STRUCTURE AS STATE, NOT SKIP: BRIDGING SEMANTIC REPRESENTATION AND STRUCTURAL DETAIL FOR 3D MEDICAL IMAGE SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformer-based architectures have become widespread in 3D medical image segmentation, yet their gains over strong convolutional baselines remain limited. We trace this mismatch to a *representation–structure dilemma*: skip connections preserve structure but let information bypass the Transformer; removing them makes the Transformer indispensable but weakens structural detail. The dilemma stems from conflating a task requirement with a U-Net design choice: structure must be preserved, but it need not pass through skip connections. Decoupling the two reveals a third possibility: *structure as state, not skip*. We realize this principle with **Flumina**, a multi-stream network that maintains multi-scale persistent states within a shared Vision Transformer backbone. Each sublayer combines these states, applies a shared transformation, and produces token-level updates for each state, while identity residual mappings maintain them across depth. Flumina thus keeps the Transformer responsible for representation learning while preserving structural detail. Experiments across multiple benchmarks and random seeds demonstrate its effectiveness. Code is available at the anonymous link.

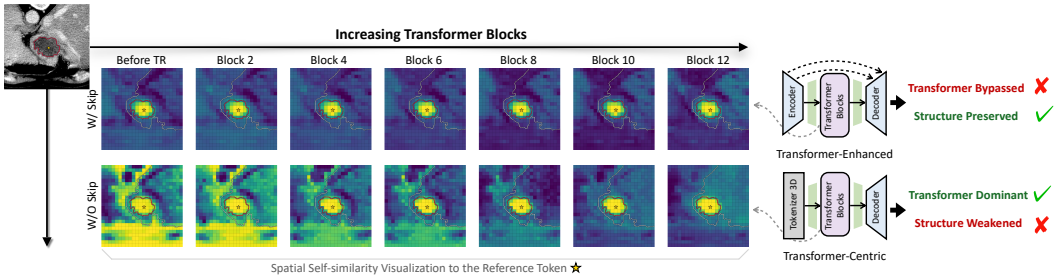

Figure 1: **The representation–structure dilemma.** Current Transformer-based medical image segmentation methods appear to exhibit a trade-off between representation learning and fine structural preservation. *Top:* U-shaped architecture with integrated Transformer blocks. Structure is preserved, but representations change only marginally. *Bottom:* Transformer-centric architecture without skip connections. Representations change substantially, but structure is weakened.

1 INTRODUCTION

3D medical image segmentation is a fundamental task in medical image analysis (Shen et al., 2017). As a dense-prediction task, it requires high-level semantic representations to identify target regions and fine-grained structural detail to delineate their boundaries (Song et al., 2026a). Semantic representations highlight task-relevant structures, whereas structural detail provides the spatial evidence needed for precise boundary delineation (Mo et al., 2022; Salpea et al., 2022). These complementary requirements motivate a central architectural question: *how can a network effectively integrate semantic representations with structural detail?*

In the convolutional era, an influential architectural response to this question is the U-shaped encoder-decoder paradigm pioneered by U-Net (Ronneberger et al., 2015), characterized by skip

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

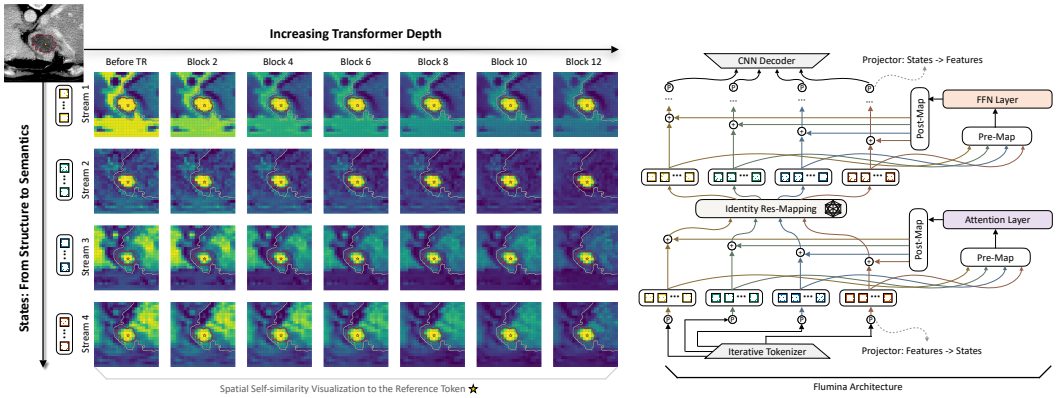

Figure 2: **Flumina: structure as state, not skip.** *Left:* The evolution of multi-scale persistent states across Transformer depth. The states evolve while retaining distinct structural and semantic roles. *Right:* The architecture of Flumina. It maintains multi-scale states within the shared ViT backbone.

