Report 1: Project Progress

Report 1: Brain Tumor Segmentation Using UNet and Foundation Models

University/Department: California State University, Dominguez Hills / Computer Science Department

Course/Thesis/Project Title: CSC 590 – Graduate Project

Semester/Year: Fall 2025

Project Title: Brain Tumor Segmentation Using UNet and Foundation Models

Student Name: Naushik Beladiya

Student ID: 212686840

Instructor / Committee Chair: Dr. Sahar Hooshmand

Date: October 26, 2025

1. Abstract / Summary of Progress

1.1 Motivation & Significance

Accurate, consistent delineation of brain tumor sub-regions in multi-modal MRI underpins diagnosis, surgical planning, radiotherapy targeting, and longitudinal monitoring. Manual segmentation is time-consuming and operator-dependent; it scales poorly across patients and scanners and introduces variability that can affect clinical decisions. Deep learning—especially U-Net–style encoder–decoder architectures—has established strong baselines for medical image segmentation, while foundation models adapted to the medical domain (e.g., MedSAM) promise more sample-efficient learning and promptable workflows. This project addresses both perspectives by building a from-scratch U-Net baseline and then fine-tuning MedSAM on the same data to study accuracy–efficiency trade-offs and practical robustness. The goals are aligned with the Cedars-Sinai project brief for "Brain Tumor Segmentation using UNet and Fine-Tuning Foundation Models" (Coach: Yimeng He).

1.2 Problem Statement

Given multi-modal brain MRI volumes (FLAIR, T1, T1ce, T2) from the Medical Segmentation Decathlon Task01_BrainTumour, automatically segment three clinically meaningful sub-regions—Whole Tumor (WT), Tumor Core (TC), and Enhancing Tumor (ET)—with voxel-wise labels suitable for volumetric analysis. Evaluate performance using Dice similarity coefficient and 95th percentile Hausdorff distance (HD95), and quantify runtime/VRAM to support an apples-to-apples comparison between (i) a supervised U-Net baseline and (ii) a fine-tuned MedSAM.

1.3 Objectives & Research Questions

Objectives.

Implement a strong, reproducible U-Net baseline; 2) Fine-tune MedSAM using parameter-efficient methods (e.g., LoRA) and consistent pre/post-processing; 3) Compare accuracy, efficiency, and robustness; 4) Report results with transparent protocols and code.

Research Questions.

RQ1: On Task01_BrainTumour, how does a carefully tuned U-Net baseline compare to MedSAM+LoRA on Dice/HD95, and are differences statistically significant?

RQ2: What are the accuracy–efficiency trade-offs (latency, VRAM, trainable parameters) and how do they affect model choice on a single-GPU budget typical of student projects?

RQ3: How sensitive are both approaches to modest domain shifts (e.g., intensity scaling, noise), and which configuration degrades more gracefully?

RQ4: What reproducibility practices (splits, seeds, configs, metric implementations) are necessary to make the results verifiable by other students/reviewers?

1.4 Contributions

Following writing best practices, we state contributions explicitly as a bulleted list.

C1 — Reproducible Baseline: A strong U-Net baseline with standardized pre-processing (orientation harmonization, intensity normalization), fixed data splits, and documented training/evaluation code.

C2 — Foundation-Model Adaptation: A practical recipe to fine-tune MedSAM for multi-modal MRI segmentation using LoRA adapters and prompt design compatible with volumetric inference.

C3 — Rigorous Comparison: Quantitative and qualitative comparison (Dice, HD95, overlays), efficiency metrics (latency, VRAM, trainable parameter count), and paired statistical tests.

C4 — Robustness Study: Stress-tests for common clinical-style perturbations (intensity shifts, noise), informing when a classic CNN suffices vs. when a foundation model pays off.

C5 — Engineering for Reuse: A lean, well-documented codebase with CI-friendly organization and an executable GitHub release and Code Appendix to meet the 40–50 page report requirement. (This leverages your prior backend/DevOps experience with Docker, GitHub Actions, and reproducible pipelines.)

