SlimInterPAS for Brain Tumor Segmentation

Jinlin Dong and Yanjun Peng*

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, Shandong, China
yjpeng@sdust.edu.cn

Abstract. Accurate segmentation of enhancing tumor (ET), whole tumor (WT), and tumor core (TC) from multisequence magnetic resonance imaging (MRI) remains challenging because tumor morphology, signal characteristics, and spatial extent vary substantially across cases. We present SlimInterPAS, a focused extension of SlimUNETRV2 that modifies only the deepest encoder stage while retaining the original decoder, segmentation head, Dice–cross-entropy objective, and inference strategy. A three-dimensional axis adapter (AxisAdapter3D) exposes depth-, height-, and width-oriented responses, InterBottleneck models complementary cross-plane interactions, a position-adaptive spatial recalibration block (PASBlock) combines local and dilated contextual responses, and Gated Residual Fusion controls the incremental correction written back to the baseline representation. Evaluation is organized around controlled component ablation on the labeled training/reference set of the 2026 Brain Tumor Segmentation (BraTS) Generalizability Across Tumors (GoAT) challenge, paired per-case statistical analysis, computational-efficiency profiling, and a separate official hidden validation evaluation. The complete configuration provides the most favorable overall balance among the evaluated ablation variants while adding only a small parameter and patch-runtime overhead relative to the direct baseline. Training/reference-set evidence and hidden-set generalization are reported separately.

Keywords: Brain tumor segmentation · Multisequence magnetic resonance imaging · Lightweight volumetric segmentation · Axis-aware feature interaction · Gated residual refinement

1 Introduction

Gliomas exhibit substantial heterogeneity in morphology, intensity, and spatial distribution. Accurate delineation of enhancing tumor (ET), whole tumor (WT), and tumor core (TC) is important for quantitative assessment, treatment planning, and longitudinal analysis [1,2]. Multisequence magnetic resonance imaging (MRI) combines T1-weighted (T1), contrast-enhanced T1-weighted (T1ce), T2-weighted (T2), and fluid-attenuated inversion recovery (FLAIR) acquisitions,

* Corresponding author.

yet ambiguous boundaries, small enhancing foci, and acquisition variation remain challenging.

U-Net and its three-dimensional (3D) extensions established the encoder–decoder paradigm for volumetric segmentation [3,4], while nnU-Net showed the importance of jointly configuring preprocessing, optimization, and inference [5]. Recent efficient architectures pursue stronger context with controlled cost: MedNeXt scales convolutional networks [6]; PHNet uses multilayer-perceptron permutation for volumetric feature mixing [7]; Slim UNETR combines convolutional and Transformer processing [8]; and SlimUNETRV2 reorganizes lightweight volumetric representation learning through convolutional feature mixing, state-space modeling, lightweight attention, and transposed-convolution decoding [9]. Other high-level approaches include self-supervised Swin UNETR [10], large-kernel 3D UX-Net [11], spatial–channel attention in UNETR++ [12], diffusion-guided refinement in Diff-UNet [13], and tri-oriented state-space modeling in SegMamba-V2 [14]. Together, these studies motivate structured contextual modeling without requiring a complete high-cost backbone redesign.

SlimInterPAS uses SlimUNETRV2 as the direct baseline and changes only its deepest encoder representation. Conventional learnable 3D convolutions can encode directional information, but the spatial axes are mixed jointly inside local kernels. We instead introduce an explicit directional inductive bias at the lowest-resolution semantic stage: AxisAdapter3D exposes depth-, height-, and width-oriented responses; InterBottleneck couples depth separately with the two in-plane directions through (d, h) and (d, w) interaction; PASBlock recalibrates local and dilated context; and Gated Residual Fusion writes only bounded incremental corrections back to the baseline feature. High-resolution skip features, the decoder, segmentation head, training objective, and inference pathway are unchanged.

Our contributions are threefold: (1) an axis-factorized deepest-stage interaction that preserves the original high-resolution pathway; (2) position-adaptive recalibration followed by conservative gated residual write-back; and (3) a BraTS Generalizability Across Tumors (BraTS-GoAT) evaluation that clearly separates controlled training/reference-set ablation and paired analysis from official hidden validation, together with computational-efficiency profiling.

