN and 0.5% Dice for 3D-CNN than **Pure**. This proves that mutual learning enables our method to benefit from weak supervision from the other two models. However, it is worth noting that the mutual learning is not optimistic about the improvement of 2D-CNN, and even decreases in some aspects. The Dice performance of 2D-CNN decreases from 0.712 to 0.703 and the Recall value decreases from 0.718 to 0.697. The reason behind this is that without the uncertainty guide, 2.5D-CNN and 3D-CNN would introduce an negative propagation of the 2D-CNN. When exploring the uncertainty-aware gating mechanism in the mutual learning framework on MeniSeg, 2D-CNN gets clear improvement compared to its **Pure** model. For Dice, it increases from 0.712 to 0.719. Both 2.5D-CNN and 3D-CNN further increase the segmentation performance. We can find that **ML+U** outperforms **ML** with a 2.3% Dice improvement for the 2D-CNN, 2.6% Recall improvement for the 2.5D-CNN, and with a 2.2% Dice improvement for the 3D-CNN.

Fig. 6(B) reports the analysis results on IBSR, further proving that all modules (i.e., **ML**, **ML+U**, **Fusion**) consistently improve the performance for most metrics and best performance is attained when equipping all proposed modules.

2) Combinations of 2D-CNN, 2.5D-CNN, and 3D-CNN: We conduct an ablation study experiment to demonstrate that the uncertainty-aware mutual learning can also help to improve the case of using two segmentation networks on the MeniSeg dataset. In Table IV, "2D&3D" means that we utilize 2D-CNN and 3D-CNN in our framework. Here, we utilize U-Net as basic network. Among two network combinations, "2D&3D" gets the best Dice value of 0.754±0.006; "2.5D&3D" achieves the best HD95 value of 3.851±0.914, and the best VR70 of 0.800±0.010. It indicates that combinations of different dimension models have their own advantages in terms of metrics. It is clear that by using our UMM framework on two segmentation models, the performance can be improved compared to single dimensional baseline. Moreover, UMM-Net boosts Dice scores by 0.9%-2.3%, HD95 scores by 20.3%-32.0%, Recall scores by 5.8%-8.1%, and VR70 by 1.6%-6.0% compared to two network combinations. It shows that

TABLE III: The comparison results on **BraTS2020**. † indicates our UMM-Net achieves statistically significant results (p-value < 0.05) in comparison with SOTA. Red and blue values mean the best and the second best among these methods.

		WT			TC			ET			Average			
Method	Type	Dice ↑	HD95 ↓	Recall ↑	Dice ↑	HD95 ↓	Recall ↑	Dice ↑	HD95 ↓	Recall ↑	mDice ↑	mHD95 ↓	mRecall ↑	mVR70 ↑
U-Net [7]	2D	0.897	4.995	0.880	0.781	6.019	0.779	0.716	6.661	0.780	0.798 †	5.892 †	0.813 †	0.831
UNet++ [9]	2D	0.892	4.123	0.902	0.757	7.178	0.739	0.741	5.162	0.793	0.797 †	5.488 †	0.811 †	0.813
Att. U-Net [10]	2D	0.845	15.174	0.936	0.782	16.381	0.809	0.716	9.095	0.777	0.781 †	13.550 †	0.840 †	0.791
Swin-Unet [22]	2D	0.917	2.813	0.902	0.822	5.002	0.829	0.734	7.471	0.754	0.824 †	5.096 †	0.829 †	0.862
3D U-Net [11]	3D	0.917	1.942	0.918	0.842	3.751	0.854	0.732	5.977	0.807	0.830	3.890 †	0.859	0.880
nnU-Net [38]	3D	0.918	2.175	0.909	0.847	3.411	0.854	0.736	4.842	0.798	0.834	3.476	0.854	0.880
SegResNet [8]	3D	0.917	1.208	0.911	0.836	2.710	0.832	0.731	3.212	0.762	0.828	2.376	0.835	0.891
TransBTS [21]	3D	0.911	3.360	0.906	0.836	2.987	0.838	0.740	3.403	0.779	0.829 †	3.250 †	0.841	0.898
UNETR [6]	3D	0.902	4.305	0.891	0.813	5.740	0.817	0.732	4.643	0.796	0.815 †	4.896 †	0.835	0.862
Swin UNETR [35]	3D	0.917	2.857	0.924	0.826	4.314	0.872	0.749	4.503	0.831	0.830 †	3.891 †	0.876	0.862
2.5D UNet	2.5D	0.917	3.853	0.911	0.831	3.909	0.841	0.728	5.889	0.807	0.825 †	4.550 †	0.853	0.862
UMM-Net (Ours)	2D&2.5D&3D	0.924	1.045	0.911	0.843	2.882	0.854	0.756	3.192	0.805	0.841	2.373	0.857	0.898