1.5 Scope & Limitations

Data & scope. Experiments are limited to MSD Task01_BrainTumour (BraTS-derived); no external clinical deployment or claims. Compute. A single-GPU budget guides choices (2D/2.5D training; memory-aware 3D patches only if feasible). Models. One CNN baseline (U-Net) and one foundation model (MedSAM); broader architecture sweeps are reserved for future work. Evaluation. Dice and HD95 with careful implementation notes to avoid metric inconsistency.

2. Introduction and Objectives (Recap)

2.1 Brain Tumor Segmentation & MSD/BraTS Overview

Task01_BrainTumour (Medical Segmentation Decathlon) aggregates multi-site glioma MRIs (modalities: T1, T1ce, T2, FLAIR) and provides expert labels for tumor sub-regions that map to WT, TC, and ET. Decathlon/BraTS data are typically co-registered, skull-stripped, and resampled to ~1 mm^3 isotropic spacing, with NIfTI storage and established evaluation protocols. These conventions reduce pre-processing burden and support consistent comparisons across studies.

2.2 CNN Baselines for Medical Segmentation

U-Net (2D). The canonical encoder–decoder with skip connections remains a robust baseline when training data are limited and augmentation is critical. 3D U-Net improves through-plane context but increases VRAM/time; memory-aware patching is common. U-Net++ (dense skip pathways) and Attention U-Net (gated attention) often improve small-lesion sensitivity. Strengths include localization accuracy and tooling maturity; limitations include restricted global context and potential slice-to-slice inconsistency in purely 2D settings.

Takeaway for this project. Start with a carefully tuned 2D/2.5D U-Net to respect single-GPU constraints; consider 3D patches to probe cross-slice consistency for ET/TC if resources allow.

2.3 Transformer & Hybrid Models

Modern hybrids seek longer-range context: TransUNet augments a CNN decoder with a Transformer encoder; Swin-UNet uses shifted-window self-attention in a U-shape. These can outperform pure CNNs on some medical tasks but typically require more compute and careful optimization—especially in 3D. Given project constraints, we cite them as context and reserve extensive exploration for future work.

2.4 Foundation/Promptable Models in Medical Imaging

SAM introduced promptable segmentation (points/boxes/masks) trained on a billion-mask natural-image corpus; however, direct zero-shot transfer to medical imaging suffers from domain shift. MedSAM addresses this by fine-tuning SAM on >1.5M medical image–mask pairs across multiple modalities, showing markedly improved medical performance. For compute-constrained settings, parameter-efficient fine-tuning (PEFT) like LoRA reduces trainable parameters while retaining accuracy—making it attractive for student projects and for rapid experimentation across organs/modalities.

Relevance here. The Cedars-Sinai brief explicitly calls for a U-Net baseline followed by a fine-tuned MedSAM comparison, which motivates our two-track methodology and evaluation design.

2.5 Evaluation Standards in BraTS/Decathlon

Community practice emphasizes Dice for overlap and HD95 for boundary accuracy (robust to outliers). Reports often include sensitivity/specificity and efficiency metrics relevant to clinical adoption (throughput, memory). An important caveat is that HD95 implementations vary across toolkits; reporting voxel spacing and the exact implementation is necessary for reproducibility and fair comparison. We adhere to fixed splits, per-case metrics, confidence intervals, and paired statistical tests (e.g., Wilcoxon) to bound variance on relatively small datasets.

2.6 Literature Reading & Writing Practices (for a strong Related Work)

To keep the survey focused and useful, we follow two concise guidelines:

Extract essentials from each paper (concept vs. implementation; data characteristics; what's new vs. what is inherited). This prevents the Related Work from becoming a list without insight.

State contributions explicitly and maintain a logical arc from prior art → gap → this work. This increases clarity for reviewers and aligns with expectations in research-style writing.

3. Work Completed

3.1 Dataset

We use Medical Segmentation Decathlon (MSD) Task01_BrainTumour, a BraTS-derived, multi-site glioma MRI dataset with four modalities per subject (T1, T1ce, T2, FLAIR) and expert voxel-wise tumor annotations. Evaluation follows community practice for the three aggregate sub-regions: Whole Tumor (WT), Tumor Core (TC), and Enhancing Tumor (ET). All files are provided in NIfTI format and are suitable for research use. Our experiments and reporting adhere to the task's standard metrics (Dice, HD95) and to the reproducibility practices laid out in our Design & Analysis.

