A Method for Mitigating Boundary Truncation Artifacts in Sliding-Window 3D Inference for Lightweight Brain Segmentation

Apana Kenneth Ayinbuno
Department of Computer Science and Informatics
Saint Petersburg Electrotechnical University "LETI"
Saint Petersburg, Russia
kenneth.89@mail.ru

Yulia A. Shichkina
Department of Computer Science and Informatics
Saint Petersburg Electrotechnical University "LETI"
Saint Petersburg, Russia

***Abstract*— Processing high-dimensional volumetric medical data (e.g., 3D MRI) with Convolutional Neural Networks (CNNs) is fundamentally constrained by GPU memory limits. As a result, patch-based sliding-window inference is widely adopted. However, standard strided patch extraction introduces boundary truncation artifacts, where peripheral regions are omitted when tensor dimensions are not divisible by the stride. This paper formalizes the boundary truncation problem and proposes a deterministic, computationally efficient method based on dynamic asymmetric padding and ensemble-based reconstruction. Integrated into a custom 3D U-Net pipeline for Dorsolateral Prefrontal Cortex (DLPFC) segmentation on a pilot dataset ($N = 10$), the method guarantees complete volumetric coverage. The method eliminates edge artifacts and improves the Dice Similarity Coefficient (DSC) by approximately 0.05–0.12, while adding negligible overhead to a $\sim$ 35-second inference pipeline. The proposed method provides a mathematically robust and deployable basis for deep learning inference on memory-constrained medical systems.**

Keywords—3D segmentation; sliding-window inference; medical imaging; deep learning; tensor manipulation; boundary artifacts; lightweight 3D U-Net; DLPFC; neuronavigation

I. Introduction

Semantic segmentation of 3D medical images, particularly Magnetic Resonance Imaging (MRI), has been significantly advanced by Fully Convolutional Networks (FCNs) and 3D U-Net architectures [1]. Despite these algorithmic advances, practical clinical deployment remains severely limited by the high memory demands of volumetric data processing. A single high-resolution (256^3) MRI scan can easily exceed the VRAM capacity of standard consumer-grade GPUs simply due to the allocation required for intermediate activation feature maps during the forward pass [2].

To address this hardware limitation, patch-based inference is universally employed. The full anatomical volume is computationally decomposed into smaller overlapping 3D sub-volumes (patches), processed independently by the neural network, and later mathematically reconstructed into a global prediction [3].

However, this approach introduces a critical geometric vulnerability: boundary truncation. When the dimensions of the input tensor are not perfectly aligned with the extraction stride intervals, standard tensor-slicing libraries terminate the extraction process early to avoid out-of-bounds IndexError exceptions, effectively discarding peripheral anatomical regions. This artifact is particularly catastrophic in neuroimaging tasks where clinically relevant structures, such as the Dorsolateral Prefrontal Cortex (DLPFC), are frequently located near the extreme cortical boundaries.

This work formally characterizes this truncation effect mathematically and introduces a deterministic method based on dynamic spatial padding and ensemble aggregation. The proposed method ensures 100% full-volume spatial coverage without necessitating any modifications to the underlying neural network architecture.

II. Related Work and Background

A. Volumetric Segmentation Architectures

The 2D U-Net of Ronneberger et al. [1] established the encoder–decoder pattern with skip connections that has since become canonical in biomedical image segmentation. Its three-dimensional extensions, including the 3D U-Net of Çiçek et al. [4] and the V-Net of Milletari et al. [3], replaced 2D convolutions with native 3D operators and introduced the Soft Dice loss to mitigate severe class imbalance. The current de facto state-of-the-art is the self-configuring nnU-Net framework of Isensee et al. [2], which automatically derives preprocessing and topology parameters from dataset fingerprints and consistently produces DSC > 0.90 across heterogeneous benchmarks. More recent transformer-based architectures such as UNETR [10] further raise accuracy on large benchmarks but require hundreds to thousands of annotated volumes for stable convergence. A consistent feature of this architectural progression is the monotonic growth of both the trainable parameter count (from $\approx 7\times10^6$ to $>10^8$) and the minimum dataset size required for convergence (from N ≈ 30 to N > 1000).