2 Methods

2.1 Overview

Let the four MRI sequences be represented by

$$X \in \mathbb{R}^{B\times 4\times D\times H\times W}, \tag{1}$$

where the channels correspond to T1, T1ce, T2, and FLAIR. SlimUNETRV2 encodes the input into a hierarchy of feature maps and passes multi-scale information to its decoder through skip connections. We denote the deepest encoder feature by

$$x \in \mathbb{R}^{B\times C\times d\times h\times w}. \tag{2}$$

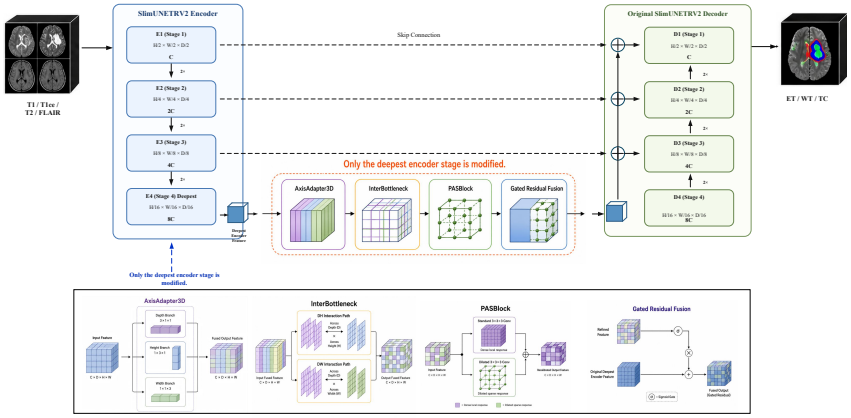

Fig. 1. Schematic overview of SlimInterPAS. The SlimUNETRV2 backbone is shown at the stage level for clarity. Only the deepest encoder representation is refined by AxisAdapter3D, InterBottleneck, PASBlock, and Gated Residual Fusion; the earlier high-resolution skip pathway and downstream decoder remain unchanged.

As illustrated in Fig. 1, the SlimUNETRV2 backbone is shown at the stage level for clarity. SlimInterPAS is inserted only after the deepest encoder stage and follows the four-module sequence shown in the figure:

$$\begin{aligned} x \rightarrow \text{AxisAdapter3D} \rightarrow \text{InterBottleneck} \rightarrow \text{PASBlock} \\ \rightarrow \text{Gated Residual Fusion} \rightarrow \hat{x}. \end{aligned} \tag{3}$$

The refined feature $\hat{x}$ is returned to the unchanged downstream SlimUNETRV2 pathway. No auxiliary segmentation head or additional supervision target is introduced.

2.2 AxisAdapter3D and InterBottleneck

AxisAdapter3D corresponds to the first refinement block in Fig. 1. It uses three depthwise-separable branches to expose depth-, height-, and width-oriented responses:

$$x_d = \phi_d(x), \qquad x_h = \phi_h(x), \qquad x_w = \phi_w(x), \tag{4}$$

where ϕ_d, ϕ_h, and ϕ_w use depthwise kernels of size $3 \times 1 \times 1$, $1 \times 3 \times 1$, and $1 \times 1 \times 3$, respectively. Each depthwise operation is followed by a $1 \times 1 \times 1$ pointwise convolution, instance normalization, and GELU activation. The output of AxisAdapter3D is therefore the triplet (x_d, x_h, x_w) with the same spatial size and channel width as x.

InterBottleneck receives these directional features and performs the two complementary interaction paths depicted in Fig. 1: a depth–height (DH) path and

a depth–width (DW) path. Both paths retain depth as the common anatomical axis while coupling it separately with the two in-plane directions. For path $r \in \{\mathrm{DH}, \mathrm{DW}\}$,

$$A_r = \mathrm{softmax}\big(\bar{Q}_r \bar{K}_r^{\top}\big), \qquad U_r = A_r V_r, \tag{5}$$

where $\bar{Q}_r$ and $\bar{K}_r$ denote ℓ_2-normalized query and key representations. The interacted features are reshaped to 3D and receive the corresponding value residual,

$$\widetilde{U}_r = \mathrm{reshape}(U_r) + V, \tag{6}$$

after which the two paths are concatenated and compressed by a grouped $3\times3\times3$ convolution:

$$z = x + \mathrm{GConv}_{3\times3\times3}\left([\widetilde{U}_{\mathrm{DH}}; \widetilde{U}_{\mathrm{DW}}]\right). \tag{7}$$

A channel-wise feed-forward block then performs residual refinement,

$$x_{\mathrm{inter}} = \mathrm{LN}(z + \mathrm{FFN}(\mathrm{LN}(z))). \tag{8}$$

This factorization avoids constructing a single all-pairs interaction over the complete 3D token set, and restricting it to the deepest feature map controls spatial cost while preserving the original high-resolution skips.