connections that pass shallow structural features directly from the encoder to the decoder. Its alignment with the local and hierarchical organization of CNNs enabled the U-shaped design to balance semantic representation and structural preservation. In the years that followed, its effectiveness was repeatedly demonstrated through successive architectural refinements (Çiçek et al., 2016; Milletari et al., 2016; Oktay et al., 2018; Zhou et al., 2019; Isensee et al., 2021), establishing the U-shaped paradigm as the de facto standard in medical image segmentation.

The rise of Vision Transformers (ViTs) and the broader adoption of sequence modeling reshaped computer vision in the 2020s (Dosovitskiy et al., 2020; Liu et al., 2021; 2022; Zhu et al., 2024; Duan et al., 2025). In medical image segmentation, ViTs and other sequence models were increasingly incorporated into U-shaped architectures, aiming to improve long-range dependency modeling and semantic representation learning (Chen et al., 2021; Hatamizadeh et al., 2021; Xie et al., 2021; Hatamizadeh et al., 2022; Xing et al., 2025; Chen et al., 2025). However, these architectural advances have not translated into equally substantial performance gains, and improvements over strong convolutional baselines are often incremental (Bassi et al., 2024; Wald et al., 2026).

Behind this mismatch lies the flip side of the U-shaped paradigm: the same skip connections that enabled its success in the CNN era can constrain representation learning in the ViT era. In particular, skip connections pass shallow encoder features directly to the decoder. This preserves structural detail, but also makes the prediction less dependent on Transformer representations (Zhang et al., 2026b; Song et al., 2026b). The top row of Figure 1 shows a typical U-shaped architecture with integrated Transformer blocks. The corresponding representation trajectory suggests that the Transformer mainly performs refinement rather than serving as the primary representation learner.

To enforce Transformer usage, recent work has taken the opposite approach by removing U-shaped skip connections (Wald et al., 2026). As shown in Figure 1, although representations change substantially across depth, the skip-free design lacks explicit states for maintaining structural detail, limiting its segmentation performance.

Although they take opposite approaches, both designs reveal the same representation–structure dilemma: existing methods for medical image segmentation do not make the Transformer responsible for representation learning while preserving structural detail. This dilemma, however, is *not* inherent to medical image segmentation. It arises because the U-shaped paradigm treats two ideas as inseparable: structural information must be preserved and forwarded, and it must pass through skip connections. The former is required by dense prediction, whereas the latter is a particular design choice of U-Net. We further provide a detailed discussion in Appendix A.

Decoupling these ideas reveals a third possibility: *structure as state, not skip*. In other words, we can remove skip connections and treat multi-scale features as *persistent states* within the shared ViT backbone. We realize this principle with **Flumina** *(Latin for "streams")*, a multi-stream network that bridges semantic representation and structural detail. As shown in Figure 2, we first use a lightweight iterative tokenizer to give the ViT a greater role in representation learning. Its multi-

scale outputs initialize separate persistent states, which are maintained throughout the backbone by identity residual mappings. Each sublayer combines all states, applies one shared transformation, and dynamically adapts its output at the token level before writing it back to each one. Thus, the states exchange information while remaining separate. Our main contributions are as follows:

- We revisit the representation–structure dilemma by separating a dense-prediction requirement from an architectural choice inherited from U-Net: preserving and forwarding structural information does not require skip connections.
- We introduce Flumina, a multi-stream network that maintains multi-scale states within the shared ViT backbone through persistent states. We theoretically justify its identity residual mappings and empirically analyze how its design choices shape representations.
- Extensive experiments on 3D medical image segmentation benchmarks, conducted across multiple random seeds, demonstrate the effectiveness of Flumina.