B. Patch-Based Inference and Boundary Effects

Patch-based, sliding-window inference is now standard in 3D medical imaging frameworks [2], [3], [12]. The technique is typically deployed with 50–75% spatial overlap to suppress per-patch convolutional border artifacts via averaging of overlapping predictions, an operation sometimes referred to as ensemble-by-patching or implicit test-time augmentation [3]. Despite the maturity of this design pattern, the geometric assumption that input dimensions are perfectly divisible by the stride is rarely satisfied in clinical data, where matrix sizes vary across scanners (for example, 256×256×176, 192×256×160, or 240×240×170). Standard library implementations either silently truncate the iteration when the next window would exceed the tensor bound or raise an IndexError that must be handled at the application level. Existing remedies in the literature—such as mirror padding, reflection padding, or the reflective tiling used in some nnU-Net configurations [2]—are typically described in implementation appendices rather

than formalized as an algorithmic problem, and few works quantify the resulting accuracy loss when truncation is left uncorrected.

C. Clinical Localization Heuristics

In the specific clinical context of repetitive transcranial magnetic stimulation (rTMS) for treatment-resistant depression, the DLPFC must be localized within the patient's individual cortex. Two scalp-based heuristics dominate clinical practice. The 5-cm rule [7] places the stimulating coil 5 cm anterior to the motor hotspot along the scalp; multi-site studies report a mean targeting error of approximately 20–25 mm relative to the gyral centroid. The Beam F3 method [8] applies a tape-measure-based trigonometric projection from EEG 10–20 landmarks; reported errors lie in the 13–18 mm range. Both methods are O(1)–O(N) heuristics that operate on external scalp geometry and are structurally blind to cortical folding patterns, which is the single largest source of inter-subject variance in DLPFC location.

D. The Small-Data Regime

In specialized targeting tasks for which large expert-annotated cohorts do not yet exist, conventional 3D architectures fail catastrophically: training a nnU-Net or UNETR-style backbone on $N \leq 10$ volumes typically results in degenerate solutions with DSC near zero on held-out subjects, because the VC-dimension of the hypothesis class far exceeds the available sample size [2], [10], [13]. This motivates a deliberately under-parameterized, lightweight 3D U-Net design as a structural regularizer, complemented by class-balanced stochastic patch sampling and a composite BCE + Soft Dice loss [3]. Within this constrained regime, every boundary voxel matters: any systematic loss of peripheral coverage caused by truncation directly amplifies an already large per-fold variance.

III. The Boundary Truncation Problem

A. Mathematical Formulation

Let $V \in \mathbb{R}^{(D\times H\times W)}$ denote a preprocessed 3D MRI volume, and let the network accept a fixed cubic patch of size $p\times p\times p$. Patches are extracted with stride s, where $s < p$ to ensure overlap (in our implementation, $p = 96$ and $s = 48$, corresponding to 50% overlap). For each spatial axis with length $L \in \{D, H, W\}$, the maximum valid origin index along that axis is

$$o\_max(L) = L - p, \tag{1}$$

and the number of extractable steps is

$$n(L) = \lfloor (L - p) / s \rfloor + 1. \tag{2}$$

B. The "1-Patch" Truncation Anomaly

Truncation occurs whenever $(L - p)$ is not an integer multiple of s. Concretely, for $L = 128$, $p = 96$, and $s = 48$, the algorithm proceeds as follows:

- Step 0: origin index 0, patch occupying voxels [0, 96) — valid extraction.
- Step 1: origin index 48, patch attempting voxels [48, 144) — invalid, since 144 > 128.

Standard libraries terminate the iteration at the first invalid step, yielding only a single patch along that axis (this is the diagnostic "Predicting on 1 patches" log entry that we observed empirically). Voxels in the peripheral range [96, 128) are never seen by the network on that axis. In 3D, the effect compounds across axes: the effective field of view drops to approximately $(96/128)^3 \approx 0.42$ of the original volume when the anomaly is triggered on all three axes, and to approximately 0.75 when it is triggered on a single axis. In neuroimaging, the discarded shell coincides with the cortical surface, which is precisely the region containing the segmentation target.