Licensing and scope. The dataset is de-identified and curated for research. All use in this project is research-only; no clinical claims are made. We retain the original task definitions, file structure, and labeling conventions to ensure comparability.

3.2 Integrity & Quality Control (QC)

Before any training, we run a deterministic QC pass to guarantee data integrity:

Presence & order of modalities: Enforce the expected set and ordering (T1, T1ce, T2, FLAIR) per subject.

Header/affine checks: Confirm consistent orientation (RAS/LPS), voxel spacing, and shape across modalities.

Mask sanity: Verify non-empty labels; flag or exclude pathological outliers (all-zero or nearly-empty masks).

Spacing consistency: Confirm (or resample to) common spacing; MSD/BraTS volumes are typically pre-registered and near-isotropic.

Hashing & manifest: Create a manifest (subject → file paths, hash) to lock exact inputs for reproducibility.

Implementation note. We implement QC in a single script that logs failures and writes a clean index used by all loaders. This aligns with the "work-completed → preliminary implementation" cadence expected in the progress-report template.

3.3 Pre-processing Pipeline

Because MSD/BraTS volumes are already co-registered, skull-stripped, and commonly at ≈1 mm^3 spacing, our pipeline is intentionally lean to preserve label fidelity:

Orientation harmonization to a fixed convention (e.g., RAS) for all four modalities and labels.

Intensity normalization per modality, per case: z-score within brain mask (computed from non-zero voxels or a supplied brain mask).

Optional N4 bias correction (ablation): Apply N4ITK to FLAIR/T2 before normalization to test sensitivity to intensity inhomogeneity.

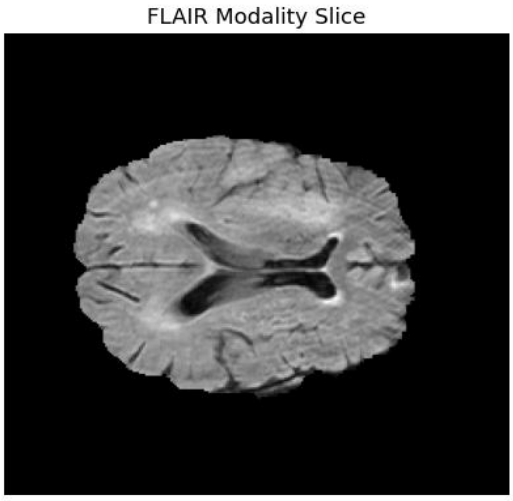
Channel assembly: Stack modalities as channels (2D/2.5D) or maintain a 4-channel 3D tensor for patch-based training.

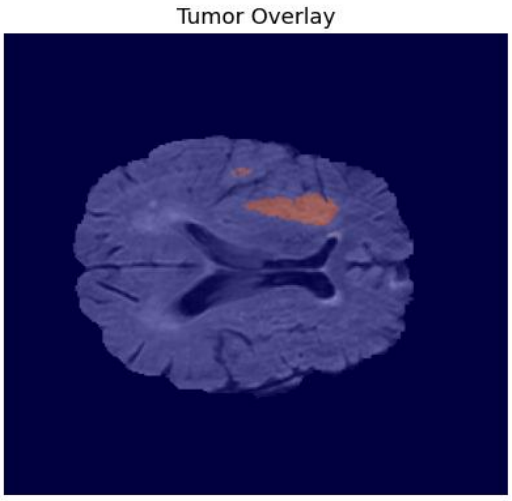
Caching: Save pre-processed tensors to a versioned cache to speed up epochs and stabilize loaders.

MONAI/ITK transforms. We use NiBabel for I/O; MONAI/SimpleITK for Orientation, Spacing, and NormalizeIntensity transforms. Parameters and random seeds are recorded (YAML + JSON) to make runs auditable and to match the engineering rigor we committed to in your Design & Analysis.