2 Related Work

Multi-stream Modeling. Recent works have explored various extensions to standard residual connections (He et al., 2016; Baykal et al., 2023; Xie et al., 2026). Notably, Hyper-Connections (HC) generalize the single residual stream into multiple interacting streams governed by learnable mappings (Zhu et al., 2025). Subsequent variants constrain or sparsify these mappings to improve training stability and scalability (Xie et al., 2025; Liu et al., 2026; Zhang et al., 2026a). Despite these advances, Hyper-Connections and their variants commonly initialize all streams by replicating the same representation, creating an initial symmetry that can lead to stream collapse (Alimaskina et al., 2026). Overall, existing multi-stream approaches are primarily designed to expand the capacity of the residual pathway, rather than to model multi-scale features as persistent states.

Transformer in 3D medical image segmentation. Recent works have explored approaches to incorporating Transformers into 3D medical image segmentation (Chen et al., 2021; Hatamizadeh et al., 2021; Xie et al., 2021; Hatamizadeh et al., 2022). Constrained by both the quadratic cost of self-attention over long token sequences from high-resolution volumes and the limited spatial inductive bias of ViTs (Vaswani et al., 2017; Dosovitskiy et al., 2020; Zhang & Bagci, 2022), existing methods typically retain the U-shaped paradigm to preserve spatial structure, while either restricting attention to local or sparse regions or applying it only at low spatial resolutions. Recent work has also explored pure ViT architectures for 3D medical image segmentation (Wald et al., 2026), but they have yet to consistently match strong convolutional models in segmentation performance. In summary, Transformers in this domain have largely served to refine representations, in contrast to their role as the main representation learner in general computer vision.

3 Methodology

In this section, we detail the design of Flumina. By modeling multi-scale structural features as persistent states, our method leverages identity residual mappings to maintain them throughout the shared ViT backbone, keeping the Transformer responsible for representation learning while preserving structural detail. We also provide further mathematical details in Appendix C.

3.1 Preliminary

As our method builds upon existing multi-stream modeling approaches (Zhu et al., 2025; Xie et al., 2025), we begin with a brief review of how they extend residual learning from a single hidden state to multiple parallel states to facilitate the subsequent introduction of Flumina.

Let $\boldsymbol{x}_{t-1}$ denote the input state to the t-th block, where t indexes network depth. To facilitate optimization, the residual block preserves an identity path and learns an additive update:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + f_t(\boldsymbol{x}_{t-1}), \tag{1}$$

where f_t denotes the branch transformation. To increase its capacity, HC expands the residual path from a single hidden state $\boldsymbol{x}_{t-1}$ to k parallel states $\boldsymbol{X}_{t-1}$ and rewrites Eq. 1 as:

$$\boldsymbol{X}_t = \boldsymbol{H}_t^{\text{res}} \boldsymbol{X}_{t-1} + \boldsymbol{H}_t^{\text{post}} f_t(\boldsymbol{H}_t^{\text{pre}} \boldsymbol{X}_{t-1}), \tag{2}$$

where $\boldsymbol{H}_t^{\text{res}} \in \mathbb{R}^{k\times k}$, $\boldsymbol{H}_t^{\text{pre}} \in \mathbb{R}^{1\times k}$, and $\boldsymbol{H}_t^{\text{post}} \in \mathbb{R}^{k\times 1}$ are learnable mappings for mixing, reading from, and writing to the k parallel states, respectively. Specifically, HC initializes $\boldsymbol{X}_0$ with k copies of the initial state $\boldsymbol{x}_0$. However, the unconstrained $\boldsymbol{H}_t^{\text{res}}$ used by HC can destabilize propagation across depth. To address this issue, Xie et al. (2025) introduce Manifold-Constrained Hyper-Connections (mHC), a variant that constrains $\boldsymbol{H}_t^{\text{res}}$ to be doubly stochastic:

$$\boldsymbol{H}_t^{\text{res}} \geq 0, \qquad \boldsymbol{H}_t^{\text{res}} \mathbf{1}_k = \mathbf{1}_k, \qquad (\boldsymbol{H}_t^{\text{res}})^\top \mathbf{1}_k = \mathbf{1}_k, \tag{3}$$

where $\mathbf{1}_k$ is the k-dimensional all-ones vector. Eq. 3 keeps residual mixing non-expansive and stabilizes propagation across depth. Overall, these methods widen the residual path to increase capacity. However, replicated initialization provides no initial state diversity, while repeated doubly stochastic mixing can work against our goal of preserving distinct multi-scale states.