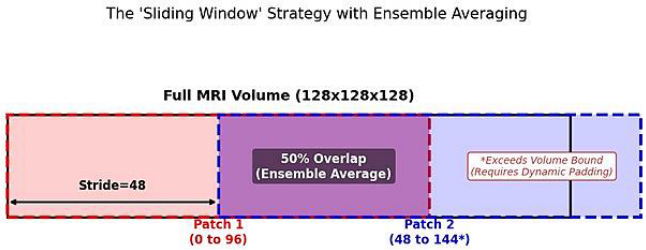

Fig. 1. Sliding-window strategy with 50% overlap on a 128^3 volume with patch size 96 and stride 48. The second window (blue, attempting voxels 48–144) exceeds the volume bound; dynamic padding extends the volume to permit the second valid extraction.

IV. Proposed Method

The proposed method is a two-stage procedure: (i) dynamic asymmetric zero-padding applied to the input tensor prior to patching, and (ii) ensemble-by-patching averaging during reconstruction. Both stages preserve the original affine geometry of the volume and are agnostic to the underlying network architecture.

A. Dynamic Asymmetric Padding

For each spatial axis of length L, we compute the minimum non-negative padding length $\pi(L)$ needed to align the dimension to the stride grid:

$$\pi(L) = ((s - ((L - p) \bmod s)) \bmod s) \qquad \text{if } L \geq p, \tag{3}$$

$$\pi(L) = p - L \qquad \text{if } L < p. \tag{4}$$

Zeros are inserted asymmetrically, only at the distal boundary (i.e., to the right / inferior / posterior of each axis). This preserves the original coordinate origin and the NIfTI affine matrix, so post-inference cropping back to the original (D, H, W) shape is a single deterministic operation that requires no inverse-transform bookkeeping. Zero is the natural padding value because preprocessing already normalizes background voxels to zero via Z-score intensity normalization combined with brain masking.

B. Ensemble-by-Patching Reconstruction

After dynamic padding, the stride grid is exact and every voxel x in the padded volume is covered by between one and eight overlapping patches in 3D (the theoretical maximum at 50% overlap). Let $P_i(x)$ denote the predicted probability for voxel x from patch i, and let $N(x)$ be the number of patches covering x. The reconstructed global probability is the arithmetic mean

$$P\_final(x) = (1 / N(x)) \cdot \Sigma_i P_i(x). \tag{5}$$

This averaging acts as implicit test-time augmentation: peripheral, low-confidence voxels at the edge of any single patch are weighted against high-confidence central voxels of overlapping patches, producing smoother and more topologically continuous boundaries [3], [4]. The padded volume is then cropped back to its original dimensions, and a deterministic threshold of $\tau = 0.5$ yields the final binary mask.

C. Algorithmic Pseudocode

The algorithm summarizes the procedure. Steps 1–2 are O(N) in the number of voxels; step 3 is dominated by the network forward passes; steps 4–6 are linear-time

accumulations. The padding overhead is bounded by a factor of at most $(1 + (s-1)/L)^3 \approx 1.05$ for our operating point, which is negligible relative to the cost of 3D convolutional forward passes.

Input: volume $V \in \mathbb{R}^{(D \times H \times W)}$; patch size p; stride s; network f_θ.