2.3 PASBlock and Gated Residual Fusion

PASBlock is the third refinement block in Fig. 1. It contains a standard $3\times3\times3$ convolutional branch and a dilation-3 $3\times3\times3$ contextual branch:

$$y = \rho(\mathrm{IN}(\mathrm{Conv}_{3\times3\times3}(x_{\mathrm{inter}}))), \tag{9}$$

$$a = \rho\Big(\mathrm{IN}\Big(\mathrm{Conv}^{\delta=3}_{3\times3\times3}(x_{\mathrm{inter}})\Big)\Big), \tag{10}$$

where ρ denotes ReLU. The dilated response is converted into a position-dependent modulation map and applied to the local branch:

$$x_{\mathrm{pas}} = (1 + \sigma(a)) \odot y. \tag{11}$$

Thus, PASBlock preserves the dense local response y while using the enlarged receptive field of the dilated branch to recalibrate it without spatial downsampling.

Gated Residual Fusion is the final block in the figure. The gate is predicted jointly from the original deepest feature, the InterBottleneck output, and the PASBlock output:

$$g = \sigma(W_2\, \rho(\mathrm{IN}(W_1[x; x_{\mathrm{inter}}; x_{\mathrm{pas}}]))). \tag{12}$$

Two learnable scalar update parameters are bounded with tanh,

$$\alpha = \tanh(s_{\mathrm{inter}}), \qquad \beta = \tanh(s_{\mathrm{pas}}), \tag{13}$$

and the final deepest representation is

$$\hat{x} = \mathrm{IN}[x + g \odot (\alpha(x_{\mathrm{inter}} - x) + \beta(x_{\mathrm{pas}} - x_{\mathrm{inter}}))] \,. \tag{14}$$

In the reported configuration, s_{inter} and s_{pas} are initialized to 0.14 and 0.04, respectively. The two subtraction terms in Eq. (14) represent the incremental changes introduced by InterBottleneck and PASBlock. The spatial gate therefore controls only the new corrections, while the original deepest feature x remains an explicit residual anchor.

2.4 Design Rationale and Information Flow

The four modules form the same sequential refinement chain shown in Fig. 1. AxisAdapter3D first exposes directional local responses. InterBottleneck then performs complementary DH and DW interactions on those factorized features. PASBlock subsequently recalibrates the interacted representation with local and dilation-3 contextual responses. Finally, Gated Residual Fusion combines the two incremental updates with the original feature through a learned spatial gate. This ordering separates directional adaptation, cross-plane interaction, spatial recalibration, and conservative write-back, while restricting all added processing to the lowest-resolution semantic stage.

2.5 Training Objective

All A0–A4 variants use the same Dice plus cross-entropy segmentation objective and the same deep-supervision setting supplied by the training configuration. SlimInterPAS introduces no auxiliary loss, prediction head, or supervision target; the component ablation therefore changes only the indicated deepest-stage refinement modules.

2.6 BraTS-GoAT Data and Evaluation Protocol

The reported challenge model is fitted only with data distributed through the BraTS-GoAT sub-challenge; no external dataset, previous BraTS challenge dataset, or pretrained segmentation checkpoint is used. The data-usage citations follow the BraTS benchmark and MedPerf guidance [2,15,16,17]. The 1,351 labeled cases correspond to `MICCAI2024-BraTS-GoAT-TrainingData-With-GroundTruth`; validation images are distributed separately as `MICCAI2024-BraTS-GoAT-ValidationData` and evaluated by the official BraTS-GoAT 2026 service with hidden labels. The training release is registered locally as `Dataset024_GoAT`, an implementation-specific nnU-Net identifier.