In contrast, Flumina initializes $\boldsymbol{X}_0$ with distinct multi-scale structural features and fixes $\boldsymbol{H}_t^{\text{res}} = \boldsymbol{I}_k$ at every depth. We further provide a theoretical analysis of this design in Appendix C.

3.2 Flumina Architecture

We now introduce the architecture of Flumina, which maintains multi-scale structural features as persistent states within a shared ViT backbone. It first uses an iterative tokenizer to construct the multi-scale initial states $\boldsymbol{X}_0$. Within the shared ViT backbone, dynamic read and write mechanisms exchange information among the states, while identity residual mappings preserve each state across depth. Finally, the states are restored to their corresponding spatial resolutions and progressively decoded into the segmentation mask. We describe these components in detail as follows.

Multi-Scale State Initialization. As discussed in Section 3.1, Flumina requires initial states with distinct roles to provide complementary information throughout the backbone. We therefore implement the iterative tokenizer as a lightweight four-stage residual CNN with channel widths $(32, 64, 128, 256)$. We limit the number of stages and their channel widths such that more of the representation learning is performed by the subsequent Transformer blocks. The first stage preserves the input resolution, while each subsequent stage halves the spatial resolution with a stride-2 convolution. For an input volume of size $D \times H \times W$, stage i produces:

$$\boldsymbol{F}_i \in \mathbb{R}^{C_i \times \frac{D}{2^{i-1}} \times \frac{H}{2^{i-1}} \times \frac{W}{2^{i-1}}}, \tag{4}$$

where C_i is the channel width of the i-th stage. Then, the i-th projector embeds $\boldsymbol{F}_i$ into the initial state $\boldsymbol{x}_i$ using a single convolution with kernel size and stride $p_i = 8/2^{i-1}$. As shown in Figure 3, the single-convolution projectors map similarly scaled tokenizer features to states with markedly different magnitudes. We therefore apply RMSNorm before flattening the embeddings into initial states. This allows the read and write mechanisms to operate on states of comparable magnitude. Without normalization, the updates become negligible relative to the high-magnitude states derived from shallow features (see Figure 7 for details).

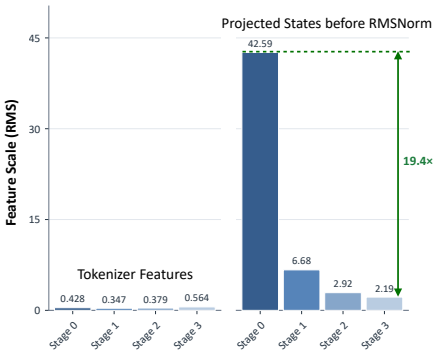

Figure 3: **Feature scales before and after convolutional embedding.** RMS magnitudes of the four tokenizer features and their corresponding states after a convolution but before RMSNorm. The resulting states in different stages differ substantially in scale.

ViTs commonly tokenize images at stride 16. We instead use stride 8 to reduce information loss and thereby retain more spatial detail in the initial states $\boldsymbol{X}_0$. This choice also reduces the receptive field of the initial tokens, giving the Transformer more scope for contextual modeling. Similar choices have also been adopted in recent methods (Wald et al., 2026; Song et al., 2026b). This finer stride increases the sequence length. Take a 128^3 crop, for example: it produces a sequence of $L = 4096$ tokens. However, Flumina reads the four states into a single sequence of length L rather than concatenating them into $4L$ tokens. With efficient exact-attention kernels (Dao et al., 2022; Zadouri et al., 2026) and no autoregressive decoding, the overall computation remains practical.

Dynamic Read and Write. Following Eq. 2, Flumina fixes the residual mapping to $\boldsymbol{I}_k$. Unlike previous methods (Zhu et al., 2025; Xie et al., 2025), Flumina makes the read and write deliberately