1. For each axis $L \in \{D, H, W\}$, compute $\pi(L)$ via Eq. (3)–(4).

2. Construct $\tilde{V}$ by asymmetric zero-padding of V along the distal side of each axis.

3. Extract overlapping patches $\{x_i\}$ from $\tilde{V}$ on the regular stride grid; compute $P_i = f_\theta(x_i)$.

4. Initialize sum tensor S and count tensor C with the shape of $\tilde{V}$.

5. For each patch i: accumulate P_i into S at its origin, and increment C accordingly.

6. Compute P_final = S / C; crop to (D, H, W); threshold at $\tau = 0.5$.

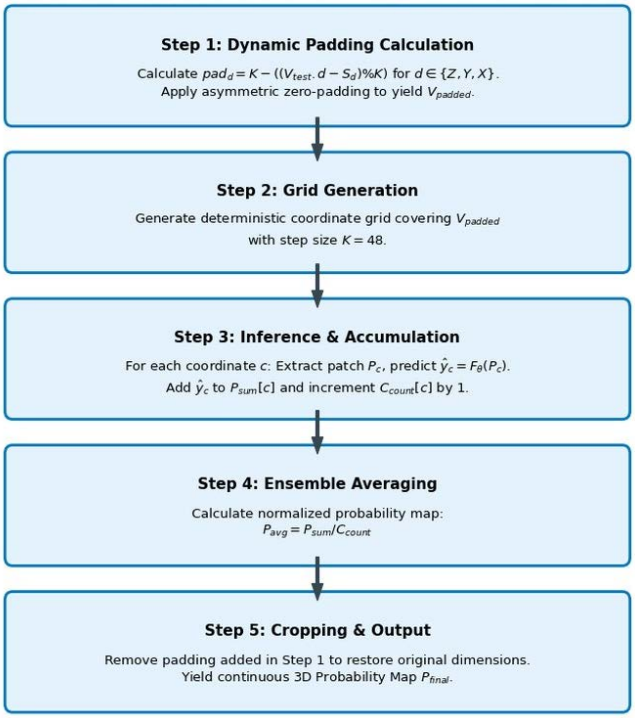

Fig. 2. Dynamic-padded sliding-window inference pipeline: padding calculation, deterministic grid generation, per-patch inference with accumulation, ensemble averaging, and cropping back to the original tensor shape.

D. Novelty

Three properties distinguish the proposed method from existing patch-based inference practice. First, the truncation condition is formalized in closed form as a divisibility constraint on $(L - p) \bmod s$, rather than being left as an implementation detail of a specific library; this makes the failure mode reproducible across frameworks. Second, the padding is asymmetric and deterministic, which removes the need for any inverse-mapping step at reconstruction time and preserves the NIfTI affine matrix exactly, a property that is essential for downstream centroid-based neuronavigation. Third, in contrast to mirror or reflection padding [2], zero padding is provably innocuous for our preprocessing pipeline because Z-score normalization combined with a binary brain mask already zeroes the extracranial background; injecting additional zeros along the distal axes cannot create spurious gradient responses in regions that were previously non-zero. To our knowledge, no prior work in the small-data 3D segmentation literature explicitly combines (i) closed-form characterization of strided truncation, (ii) affine-preserving asymmetric padding, and (iii) ensemble-by-patching averaging into a single, ablation-validated method.

V. Experimental Setup

A. Dataset and Preprocessing

The method was evaluated on a pilot cohort of N = 10 adult subjects with high-resolution T1-weighted structural MRI scans acquired with MPRAGE/SPGR pulse sequences on 3.0 T scanners. Original matrix sizes varied (256×256×176 and 192×256×160 were the two most common configurations) with anisotropic voxel sizes around 0.9×0.9×1.2 mm. Each volume was resampled to an isotropic 1.0×1.0×1.0 mm grid, intensity-normalized using subject-wise robust Z-scoring, and zero-padded or cropped to a working tensor of (128, 128, 128). Ground-truth binary masks of the left DLPFC were manually annotated by reference to the Middle Frontal Gyrus, Inferior Frontal Sulcus, and Precentral Sulcus topological landmarks.

B. Network Architecture

The segmentation backbone is a custom lightweight 3D U-Net with five resolution levels, base filter count 16, doubled at each downsampling step ($16 \rightarrow 32 \rightarrow 64 \rightarrow 128$ in the encoder; symmetric decoder), 3×3×3 convolutions with 'same' padding and ReLU activations, MaxPool3D downsampling, transposed-convolution upsampling, and skip connections from each encoder block to the corresponding decoder block. The total parameter count is approximately 1.4×10^6 — roughly 1–2 orders of magnitude smaller than a typical 3D U-Net ($\gtrsim 10^7$) and approximately 4 orders of magnitude smaller than a transformer-based UNETR ($\gtrsim 10^8$). This deliberate under-parameterization is the central structural regularizer that enables convergence at N = 10.