Annotations contain background (0), necrotic/non-enhancing tumor core (NCR, 1), edema (ED, 2), and ET (3), with ET = $\{3\}$, TC = $\{1, 3\}$, and WT = $\{1, 2, 3\}$. nnU-Net preprocessing includes foreground cropping, resampling, per-sequence normalization, patch sampling, online spatial/intensity augmentation, and sliding-window inference with Gaussian weighting and mirroring.

The 3D full-resolution plan uses 1 mm^3 isotropic spacing, a $128 \times 160 \times 112$ patch, and batch size 2.

For the `fold=all` ablation, all 1,351 labeled cases are used for fitting and the same labeled reference set is used for local quantitative analysis; these are therefore training/reference-set resubstitution metrics, not independent validation estimates. Generalization is reported only from the official hidden validation service. Dice and the 95th-percentile Hausdorff distance (HD95) are computed case-wise for each composite region and then averaged across cases, rather than globally pooling voxels or matching lesions. If reference and prediction are both empty, Dice= 1 and HD95= 0; if exactly one is empty, Dice= 0 and HD95 is the physical image diagonal. The policy is identical for A0–A4.

2.7 Implementation Details

SlimUNETRV2 follows the public implementation at `https://github.com/deepang-ai/Slim-UNETRV2`. For source traceability, the baseline source released with this study is pinned to commit `1479bb0e2f086548deb9d31aa3502ae0b289c3af`; no pretrained checkpoint is loaded. A0–A4 construct the same base network through the same nnU-Net v2 plan-driven initialization path, and SlimInterPAS is appended only to the deepest encoder stage for the enabled variants. SlimInterPAS-specific convolutional and normalization layers use their standard PyTorch initialization. The two Gated Residual Fusion update-scale parameters associated with the InterBottleneck and PASBlock corrections are initialized to 0.14 and 0.04, respectively.

Training uses 350 epochs with 250 training and 50 validation iterations per epoch. AxisAdapter3D uses $3 \times 1 \times 1$, $1 \times 3 \times 1$, and $1 \times 1 \times 3$ kernels; InterBottleneck uses eight heads over the DH and DW paths; PASBlock uses dilation 3. A0–A4 share data, preprocessing, optimization schedule, objective, deep-supervision setting, augmentation, and inference; only the indicated deepest-stage components differ. For the five models included in the efficiency comparison (nnU-Net, MedNeXt, PHNet, SlimUNETRV2, and SlimInterPAS), all models were trained from scratch using the same BraTS-GoAT training cases and the same `fold=all` setting; no pretrained segmentation weights or published checkpoints were used. Dataset preprocessing and evaluation definitions were kept consistent across methods, while architecture-specific settings followed their respective implementations. Experiments use an NVIDIA GeForce RTX 4090 graphics processing unit (GPU, 24 GB), Intel Xeon Platinum 8352V central processing unit (CPU), PyTorch 2.7.0, and NVIDIA Compute Unified Device Architecture (CUDA) 11.8.

3 Results

3.1 Component-wise Ablation on the GoAT Training/Reference Set

Table 1 uses the same 350-epoch fold-all protocol for all variants. A0 is the SlimUNETRV2 baseline; A1 adds AxisAdapter3D+InterBottleneck; A2 adds

Table 1. Component-wise ablation on the 1,351-case local GoAT fold-all training/reference set. Higher Dice and lower HD95 are better. These values are resubstitution metrics rather than independent validation estimates.

Variant	AxisAdapter3D + InterBottleneck	PASBlock	Gated Residual Fusion	Dice ↑				HD95 (mm) ↓			
				ET	WT	TC	Mean	ET	WT	TC	Mean
A0 SlimUNETRV2	–	–	–	0.8842	0.9408	0.9369	0.9206	9.8735	3.9334	3.3334	5.7134
A1	✓	–	–	0.8854	0.9392	**0.9373**	0.9206	10.6407	3.2402	**3.3047**	5.7285
A2	–	✓	–	0.8850	**0.9410**	0.9356	0.9205	10.1396	3.2089	3.4960	5.6148
A3	✓	✓	–	0.8844	0.9403	0.9362	0.9203	10.1667	3.2437	3.8138	5.7414
A4 SlimInterPAS	✓	✓	✓	**0.8862**	0.9409	**0.9373**	**0.9215**	**9.7918**	**3.0974**	3.4031	**5.4308**

Table 2. Per-case variability and paired A0–A4 comparison on the local GoAT training/reference set. Dice is reported as mean±SD with 95% confidence intervals of the mean. HD95 is reported as median [IQR] with percentile-bootstrap 95% confidence intervals of the median from 10,000 resamples. Two-sided paired Wilcoxon signed-rank tests are used.