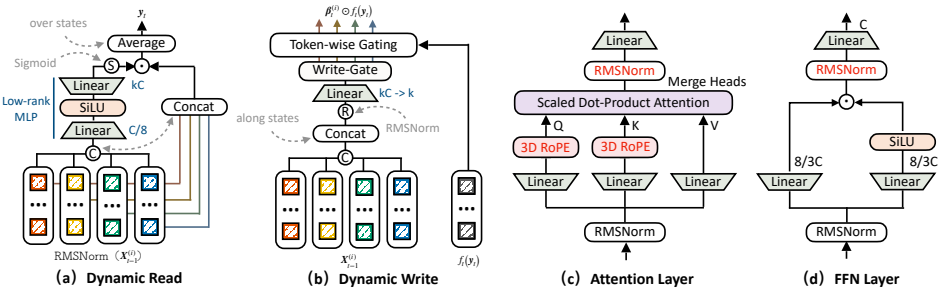

Figure 4: **Micro design of Flumina.** **(a)** A low-rank MLP produces feature-wise read gates; the gated normalized states are then averaged to get $\boldsymbol{y}_t$. **(b)** Token-wise write gates scale the shared update for each state. **(c)** A modernized attention layer. It applies 3D RoPE to queries and keys, with RMSNorm at the input and before the output projection. **(d)** A modernized FFN layer. It replaces the standard MLP with SwiGLU and uses the same extra-normalization design.

asymmetric. Specifically, the read uses feature-wise gates to combine the distinct structural and semantic information carried by the multi-scale states at a finer granularity. The resulting update is

$$\boldsymbol{y}_t = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{\alpha}_t^{(i)} \odot \mathrm{RMSNorm}\Big(\boldsymbol{X}_{t-1}^{(i)}\Big), \quad \boldsymbol{X}_t^{(i)} = \boldsymbol{X}_{t-1}^{(i)} + \boldsymbol{\beta}_t^{(i)} \odot f_t(\boldsymbol{y}_t)\,. \tag{5}$$

Here, $\boldsymbol{\alpha}_t^{(i)}$ is a feature-wise read gate with values in $(0, 1)$, whereas $\boldsymbol{\beta}_t^{(i)}$ is a token-wise write gate with values in $(0, 2)$ and is shared across channels.

As shown in Figure 4(a), the read normalizes each state independently and concatenates the normalized states along the feature dimension. A low-rank MLP with bottleneck width $C/8$ predicts the read gates; the gated normalized states are then averaged to form $\boldsymbol{y}_t$. For the write, we concatenate the current states along the feature dimension and denote the result by $\boldsymbol{Z}_{t-1}$, with feature width kC. As shown in Figure 4(b), the write mechanism maps each token in $\boldsymbol{Z}_{t-1}$ to k write coefficients, one for each state:

$$\boldsymbol{\beta}_t = \mathcal{G}_t^{\mathrm{write}}(\boldsymbol{Z}_{t-1}) := 2\sigma\Big(\lambda_t\,\mathrm{RMSNorm}(\boldsymbol{Z}_{t-1})\,\boldsymbol{W}_t^{\beta} + \boldsymbol{b}_t^{\beta}\Big)\,, \tag{6}$$

where $\boldsymbol{W}_t^{\beta}$ projects each token from kC features to k state-specific logits. The scalar λ_t scales this input-dependent term, while $\boldsymbol{b}_t^{\beta}$ provides a learned state-wise bias shared across tokens. The scaled sigmoid then yields one write coefficient for each token–state pair.

Transformer Block Design. We build the shared backbone on a plain ViT and modernize its blocks with recent advancements (Zhang & Sennrich, 2019; Shazeer, 2020; Shleifer et al., 2021; Touvron et al., 2021; Gervet et al., 2023; Fang et al., 2024; Su et al., 2024; Wald et al., 2026; Song et al., 2026b). Figure 4(c) and (d) illustrate their designs.

First, we adopt an extra-normalization design for stable optimization, applying RMSNorm at each sublayer input and before its output projection (Shleifer et al., 2021; Fang et al., 2024; Wald et al., 2026). RMSNorm removes the mean-centering step of LayerNorm, reducing computation (Zhang & Sennrich, 2019). Second, we apply RoPE to queries and keys along the depth, height, and width axes to improve relative-position modeling in 3D (see Appendix B for details) (Gervet et al., 2023; Wald et al., 2026; Song et al., 2026b). Third, we replace the standard MLP with a SwiGLU FFN (Shazeer, 2020; Fang et al., 2024). Moreover, a learnable channel-wise LayerScale vector rescales each branch output before stochastic depth and dynamic writing (Touvron et al., 2021).

In Flumina, each modernized Transformer block serves as a shared interaction module within the backbone: attention and FFN use separate read and write parameters to exchange information among states, while identity residual mappings retain a direct path for each state across depth.