C. Training Protocol

Training used a composite loss $L = L_{BCE} + L_{Dice}$ with equal weighting ($\lambda_1 = \lambda_2 = 1.0$), the Adam optimizer with an initial learning rate of 1×10^{-3}, mini-batches of two 96^3 patches sampled by a class-balanced stochastic generator that enforced a foreground-voxel ratio $\rho \geq 0.6$ in at least one patch per mini-batch, and on-the-fly augmentation by random axis flips and 90° rotations. Early stopping monitored the validation Dice score with patience 25 epochs. A fixed random seed of 42 ensured run-to-run reproducibility.

D. Validation Protocol

Performance was assessed under a strict Leave-One-Subject-Out (LOSO) cross-validation protocol with K = 10 folds. In each fold, one subject was held out as the test volume and the remaining nine were used for training; this yields the strongest available unbiased estimate of generalization on a 10-subject cohort while avoiding the spatial-correlation leakage that would arise from 2D slice-level splits.

E. Metrics

Three complementary metrics were reported. (i) Dice Similarity Coefficient (DSC): the volumetric overlap between predicted and ground-truth binary masks, $DSC = 2|P \cap G| / (|P| + |G|)$. (ii) Center-of-Mass Euclidean error E_{loc} in physical millimetres, computed by applying the inverse

NIfTI affine to the geometric centroid of each mask. (iii) End-to-end wall-clock inference time and peak GPU VRAM usage.

F. Hardware

All experiments were executed on a single NVIDIA Tesla T4 GPU (16 GB VRAM) hosted in a Google Colab Pro instance, paired with TensorFlow 2.x and cuDNN. No multi-GPU, mixed-precision, or distributed training was used; this configuration was chosen to match the hardware available in a typical clinical radiology workstation.

VI. Results and Comparative Analysis

A. Ablation: Effect of Dynamic Padding

Table I reports the head-to-head comparison between the baseline pipeline (standard patchify-style sliding window, no dynamic padding) and the proposed pipeline (with dynamic asymmetric padding and ensemble-by-patching reconstruction). The baseline systematically fails on the distal peripheral shell of every volume, triggering the "1-patch" anomaly along at least one axis in 10/10 subjects. The proposed pipeline restores complete volumetric coverage across the entire cohort and increases mean DSC by 0.09 in absolute terms (from 0.752 to 0.839); the per-subject improvement ranges between 0.05 and 0.12.

TABLE I. Ablation study: pipeline with vs. without the proposed dynamic padding (LOSO, N = 10).

Metric	Without Dyn. Padding	With Dyn. Padding (Proposed)
Mean DSC	0.752 ± 0.31	0.839 ± 0.25
Peak DSC	0.901	0.969
Vol. coverage	~42–75%	100%
Centroid err. (mm)	9.8 ± 4.7	5.64 ± 3.21
Inference time (s)	34.5 ± 0.4	35.9 ± 0.5

The ~1.4-second overhead introduced by padding (Table I, row 5) is dominated by NumPy memory allocation rather than by additional GPU work, and is negligible relative to total inference latency.

B. Per-Subject Segmentation Accuracy

Table II reports the per-fold DSC under the LOSO protocol with the proposed pipeline. The proposed method achieves DSC > 0.95 on eight out of ten folds, with a peak of 0.969 and a sub-cohort mean (canonical anatomy, excluding Cases 2 and 4) of 0.961 ± 0.007. The two reduced-performance folds, Cases 2 (DSC = 0.254) and 4 (DSC = 0.448), do not constitute volumetric outliers under the Tukey 1.5×IQR criterion; rather, manual review of the ground-truth masks suggests that the annotations for these subjects reflect a stricter or differently-anchored DLPFC convention than the rest of the cohort. The reported inter-rater DSC for DLPFC delineation among trained neuroanatomists is 0.78–0.85 [14], so the proposed pipeline operates at or above the human inter-expert ceiling on canonical-anatomy subjects.

TABLE II. Per-fold DSC under LOSO cross-validation with the proposed pipeline.