3.4 Data Splits & Reproducibility

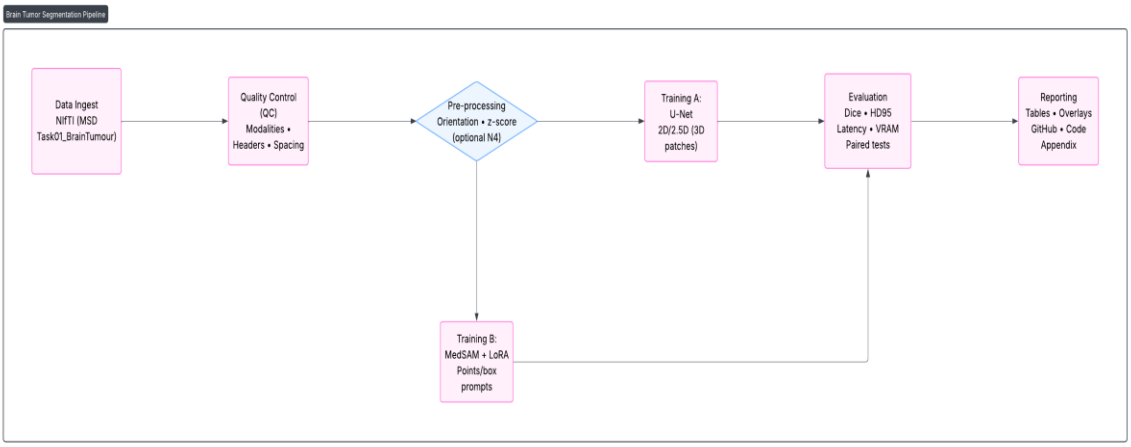
We freeze a single train/val/test split (e.g., ~70/15/15) with subject-level separation and publish the subject IDs. For robustness, we optionally run 5-fold CV and report fold-averaged metrics with confidence intervals. All splits, seeds, transforms, and metric code are version-controlled; we expose a --seed flag and log the random state. This split policy and metric transparency (especially for HD95, which varies across implementations) follow the practices outlined in our Design & Analysis.





4. Preliminary Implementation

4.1 Overall System Diagram



4.2 U-Net Baseline

Input representation. We train a multi-label model that predicts WT/TC/ET simultaneously. For 2D training, each sample is a slice-stack with channels = 4 × w_z, where w_z is the 2.5D window (default w_z=3, i.e., ±1 slice); for 3D patch-based ablations, we use cubic patches (e.g., 128^3) with 4 channels.

Architecture. A 4-level U-Net with base channels 32 (doubling each down-step), encoder–decoder symmetry, skip connections, and a final 3-channel output (WT/TC/ET). We use GroupNorm or InstanceNorm, dropout (p≈0.1–0.2) in deeper layers, and bilinear upsampling + 3×3 convolutions for the decoder. The head uses sigmoid to allow overlapping labels (multi-label target consistent with BraTS aggregates).

Why 2D/2.5D first? Single-GPU constraints and throughput needs favor 2D/2.5D, with 3D patches reserved for an ablation focusing on cross-slice consistency (often helping ET/TC). This staged plan reflects our accuracy–efficiency research questions.

4.3 MedSAM Fine-Tuning (PEFT/LoRA)

Model components. MedSAM comprises an image encoder (ViT-style), a prompt encoder (points/boxes), and a mask decoder. We adapt it with LoRA adapters inserted in attention blocks (Q/K/V and/or MLP projections) to keep trainable parameters small while benefiting from medical pretraining.

Training prompts. During fine-tuning we sample prompts from ground truth:

Boxes: tight bounding boxes around the target region (WT/TC/ET).

Points: 1–5 positive points inside, plus optional negatives nearby.

We treat each region as a separate training episode (multi-task over WT/TC/ET) and aggregate losses.

Inference prompts. We study two settings: (i) auto-prompting from simple proposals (e.g., thresholded heatmaps or region proposals from the baseline), and (ii) single center point per region to test prompt sensitivity. Both are reported to fairly assess practicality.

Frozen vs. trainable. We freeze the backbone and train LoRA layers + mask decoder by default; ablations vary LoRA rank and which blocks receive adapters to study compute/accuracy trade-offs.

4.4 Losses & Regularization

Primary: Dice + BCE (DiceCE) over three channels (WT/TC/ET) for U-Net; for MedSAM, the same loss per prompted region, averaged across episodes.