(a) Input (b) GT (c) **Ours** (d) UNet++ (e) Att. U-Net (f) Swin-Unet (g) nnU-Net (h) SegResNet (i) TransBTS (j) Swin UR

Fig. 5: Visual comparisons on the WT, TC, and ET segmentation results on **BraTS2020**. Apparently, our UMM-Net achieves best visual results. **Pink** represents WT. **Yellow** represents TC. **Green** represents ET.

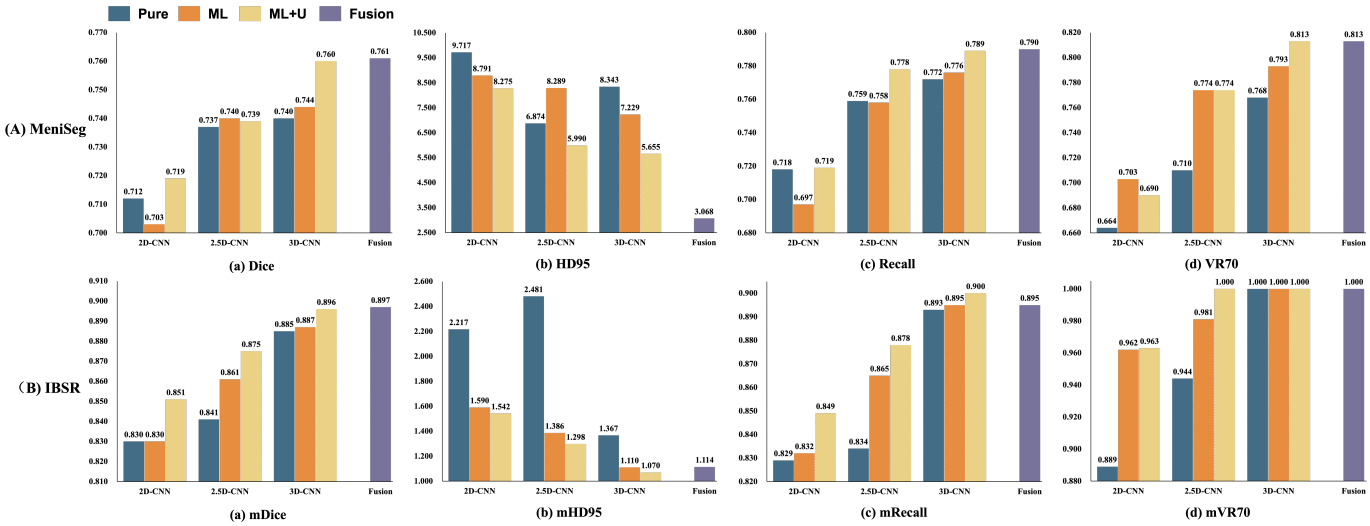


Fig. 6: Ablation studies on (A) **MeniSeg** and (B) **IBSR**. We denote the single segmentation network as **Pure**, mutual learning of three segmentation networks as **ML**, mutual learning with an uncertainty-aware knowledge gating mechanism as **ML+U**, and the final performance as **Fusion**.

TABLE IV: Quantitative results of our framework with two segmentation models on **MeniSeg**. **Dimensions** denote which dimensions of U-Net are utilized in our framework. Red values mean the best among these methods.