Metric	Region	A0 summary	A4 summary	A0 95% CI	A4 95% CI	Wilcoxon p
Dice	ET	0.8842 ± 0.1679	0.8862 ± 0.1665	[0.8753, 0.8932]	[0.8773, 0.8951]	6.71×10^{-7}
Dice	WT	0.9408 ± 0.0688	0.9409 ± 0.0698	[0.9372, 0.9445]	[0.9372, 0.9446]	0.2044
Dice	TC	0.9369 ± 0.0908	0.9373 ± 0.0914	[0.9320, 0.9417]	[0.9324, 0.9422]	0.0036
Dice	Mean	0.9206 ± 0.0900	0.9215 ± 0.0900	[0.9158, 0.9254]	[0.9167, 0.9263]	2.44×10^{-6}
HD95	ET	1.0000 [1.0000, 1.5255]	1.0000 [1.0000, 1.4142]	[1.0000, 1.0000]	[1.0000, 1.0000]	0.28
HD95	WT	1.4142 [1.0000, 3.0000]	1.4142 [1.0000, 2.6458]	[1.4142, 1.7321]	[1.4142, 1.4142]	0.0015
HD95	TC	1.0000 [1.0000, 2.0000]	1.0000 [1.0000, 2.0000]	[1.0000, 1.4142]	[1.0000, 1.4142]	0.65
HD95	Mean	1.4714 [1.0000, 2.5500]	1.4142 [1.0000, 2.2500]	[1.4142, 1.5811]	[1.4142, 1.4714]	0.0010

PASBlock only; A3 combines AxisAdapter3D, InterBottleneck, and PASBlock without Gated Residual Fusion; and A4 is the complete SlimInterPAS. When Gated Residual Fusion is disabled, the output of the last enabled refinement component is passed directly to the unchanged downstream network. Only the indicated deepest-stage components change.

A4 obtains the best overall mean Dice (0.9215) and mean HD95 (5.4308 mm), compared with 0.9206 and 5.7134 mm for A0. A1–A3 do not consistently improve every region or metric, showing that the complete model is not a simple accumulation of independently beneficial branches. A4 gives the best ET Dice/HD95 and WT HD95; A2 gives the best WT Dice; A1 and A4 tie on TC Dice, while A1 gives the best TC HD95. This pattern supports Gated Residual Fusion as a controlled write-back mechanism.

3.2 Per-case Variability and Paired Statistical Analysis

Because A0 and A4 are evaluated on the same 1,351 cases, Table 2 reports Dice as mean±standard deviation (SD) with 95% confidence intervals (CIs) of the mean. Because case-wise HD95 is strongly skewed, HD95 is summarized using the median and interquartile range (IQR), together with percentile-bootstrap 95% CIs of the median estimated from 10,000 bootstrap resamples. Two-sided paired Wilcoxon signed-rank tests are used for the matched A0–A4 comparisons, with statistical significance assessed at a two-sided $\alpha = 0.05$ level.

Table 3. Computational efficiency under the same hardware and profiling protocol. Runtime and throughput refer to a single $1 \times 4 \times 128 \times 128 \times 128$ input patch.

Method	Runtime (s/patch) ↓	Throughput (patches/s) ↑	Inference GPU Alloc. (GB) ↓	Inference GPU Reserved (GB) ↓	Params (M) ↓	Trainable Params (M) ↓	MACs (G) ↓	Train Peak Mem. (GB) ↓	Train Time (s/epoch) ↓	Model Size (MB) ↓
nnU-Net	**0.0358**	**27.9330**	**1.7360**	**2.4805**	31.1982	31.1982	539.5591	11.7	**128.4**	119.0
MedNeXt	0.1145	8.7336	4.8412	6.7520	10.6163	10.6163	**252.7164**	16.9	347.6	40.5
PHNet	0.0958	10.4384	2.6205	3.2598	24.9135	24.9135	1027.1128	14.3	294.2	95.0
SlimUNETRV2	0.1176	8.5034	2.5834	3.0539	**10.1780**	**10.1780**	306.9513	**10.7**	361.8	**38.8**
SlimInterPAS	0.1181	8.4674	2.2928	3.1004	10.3415	10.3415	566.9292	11.2	379.6	39.4