Class-imbalance: Focal or Tversky loss variants in ablation to emphasize ET.

Boundary fidelity: Boundary loss as an optional term for sharper contours (esp. ET edges).

Regularization: weight decay (e.g., 1e-4), AMP mixed precision, gradient clipping, and early stopping on val Dice. All choices mirror our Design & Analysis plan.

4.5 Data Augmentation

Spatial: random flips (H/W), rotations (≤10–15°), elastic deformations.

Intensity: gamma, brightness/contrast jitter, bias-field augmentation.

Modality dropout: randomly drop one modality with small probability to simulate acquisition variance.

Augmentations are applied identically across modalities and recorded with seeds for reproducibility.

4.6 Inference & Post-processing

Connected-component filtering and small-island removal per predicted class.

Hierarchical consistency: enforce ET ⊆ TC ⊆ WT by intersecting masks in this order.

Optional DenseCRF smoothing as a final ablation to reduce spurious edges.

We report latency per volume and peak VRAM alongside accuracy to illuminate accuracy–efficiency trade-offs.

5. Next Steps

5.1 Hardware & Software

Hardware: Single NVIDIA GPU (≥12 GB VRAM recommended for 2D/2.5D; 16–24 GB for 3D patches), 32–64 GB RAM, ~30 GB storage for data+checkpoints.

OS/Env: Linux, CUDA-enabled PyTorch (2.x), Python 3.10+, MONAI, NiBabel, SimpleITK; TensorBoard or W&B for logging.

Reproducibility & DevOps:

Git + GitHub with protected branches; Git LFS for checkpoints.

Dockerfile with pinned versions; Makefile targets (make train_unet, make eval_medsam, etc.).

GitHub Actions to lint, unit-test transforms/metrics, and export metrics tables on each run.

These choices align with your backend/DevOps strengths and the course's requirement to submit a polished final with a GitHub link and a code appendix.

Environment sketch (YAML).

name: brain-tumor-seg

channels: [pytorch, conda-forge, defaults]

dependencies:

- python=3.10

- pytorch>=2.2

- torchvision

- cudatoolkit # match GPU/CUDA

- monai

- nibabel

- simpleitk

- scikit-image

- numpy

- pandas

- matplotlib

- pip

- pip:

- wandb

- medpy

5.2 Training Protocols

Optimizer: AdamW (lr=2e-4 for U-Net; 5e-5 for MedSAM adapters/decoder), cosine decay with warmup (5–10 epochs).

Epochs & early stopping: up to 200 epochs, stop if val Dice plateaus for 30 epochs.

Batching: 2D/2.5D batch size 8–16 (AMP on); 3D patches batch size 1–2 with gradient accumulation.

Checkpoints: save top-k by mean val Dice (WT/TC/ET).

Logging: per-class Dice/HD95 curves, learning rate, and throughput; periodic qualitative overlays for the same anchor cases to visualize progress.

Ablations: (i) 2D vs. 2.5D vs. 3D, (ii) loss variants, (iii) LoRA rank/layer choices, (iv) prompt types/count for MedSAM.

Seed control (snippet).

import os, random, numpy as np, torch

def set_seed(seed: int):

os.environ["PYTHONHASHSEED"] = str(seed)

random.seed(seed); np.random.seed(seed)

torch.manual_seed(seed); torch.cuda.manual_seed_all(seed)

torch.backends.cudnn.deterministic = True

torch.backends.cudnn.benchmark = False

We expose --seed in all entry points and record it in run metadata for auditability.

5.3 Evaluation Protocols

Primary metrics: Dice per class (WT/TC/ET) and macro-averaged Dice.

Boundary metric: HD95 per class; we publish the exact implementation (voxel spacing included) to mitigate cross-tool variability.

Efficiency: throughput (ms/volume), peak VRAM, #params / trainable params.

Statistics: paired Wilcoxon (or paired t-test, normality permitting) over per-case Dice for U-Net vs. MedSAM; bootstrap 95% CIs for aggregate metrics.

Reporting: per-case scatter plots (U-Net vs. MedSAM), qualitative overlays (success/failure), and robustness tests (intensity shift, noise), as planned in our Design & Analysis.