Dimensions	Dice ↑	HD95 ↓	Recall ↑	VR70 ↑
U-Net	0.712±0.003	9.717±0.195	0.718±0.014	0.664±0.007
3D U-Net	0.740±0.000	8.343±0.239	0.772±0.014	0.768±0.002
2.5D UNet	0.737±0.006	6.874±1.200	0.759±0.020	0.710±0.012
2D&2.5D	0.744±0.021	4.514±1.834	0.731±0.031	0.767±0.056
2D&3D	0.754±0.006	4.247±0.379	0.747±0.011	0.761±0.434
2.5D&3D	0.752±0.016	3.851±1.287	0.736±0.030	0.800±0.010
UMM-Net (Ours)	0.761±0.009	3.068±0.914	0.790±0.006	0.813±0.029

TABLE V: **Attention U-Net** is utilized as the backbone on **MeniSeg**, which denoted as **UMM+Att.**. "Att." denotes Attentinon U-Net. The learnt sub-networks are denoted as 2D/2.5D/3D Att.+UMM. Red means the best result.

Method	Dice ↑	HD95 ↓	Recall ↑	VR70
2D Att.	0.697±0.010	16.286±10.224	0.681±0.020	0.652±0.033
2.5D Att.	0.735±0.007	8.087±4.564	0.688±0.025	0.716±0.076
3D Att.	0.741±0.018	9.797±5.055	0.776±0.018	0.749±0.043
2D Att.+UMM	0.702±0.002	7.927±0.943	0.698±0.035	0.729±0.021
2.5D Att.+UMM	0.742±0.001	6.216±2.608	0.736±0.027	0.787±0.048
3D Att.+UMM	0.754±0.012	6.400±1.183	0.766±0.044	0.774±0.025
UMM+Att.	0.758±0.035	4.406±2.067	0.741±0.023	0.832±0.056

adding more segmentation networks leads to a more stable generalization ability, as complementary information from different dimension subnetworks is successfully fused via our uncertainty-aware mutual learning strategy.

V. Discussion

A. The Generalization of Framework

Our framework is a general structure to be compatible with different backbones. In this section, we discuss the application of our framework on Attention U-Net, and a cross-combination network (i.e., we use 2D Swin-Unet, 2.5D UNet and 3D TransBTS on our framework).

1) Application on Attention U-Net: As shown in Table V, each Att.+UMM achieves better performance compared to its corresponding single dimensional Attention U-Net. Except for UMM+Att., 3D Att.+UMM achieves the best Dice score of 0.754±0.012 with improvements of 1.8%. 2.5D Att.+UMM achieves the best HD95 and VR70. 2D Att.+UMM has 51.3% improvements on HD95. What's more, UMM+Att, which fuses the sub-network results, has further improvements. It demonstrates that our UMM framework not only works for U-Net but also works for other baseline networks

2) Application on Cross Networks: To demonstrate the generality of our framework, we also examine the performance of our framework on BraTS2020 when utilizing Swin-Unet (2D-CNN), 2.5D UNet, and TransBTS (3D-CNN), which is denoted as UMM+Cross. As shown in Table VI, our UMM boosts the performance of either of the three models; while the UMM+Cross which utilizes the uncertainty-based fusion scheme can further improve the performance. Compared to the baseline Swin-Unet in 2D, UMM+Cross has 2.9%, 60.1%, and 4.4% significant improvements on mDice, mHD95, mRecall, respectively. Compared to TransBTS in 3D, UMM+Cross has 2.4%, 37.5%, and 2.8% improvements. It is worth noting that, compared to the 2.5D U-Net, UMM+Cross achieves greater improvements of 2.8%, 55.3%, and 1.3% in mDice, mHD95, mRecall, respectively, than UMM-Net (in Section IV), which shows 1.9%, 47.8%, and 0.4% improvements in same metrics.