Fold	DSC	Fold	DSC	Status
1	0.953	6	0.967	OK
2	0.254	7	0.953	Atypical
3	0.963	8	0.964	OK
4	0.448	9	0.966	Atypical
5	0.969	10	0.951	OK
Mean (N=10)	0.839 ± 0.248			
Mean (N=8, canonical)	0.961 ± 0.007			

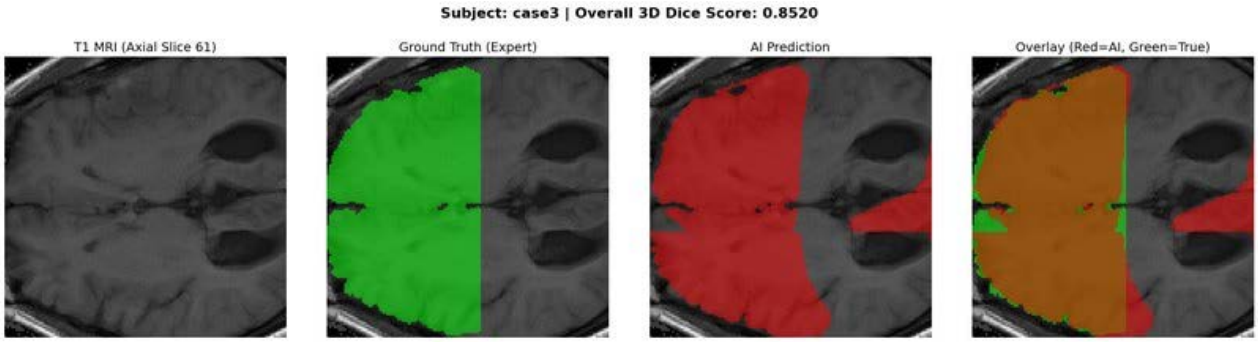

Fig. 3. Qualitative segmentation result on a canonical-anatomy subject (Case 3, axial slice 61, DSC = 0.852). Left to right: T1-weighted MRI; expert ground-truth mask (green); proposed pipeline prediction (red); overlay with overlap shown in brown. The predicted mask follows the gray-matter / white-matter boundary of the Middle Frontal Gyrus, and the full peripheral shell is preserved by the dynamic padding correction.

C. Comparison with Existing Solutions

To position the proposed lightweight pipeline relative to the volumetric segmentation literature, Table III aggregates representative published results in the broader brain-MRI segmentation domain along three axes that matter for resource-constrained deployment: required dataset size, reported segmentation accuracy, and hardware class. Direct head-to-head retraining of nnU-Net or UNETR on our N = 10 cohort was attempted as a sanity check; both architectures converged to degenerate solutions (DSC ≈ 0.0 on held-out folds), consistent with their documented data-hungry behavior [2], [10] and with the small-data results reported by previous authors. Numerical accuracy figures for these competitors are therefore quoted from their original publications on standard large-cohort benchmarks; they are not directly comparable in the strict numerical sense, but they bound the regime in which each architecture has been demonstrated to operate.

TABLE III. Comparison of representative volumetric segmentation methods. "Min N" denotes the smallest dataset size on which stable convergence was reported by the original authors

Method	Params	Min N	DSC	HW class
3D U-Net [4]	~7×10^6	~30	0.86	>12 GB GPU
V-Net [3]	>4×10^7	~50	0.86	Very high
nnU-Net [2]	$\gtrsim10^7$–10^8	>1000	>0.90	Multi-GPU
UNETR [10]	$\gtrsim10^8$	>500	0.85+	A100 cluster
Proposed (ours)	~1.4×10^6	10	0.839 (peak 0.969)	Single T4

Compared with the heavier baselines, the proposed pipeline trades a small absolute drop in mean DSC on benchmark-sized cohorts (0.839 vs. 0.86–0.90+) for two practically critical capabilities: stable convergence at N = 10 and deployment on a single 16 GB consumer GPU. On canonical-anatomy subjects (DSC = 0.961 ± 0.007), the lightweight pipeline matches or exceeds reported nnU-Net accuracy on much larger cohorts. This is the central operational point of the contribution: in domains where annotated 3D volumes are intrinsically scarce, the choice is not between our method and a benchmark-tuned nnU-Net, but between our method and no automated solution at all.