The paired Dice analysis detects differences for ET, TC, and the per-case mean, but not WT. For HD95, A4 narrows the ET and WT IQRs and lowers the median per-case mean HD95 from 1.4714 to 1.4142 mm. Significant paired differences are observed for WT ($p = 0.0015$) and mean HD95 ($p = 0.0010$), whereas ET ($p = 0.28$) and TC ($p = 0.65$) do not show significant paired differences. Because these are fold-all resubstitution measurements, the paired analyses characterize component behavior rather than independent generalization.

3.3 Computational Efficiency

Table 3 profiles all five methods on the same workstation equipped with an NVIDIA GeForce RTX 4090 (24 GB) and Intel Xeon Platinum 8352V CPU. Inference uses 32-bit floating-point precision (FP32), one $1 \times 4 \times 128 \times 128 \times 128$ patch, 20 warm-up passes, and 100 timed passes; throughput is the reciprocal of mean patch runtime. GPU allocation/reservation, parameters, trainable parameters, and multiply–accumulate operations (MACs) use the same input. Training peak memory, time per epoch, and model-file size are also measured on the common workstation.

Relative to SlimUNETRV2, SlimInterPAS changes runtime from 0.1176 to 0.1181 s/patch, throughput from 8.5034 to 8.4674 patches/s, trainable parameters from 10.1780 to 10.3415 M, and measured MACs from 306.9513 to 566.9292 G. Thus, the added modules have small parameter and observed patch-runtime overhead but a higher operation count. nnU-Net is the fastest method and MedNeXt has the lowest measured MAC count; we therefore describe SlimInterPAS as parameter-compact relative to its direct baseline rather than claiming globally optimal computational efficiency.

3.4 Official BraTS-GoAT 2026 Validation Results

The official hidden-label validation service returned ET/WT/TC Dice scores of 0.7830/0.8680/0.8130 (mean 0.8213) and normalized surface Dice (NSD) scores of 0.5380/0.4670/0.4870 (mean 0.4973). These scores are reported separately from fold-all measurements; HD95 was not included in the available official validation record.

4 Discussion

SlimInterPAS confines structured interaction to the deepest SlimUNETRV2 stage. It does not assume that 3D convolution cannot learn directional informa-

tion; rather, it adds an explicit axis-factorized prior where spatial resolution is lowest, followed by local/dilated recalibration and bounded incremental write-back. This preserves the original high-resolution skips, decoder, objective, and inference pathway while controlling cross-plane interaction cost.

A1–A3 are not uniformly better than A0, whereas A4 provides the best overall mean Dice and HD95, supporting the complete refinement chain rather than an additive claim for every branch in isolation. The fold-all experiment is explicitly treated as resubstitution evidence for component analysis, while independent generalization is represented by the official hidden validation result. Efficiency measurements show small parameter and observed patch-runtime overhead relative to SlimUNETRV2, while the measured MAC count is higher; accordingly, we do not claim that SlimInterPAS is optimal on every computational metric.

5 Conclusion

SlimInterPAS refines only the deepest SlimUNETRV2 representation through axis-aware interaction, position-adaptive recalibration, and gated residual write-back. Controlled GoAT ablation, paired case-wise analysis, efficiency profiling, and official hidden validation are reported as distinct sources of evidence. The results support localized deepest-stage semantic refinement without replacing the full segmentation pipeline, while the efficiency profile indicates a trade-off between small parameter/runtime overhead and increased measured MACs.

Code availability. The implementation, training configuration, ablation code, and evaluation scripts are publicly available at `https://github.com/Ddddsweet/SlimInterPAS`.