B. Analysis of Model Uncertainty

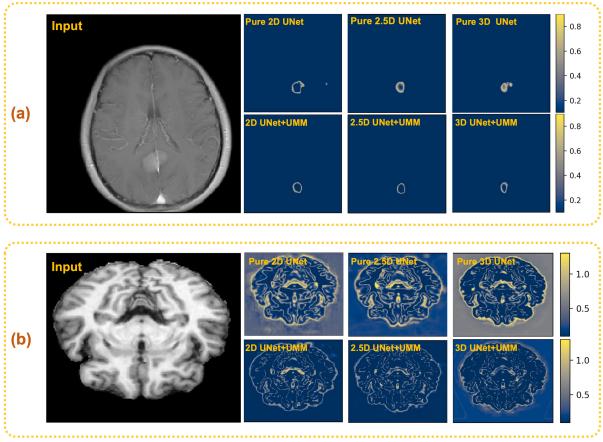


Fig. 7: Visualization of uncertainty maps on (a) **MeniSeg**, and (b) **IBSR**. By utilizing our framework **UMM**, the value of model uncertainty decreases significantly.

Generally, the more uncertain the model, the more reliable the model segmentation results tend to be. In this section, we analyze the validity of our approach by observing uncertainty maps. As can be seen from Fig. 7, the higher the yellow brightness is, the higher uncertainty the model has, and the model uncertainty decreases greatly with our mutual learning scheme. This is due to the fact that mutual learning with multiple supervision can be seen as a type of regularization.

C. Model Complexity and Inference Cost

As listed in Table VII, we present the number of parameters and count FLOPs of the models on BraTS2020 dataset. Here, we compute the average rank (i.e., mRank) of mDice, mHD95, mRecall, and mVR70 among these methods. Our UMM-Net achieves the top-1 mRank while having the minimum number of parameters except for the baseline network (i.e., 2D/2.5D/3D U-Net). Note that although our method does not has the minimum FLOPs, it outperforms comparable methods that have a similar FLOPs, such as UNet++, UNETR, and Swin UNETR, in terms of mRank and parameter amount. This suggests our method gets a better trade-off in efficiency and accuracy.

D. Limitations of Our Framework

Since not all medical data contains 3-dimensional information, our framework cannot be applied to 2D X-ray images and electron microscope images. What's more, during training, we need to maintain the simultaneous training of three models, which requires a relatively large amount of memory and consumes more training time compared to training a single dimensional model.

TABLE VI: The comparison of **BraTS** between our UMM+Cross and its baseline methods. UMM+Cross utilizes 2D Swin-Unet, 2.5D UNet, and 3D TransBTS. The learnt sub-networks are denoted as Swin-Unet/2.5D UNet/TransBTS+UMM. † indicates our UMM+Cross achieves statistically significant results on mDice, mHD95 or mRecall (p-value < 0.05) in comparison with baseline methods. Red values mean the best and second best among these methods.

		WT			TC			ET			Average			
Method	Type	Dice ↑	HD95 ↓	Recall ↑	Dice ↑	HD95 ↓	Recall ↑	Dice ↑	HD95 ↓	Recall ↑	mDice ↑	mHD95 ↓	mRecall ↑	mVR70 ↑
Swin-Unet [22]	2D	0.917	2.813	0.902	0.822	5.002	0.829	0.734	7.471	0.754	0.824 †	5.096 †	0.829 †	0.862
2.5D UNet	2.5D	0.917	3.853	0.911	0.831	3.909	0.841	0.728	5.889	0.807	0.825 †	4.550 †	0.853	0.862
TransBTS [21]	3D	0.911	3.360	0.906	0.836	2.987	0.838	0.740	3.403	0.779	0.829 †	3.250 †	0.841 †	0.898
Swin-Unet+UMM	2D	0.917	1.658	0.908	0.838	3.082	0.843	0.746	3.218	0.814	0.834	2.653	0.855	0.898
2.5D UNet+UMM	2.5D	0.918	1.870	0.920	0.844	3.082	0.855	0.743	3.426	0.833	0.835	2.793	0.869	0.889
TransBTS+UMM	3D	0.920	2.453	0.924	0.846	3.795	0.848	0.749	3.099	0.796	0.838	3.116	0.856	0.898
UMM+Cross (Ours)	2D&2.5D&3D	0.926	0.996	0.919	0.854	2.219	0.856	0.766	2.881	0.819	0.848	2.032	0.865	0.911

TABLE VII: Comparison of number of parameters and FLOPs for models on **BraTS2020**. "mRank" means the mean rank of mDice, mHD95, mRecall and mVR70 among these methods.