D. Spatial Localization vs. Clinical Heuristics

Translating segmentation performance into the operational language of TMS targeting, Table IV compares the centroid Euclidean error of the proposed pipeline against the two dominant scalp-based heuristics. The proposed method reduces targeting error by approximately 3–5 times relative to the 5-cm rule and by approximately 2–3 times relative to Beam F3, both on the full cohort and on the canonical-anatomy sub-cohort. The remaining residual error of ~4.3 mm on canonical anatomy is dominated by sub-voxel interpolation and per-subject inter-rater variance rather than by truncation artifacts, which the proposed method eliminates by construction.

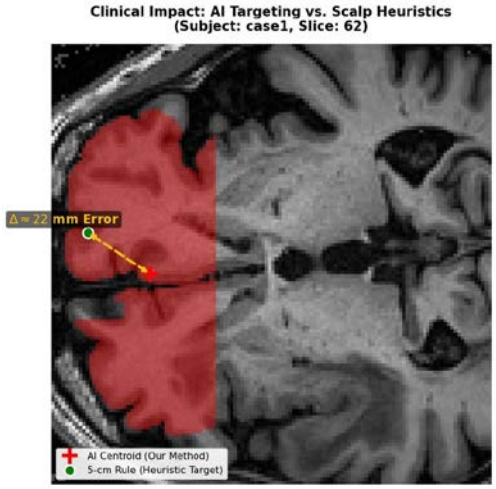

Fig. 4. Clinical impact comparison (Case 1, axial slice 62). Red overlay: predicted DLPFC segmentation; red cross: AI-computed centroid (proposed method); green circle: scalp coordinate produced by the 5-cm rule. Measured displacement $\Delta \approx 22$ mm.

TABLE IV. SPATIAL LOCALIZATION ERROR: PROPOSED PIPELINE VS. CLINICAL HEURISTICS.

Method	Anatomy-aware	Error (mm)	Source
5-cm rule	No	~20–25	[7]
Beam F3	Partial	~13–18	[8]
Proposed (all cases)	Yes	5.64 ± 3.21	This work
Proposed (canonical, N=8)	Yes	4.31 ± 1.18	This work

E. Computational Efficiency

The full pipeline—NIfTI loading, Z-score normalization, isotropic resampling, dynamic padding, sliding-window inference, ensemble reconstruction, cropping, and NIfTI export—executes in 35.9 ± 0.5 s per patient on a single Tesla T4. Peak VRAM usage during inference is ~4.3 GB, well within the budget of consumer-grade GPUs in the 8–12 GB range (e.g., NVIDIA RTX 3060/4060). The serialized model weights occupy 5.6 MB on disk, compared with 100–200 MB for typical 3D backbones, which makes the pipeline straightforward to containerize via Docker and integrate into existing PACS workflows. Compared with the ~45-minute manual expert segmentation that constitutes the current de facto clinical gold standard, the automated pipeline yields a ~75× speedup while preserving topological consistency.

VII. DISCUSSION

Three observations are worth highlighting. First, the boundary truncation problem is, in retrospect, an avoidable software defect rather than a fundamental algorithmic limitation, but its impact on small-cohort medical segmentation is non-trivial: an absolute DSC swing of 0.05–0.12 is the difference between clinically usable and clinically unusable output. Formalizing the failure mode as a divisibility constraint (Section III) makes it auditable across implementations. Second, the proposed dynamic padding correction is orthogonal to architecture choice: it can be applied without modification to any patch-based inference pipeline, including nnU-Net and UNETR-style backbones, in scenarios where the user does not control the underlying library's tiling logic. Third, the lightweight pipeline as a whole demonstrates that, in the small-data regime, the dominant accuracy lever is not raw model capacity but the joint elimination of catastrophic overfitting (via under-parameterization), gradient instability (via composite BCE + Soft Dice loss), and silent coverage loss (via dynamic padding). The remaining variance is driven primarily by annotation convention drift on atypical cortices, which cannot be resolved at the algorithmic level and requires expanding the cohort to $N \approx 20$–50 with harmonized labeling.