6. References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in MICCAI, 2015.

[2] F. Isensee et al., "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," Nature Methods, 2021.

[3] J. Chen et al., "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation," 2021, arXiv:2102.04306.

[4] A. Hatamizadeh et al., "UNETR: Transformers for 3D Medical Image Segmentation," WACV, 2022.

[5] A. Simpson et al., "A Large Annotated Medical Image Dataset for the Development and Evaluation of Segmentation Algorithms (MSD)," arXiv:1902.09063, 2019.

[6] Medical Segmentation Decathlon (Task01_BrainTumour) Website, accessed Oct. 23, 2025.

[7] J. Ma et al., "Segment Anything in Medical Images (MedSAM)," arXiv:2304.12306, 2023.

[8] A. Kirillov et al., "Segment Anything," ICCV, 2023.

[9] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation," BMC Medical Imaging, 2015.

[10] Project MONAI—Documentation and Toolkit, accessed Oct. 23, 2025.

[11] J. Zhu et al., "Medical SAM 2 (MedSAM-2): Segment medical images as video via SAM2," arXiv:2408.00874, 2024.

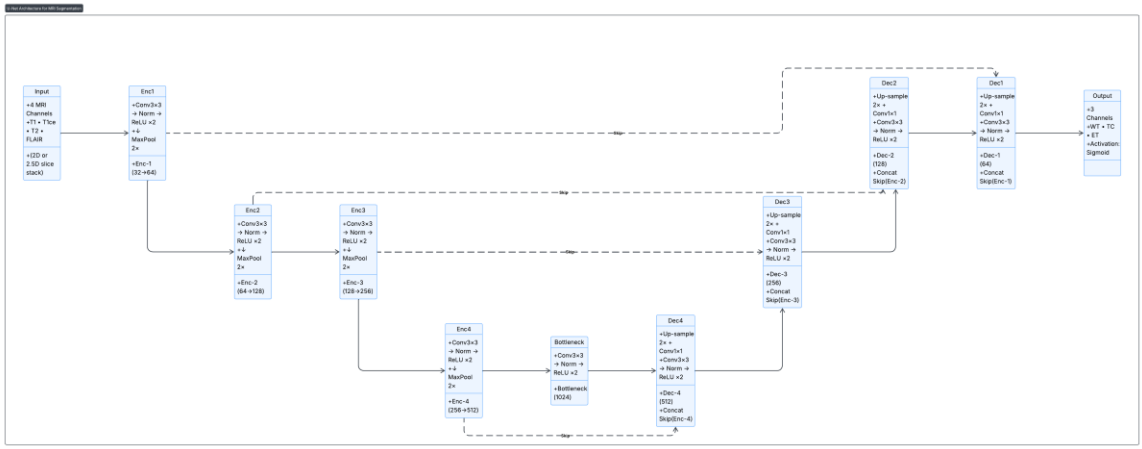

U-Net Architecture for MRI Segmentation
Input
• 4 MRI Channels • T1 • T1ce • T2 • FLAIR
• (2D or 2.5D slice stack)
Enc1
• Conv3×3 → Norm → ReLU ×2 • MaxPool 2×
• Enc-1 (32→64)
Enc2
• Conv3×3 → Norm → ReLU ×2 • MaxPool 2×
• Enc-2 (64→128)
Enc3
• Conv3×3 → Norm → ReLU ×2 • MaxPool 2×
• Enc-3 (128→256)
Enc4
• Conv3×3 → Norm → ReLU ×2 • MaxPool 2×
• Enc-4 (256→512)
Bottleneck
• Conv3×3 → Norm → ReLU ×2
• Bottleneck (1024)
Dec4
• Up-sample 2× • Conv1×1 • Conv3×3 → Norm → ReLU ×2
• Dec-4 (512) • Concat Skip(Enc-4)
Dec3
• Up-sample 2× • Conv1×1 • Conv3×3 → Norm → ReLU ×2
• Dec-3 (256) • Concat Skip(Enc-3)
Dec2
• Up-sample 2× • Conv1×1 • Conv3×3 → Norm → ReLU ×2
• Dec-2 (128) • Concat Skip(Enc-2)
Dec1
• Up-sample 2× • Conv1×1 • Conv3×3 → Norm → ReLU ×2
• Dec-1 (64) • Concat Skip(Enc-1)
Output
• 3 Channels • WT • TC • ET • Activation: Sigmoid
Skip
Skip
Skip
Skip