Method	mRank	#Params (M)	FLOPs (G)
U-Net [7]	10.50	0.496	19.257
UNet++ [9]	11.00	6.438	95.964
Att. U-Net [10]	10.75	5.863	147.517
Swin-Unet [22]	8.50	6.303	64.91
3D U-Net [11]	3.50	1.440	35.496
nnU-Net [38]	3.75	5.752	190.147
SegResNet [8]	4.75	4.702	62.788
TransBTS [21]	4.00	31.775	111.509
UNETR [6]	8.25	102.222	85.757
Swin UNETR [35]	4.00	15.705	84.399
2.5D UNet	6.50	1.323	30.196
UMM-Net (Ours)	1.50	3.259	84.949

VI. Conclusion

This work presents an uncertainty-aware multi-dimensional mutual learning framework to boost models for brain tissue and brain tumor segmentation in MRI. We show that low-dimensional 2D/2.5D-CNN models can capture fine features and further enhance the performance of stronger high-dimensional 3D-CNN models, even though the latter have larger spatial receptive fields and more parameters. This issue has not been exploited before, and our key idea is to design a multi-dimensional mutual learning scheme, built upon 2D-CNN, 2.5D-CNN, and 3D-CNN, to provide predictions as the regularization to each other. We also introduce the uncertainty-aware gating mechanism into the mutual learning process to select qualified and reliable regions, which leads to more credible learning. Experiments on three datasets demonstrate that our proposed framework effectively enhances the performance of the backbone networks.

Acknowledgment

This work has been approved by the ethics committee of Tianjin Huanhu Hospital, Tianjin, China (No. 2022-046, date: 2022-5-9); and was supported by the grant from Tianjin Natural Science Foundation (Grant No. 20JCYBJC00960), Guangzhou Municipal Science and Technology Project (Grant No. 2023A03J0671), the Huazhu Fu's A*STAR Central Research Fund, and AISG Tech Challenge Funding (AISG2-TC-2021-003).