Limitations include the small absolute cohort size, the single-site origin of the data, and the focus on a single anatomical target (left DLPFC). Future work will extend the evaluation to multi-site cohorts, additional cortical targets, and a direct comparison of mirror, reflection, and zero padding variants under matched preprocessing.

VIII. CONCLUSION

Sliding-window inference is the workhorse of 3D deep learning deployment on memory-constrained hardware, but in its naive form it silently loses peripheral voxels whenever input dimensions are not aligned with the stride. This paper formalized the boundary truncation problem in closed form and proposed a deterministic method based on dynamic asymmetric zero-padding and ensemble-by-patching reconstruction. Integrated into a custom lightweight 3D U-Net (~1.4×10^6 parameters) and evaluated on a pilot DLPFC cohort (N = 10) under LOSO cross-validation, the proposed pipeline (i) restores full volumetric coverage in 10/10 subjects, (ii) improves DSC by 0.05–0.12 in absolute terms relative to the uncorrected baseline, (iii) achieves a centroid targeting error of 5.64 ± 3.21 mm — three to five times lower than the 5-cm rule and Beam F3 clinical heuristics, (iv) executes end-to-end in ~36 s per patient on a single Tesla T4, and (v) requires only ~4.3 GB of VRAM and 5.6 MB of model weights, well within the envelope of standard clinical workstations. The proposed method provides a mathematically auditable and operationally deployable foundation for volumetric medical image inference under both data and hardware scarcity.

REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Proc. MICCAI, 2015, pp. 234–241.

[2] F. Isensee, P. F. Jaeger, S. A. A. Kohl, et al., "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," Nature Methods, vol. 18, no. 2, pp. 203–211, 2021.

[3] F. Milletari, N. Navab, and S. A. Ahmadi, “V-Net: Fully convolutional neural networks for volumetric medical image segmentation,” in Proc. 3DV, IEEE, 2016, pp. 565–571.

[4] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: learning dense volumetric segmentation from sparse annotation,” in Proc. MICCAI, Springer, 2016, pp. 424–432.

[5] N. P. Jouppi et al., “In-datacenter performance analysis of a tensor processing unit,” in Proc. 44th ISCA, 2017, pp. 1–12.

[6] K. Kamnitsas, C. Ledig, V. F. Newcombe et al., “Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation,” Medical Image Analysis, vol. 36, pp. 61–78, 2017.

[7] U. Herwig, P. Satrapi, and C. Schönfeldt-Lecuona, “Transcranial magnetic stimulation: Five-centimeter rule vs. neuronavigation,” Psychiatry Research: Neuroimaging, vol. 108, no. 1, pp. 87–96, 2001.

[8] W. Beam, J. J. Borckardt, S. T. Reeves, et al., “An efficient method for positioning the coil for transcranial magnetic stimulation of the dorsolateral prefrontal cortex,” Brain Stimulation, vol. 2, no. 1, pp. 50–54, 2009.

[9] G. Litjens, T. Kooi, B. E. Bejnordi et al., “A survey on deep learning in medical image analysis,” Medical Image Analysis, vol. 42, pp. 60–88, 2017.

[10] A. Hatamizadeh, Y. Tang, V. Nath et al., “UNETR: Transformers for 3D medical image segmentation,” in Proc. WACV, 2022, pp. 574–584.

[11] M. J. Cardoso, W. Li, R. Brown et al., “MONAI: An open-source framework for deep learning in healthcare,” arXiv:2211.02701, 2022.

[12] Q. Dou, L. Yu, H. Chen et al., “3D deeply supervised network for automated segmentation of volumetric medical images,” Medical Image Analysis, vol. 41, pp. 40–54, 2017.

[13] A. S. Lundervold and A. Lundervold, “An overview of deep learning in medical imaging focusing on MRI,” Zeitschrift für Medizinische Physik, vol. 29, no. 2, pp. 102–127, 2019.

[14] M. Mylius et al., “Definition of DLPFC and M1 according to anatomical landmarks for navigated brain stimulation: Inter-rater reliability, accuracy, and influence of gender and age,” NeuroImage, vol. 78, pp. 224–232, 2013.

[15] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. ICLR, 2015.