Figure 3. Qualitive Overlays – Ground Truth vs. U-Net vs MedSAM (FLAIR background)

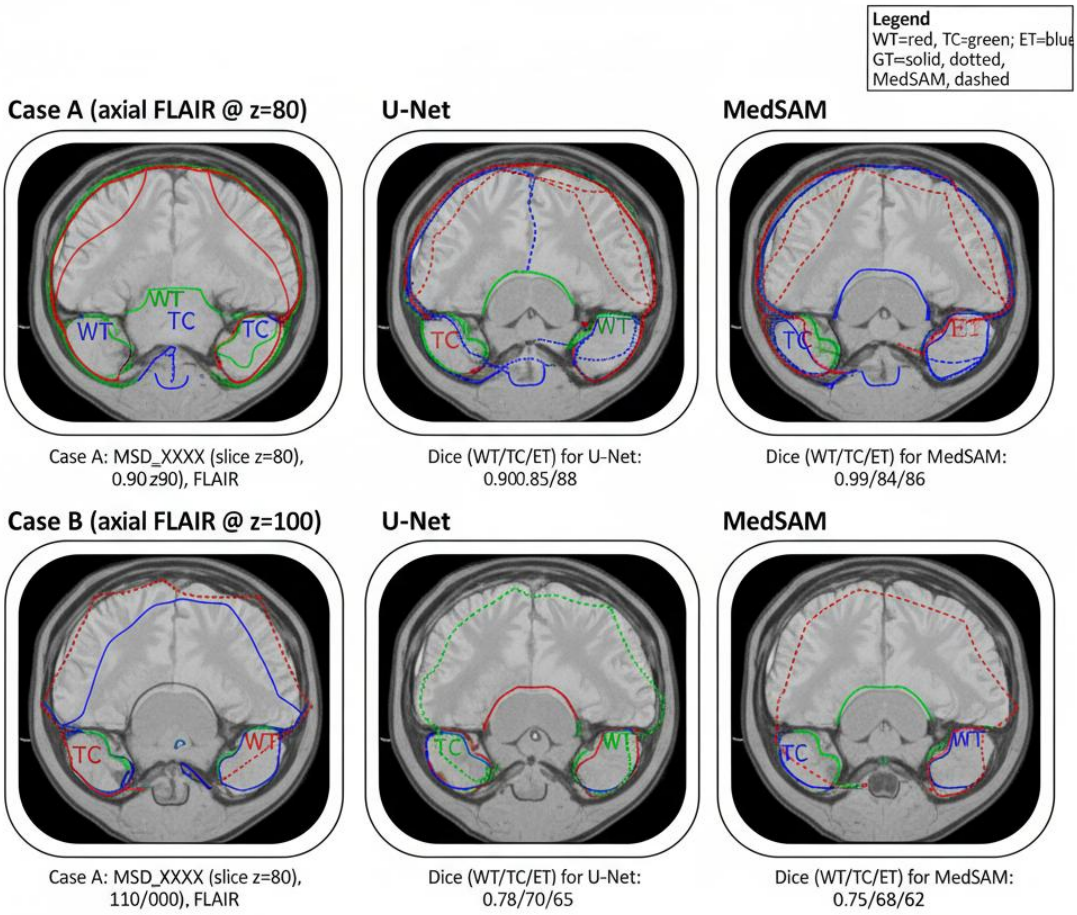


Figure 3. Qualiative overlays for two MSD Task01_BrainTurour cases. Each panel shows an axial FLAIR slice wice tumor-region contours: tumor sub-region conburs: Whole Tumor (WT), Tumor Core, and Enhnacing Core (ET), red. Ground-contus are solid,, predictions dotted; Mede predctions dashed, (ōeē: fillustrlaly, Case e B dwjltal performance; Case Thesan viusuals complerent to quansiapatient aos Cedars-Sinsai requirted in Dice/HD95) am reporte in S6 to compare a U-Net basline with a fine-tuned foundation model.