References

[1] T. Logeswari and M. Karnan, "An improved implementation of brain tumor detection using segmentation based on soft computing," *Cancer Research and Experimental Oncology*, vol. 2, pp. 006–014, 2010.
[2] E. Yee, D. Ma, K. Popuri *et al.*, "3d hemisphere-based convolutional neural network for whole-brain mri segmentation," *Comput. Med. Imaging Graphics*, vol. 95, p. 102000, 2022.
[3] Y. Zhao, H. Li *et al.*, "Knowledge-aided convolutional neural network for small organ segmentation," *JBHI*, vol. 23, pp. 1363–1373, 2019.
[4] L. Liu, F. Wu, and J. Wang, "Multi-receptive-field cnn for semantic segmentation of medical images," *JBHI*, vol. 24, pp. 3215–3225, 2020.
[5] A. Sinha and J. Dolz, "Multi-scale self-guided attention for medical image segmentation," *JBHI*, vol. 25, pp. 121–130, 2021.
[6] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, "Unetr: Transformers for 3d medical image segmentation," in *Proc. WACV*, 2022.
[7] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015.
[8] A. Myronenko, "3d mri brain tumor segmentation using autoencoder regularization," in *Proc. MICCAI Workshop*, 2018.
[9] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Proc. DLMIA Workshop*, 2018.
[10] O. Oktay *et al.*, "Attention U-Net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.
[11] Ö. Çiçek, A. Abdulkadir *et al.*, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *Proc. MICCAI*, 2016.
[12] H.-C. Shin, H. R. Roth, M. Gao *et al.*, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *TMI*, vol. 35, pp. 1285–1298, 2016.
[13] G. Wang, W. Li, S. Ourselin *et al.*, "Automatic brain tumor segmentation based on cascaded convolutional neural networks with uncertainty estimation," *Front. Comput. Neurosci.*, vol. 13, p. 56, 2019.
[14] K. Ono, Y. Iwamoto *et al.*, "Automatic segmentation of infant brain ventricles with hydrocephalus in mri based on 2.5 d u-net and transfer learning," *Int. J. Image and Graphics*, vol. 8, pp. 42–46, 2020.
[15] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng, "H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes," *TMI*, vol. 37, pp. 2663–2674, 2018.
[16] A. Stadlbauer *et al.*, "Improved delineation of brain tumors: an automated method for segmentation based on pathologic changes of 1h-mrsi metabolites in gliomas," *Neuroimage*, vol. 23, pp. 454–461, 2004.
[17] W. Deng, W. Xiao, H. Deng, and J. Liu, "Mri brain tumor segmentation with region growing method based on the gradients and variances along and inside of the boundary curve," in *Proc. BMEI*, 2010.
[18] S. Ho *et al.*, "Level-set evolution with region competition: automatic 3-d segmentation of brain tumors," in *Proc. ICPR*, 2002.
[19] L.-H. Juang and M.-N. Wu, "Mri brain lesion image detection based on color-converted k-means clustering segmentation," *Measurement*, vol. 43, pp. 941–949, 2010.
[20] J. Liu, M. Li, J. Wang, F. Wu, T. Liu, and Y. Pan, "A survey of mri-based brain tumor segmentation methods," *Tsinghua Science and Technology*, vol. 19, pp. 578–595, 2014.
[21] W. Wang, C. Chen, M. Ding *et al.*, "Transbts: Multimodal brain tumor segmentation using transformer," in *Proc. MICCAI*, 2021.
[22] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," *arXiv preprint arXiv:2105.05537*, 2021.

[23] Y. Zhou, W. Huang, P. Dong *et al.*, "D-unet: a dimension-fusion u shape network for chronic stroke lesion segmentation," *TCBB*, 2019.
[24] Q. Zhu, B. Du, B. Turkbey, P. Choyke, and P. Yan, "Exploiting interslice correlation for mri prostate image segmentation, from recursive neural networks aspect," *Complexity*, vol. 2018, 2018.
[25] H. Cui, X. Liu, and N. Huang, "Pulmonary vessel segmentation based on orthogonal fused u-net++ of chest ct images," in *Proc. MICCAI*, 2019.
[26] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. CVPR*, 2018.
[27] M. Jaritz, T.-H. Vu, R. de Charette, E. Wirbel, and P. Pérez, "xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation," in *Proc. CVPR*, 2020.
[28] H. He, J. Zhang, Q. Zhang, and D. Tao, "Grapy-ml: Graph pyramid mutual learning for cross-dataset human parsing," in *Proc. AAAI*, 2020.
[29] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proc. CVPR*, 2017.
[30] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *NeurIPS*, vol. 30, 2017.
[31] W. Cui, Y. Liu, Y. Li *et al.*, "Semi-supervised brain lesion segmentation with an adapted mean teacher model," in *Proc. IPMI*, 2019.
[32] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Machine Learning Research*, 2016.
[33] L. Yu, S. Wang, X. Li, C.-W. Fu, and P.-A. Heng, "Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation," in *Proc. MICCAI*, 2019.
[34] F. Milletari *et al.*, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 3DV*, 2016.
[35] Y. Tang, D. Yang, W. Li *et al.*, "Self-supervised pre-training of swin transformers for 3d medical image analysis," in *Proc. CVPR*, 2022.
[36] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in mri images," *TMI*, vol. 35, pp. 1240–1251, 2016.
[37] K. He *et al.*, "Mask r-cnn," in *Proc. ICCV*, 2017.
[38] F. Isensee, J. Petersen, A. Klein *et al.*, "nnu-net: Self-adapting framework for u-net-based medical image segmentation," *arXiv preprint arXiv:1809.10486*, 2018.