Acknowledgments. Data used in this publication were obtained as part of the Challenge project through Synapse ID (syn74274097). The authors thank the BraTS-GoAT organizers and data contributors for providing the challenge data and evaluation infrastructure.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Louis, D.N., Perry, A., Wesseling, P., et al.: The 2021 WHO Classification of Tumors of the Central Nervous System: A summary. Neuro-Oncology **23**(8), 1231–1251 (2021). `https://doi.org/10.1093/neuonc/noab106`
2. Menze, B.H., Jakab, A., Bauer, S., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans. Med. Imaging **34**(10), 1993–2024 (2015). `https://doi.org/10.1109/TMI.2014.2377694`
3. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Navab, N., et al. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). `https://doi.org/10.1007/978-3-319-24574-4_28`

4. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., et al. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 424–432. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_49
5. Isensee, F., Jaeger, P.F., Kohl, S.A.A., Petersen, J., Maier-Hein, K.H.: nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. Nat. Methods **18**(2), 203–211 (2021). https://doi.org/10.1038/s41592-020-01008-z
6. Roy, S., Koehler, G., Ulrich, C., et al.: MedNeXt: Transformer-driven scaling of ConvNets for medical image segmentation. In: Greenspan, H., et al. (eds.) MICCAI 2023. LNCS, vol. 14223, pp. 405–415. Springer, Cham (2023).
7. Lin, Y., Fang, X., Zhang, D., Cheng, K.-T., Chen, H.: Boosting convolution with efficient MLP-permutation for volumetric medical image segmentation. IEEE Trans. Med. Imaging **44**(5), 2341–2352 (2025). https://doi.org/10.1109/TMI.2025.3530113
8. Pang, Y., Liang, J., Huang, T., Chen, H., Li, Y., Li, D., Huang, L., Wang, Q.: Slim UNETR: Scale Hybrid Transformers to Efficient 3D Medical Image Segmentation Under Limited Computational Resources. IEEE Trans. Med. Imaging **43**(3), 994–1005 (2024). https://doi.org/10.1109/TMI.2023.3326188
9. Pang, Y., Liang, J., Yan, J., Hu, Y., Chen, H., Wang, Q.: Slim UNETRV2: 3D Image Segmentation for Resource-Limited Medical Portable Devices. IEEE Trans. Med. Imaging **45**(2), 542–553 (2026). https://doi.org/10.1109/TMI.2025.3602145
10. Tang, Y., Yang, D., Li, W., Roth, H.R., Landman, B., Xu, D., Nath, V., Hatamizadeh, A.: Self-supervised pre-training of Swin Transformers for 3D medical image analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20730–20740 (2022).
11. Lee, H.H., Bao, S., Huo, Y., Landman, B.A.: 3D UX-Net: A large kernel volumetric ConvNet modernizing hierarchical Transformer for medical image segmentation. In: International Conference on Learning Representations (ICLR) (2023).
12. Shaker, A.M., Maaz, M., Rasheed, H., Khan, S., Yang, M.-H., Khan, F.S.: UNETR++: Delving into efficient and accurate 3D medical image segmentation. IEEE Trans. Med. Imaging **43**(9), 3377–3390 (2024). https://doi.org/10.1109/TMI.2024.3398728
13. Xing, Z., Wan, L., Fu, H., Yang, G., Yang, Y., Yu, L., Lei, B., Zhu, L.: Diff-UNet: A diffusion embedded network for robust 3D medical image segmentation. Med. Image Anal. **105**, 103654 (2025). https://doi.org/10.1016/j.media.2025.103654
14. Xing, Z., Ye, T., Yang, Y., Cai, D., Gai, B., Wu, X.-J., Gao, F., Zhu, L.: SegMamba-V2: Long-range sequential modeling Mamba for general 3-D medical image segmentation. IEEE Trans. Med. Imaging **45**(1), 4–15 (2026). https://doi.org/10.1109/TMI.2025.3589797
15. Karargyris, A., Umeton, R., Sheller, M.J., Aristizabal, A., George, J., Wuest, A., Pati, S., et al.: Federated benchmarking of medical artificial intelligence with MedPerf. Nat. Mach. Intell. **5**, 799–810 (2023). https://doi.org/10.1038/s42256-023-00652-2
16. Baid, U., Ghodasara, S., Mohan, S., Bilello, M., Calabrese, E., Colak, E., Farahani, K., Kalpathy-Cramer, J., Kitamura, F.C., Pati, S., et al.: The RSNA-ASNR-MICCAI BraTS 2021 benchmark on brain tumor segmentation and radiogenomic classification. arXiv:2107.02314 (2021).

17. 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. Sci. Data **4**, 170117 (2017). https://doi.org/10.1038/sdata.2017.117