Multi-Scale State Decoding. After the final Transformer block, each state is reshaped into a 3D feature map and restored to its corresponding tokenizer resolution. We then decode the states from coarse to fine, progressively fusing them until the input resolution is recovered. A lightweight prediction head produces the voxel-wise segmentation logits. Since the decoder receives only the final states, no tokenizer feature bypasses the ViT through a skip connection.

3.3 Scaling Up Flumina

We construct four Flumina variants for different computational budgets by scaling the shared ViT backbone. We vary its embedding width, depth, and number of attention heads, while keeping the four-state design, tokenizer widths, read and write mechanisms, and decoder widths fixed. Table 1 summarizes the resulting configurations. Further details on computational complexity and model efficiency are provided in Appendix H.

Table 1: **Scaling configurations of Flumina.** Parameters are reported in millions (M).

Flumina	ViT Backbone			Params (M)
	Embed Dim	Layers	Heads	
S	288	8	6	27.3
B	480	12	10	66.1
M	792	12	12	147.9
L	864	16	12	213.5

Note. Parameters are measured using a crop size of 128^3.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate Flumina on five public 3D segmentation benchmarks spanning diverse modalities, anatomies, and target scales—from large organs to small lesions: 1) **AMOS-22** (Ji et al., 2022), 300 CT and 60 MRI scans with abdominal multi-organ annotations; 2) **LiTS** (Bilic et al., 2023), 131 contrast-enhanced CT volumes with liver and tumor annotations; 3) **BraTS-25-GLI** (Aboian et al., 2025), 2,872 labeled multi-parametric MRI examinations (1,251 pre-treatment and 1,621 post-treatment), each with four co-registered modalities and glioma-subregion annotations; 4) **KiTS-23** (Heller et al., 2023), 489 contrast-enhanced CT volumes with kidney, tumor, and cyst annotations; and 5) **MAMA-MIA** (Garrucho et al., 2025), 1,506 multi-center breast DCE-MRI examinations with expert primary-tumor masks. Each dataset is standardized and split into training, validation, and test sets at approximately a 7:1:2 ratio. Further details, including resolution normalization, voxel spacing and pre-processing pipelines, are provided in Appendix D.

Metrics. We evaluate region overlap using the Dice similarity coefficient (**DSC**) and intersection over union (**IoU**), and boundary accuracy using the 95th-percentile Hausdorff distance (**HD95**). DSC and IoU are reported as percentages, whereas HD95 is computed using the physical voxel spacing and reported in millimeters.

Comparison methods. For a comprehensive evaluation, we compare Flumina with 12 representative 3D medical image segmentation models[1]: U-Net (Ronneberger et al., 2015), CoTr (Xie et al., 2021), SwinUNETR (Hatamizadeh et al., 2021), UNETR (Hatamizadeh et al., 2022), MedNeXt (Roy et al., 2023), UNETR++ (Shaker et al., 2024), SegMamba (Xing et al., 2024), Zig-RiR (Chen et al., 2025), SegMamba-V2 (Xing et al., 2025), Primus (Wald et al., 2026), CROWn (Huang et al., 2026), and Primus-V2 (Wald et al., 2026). Together, these baselines cover a broad range of architectures, including CNNs, Transformers, and sequence models.

Implementation details. Our implementation is based on PyTorch 2.12.1. All training experiments are conducted on a node equipped with four NVIDIA A100 GPUs. Within each dataset, all methods use the same crop size and are trained for 1,000 epochs with 250 iterations per epoch. All models are trained with automatic mixed precision (AMP) using bfloat16. For Flumina, we additionally enable PyTorch's native scaled dot-product attention (SDPA) to improve the computational and memory efficiency of attention. For the main comparisons, each experiment is repeated with three random seeds, and the mean performance is reported. Optimizers and learning-rate schedules follow the recommended configurations for each method. All models are trained from scratch under direct supervision using the same compound objective, which combines Dice loss with weighted cross-entropy. Further details on the training configurations are provided in Appendix E.

4.2 Main Results

Tables 2 and 3 evaluate the models in two complementary regimes, covering multi-organ semantic segmentation and fine-grained delineation of localized targets.

[1]A separate note concerns nnU-Net (Isensee et al., 2021). Although nnU-Net is widely used for medical image segmentation, its results, implementation details, and further analysis are provided in Appendix G.4 because its dataset-adaptive configuration would confound an architecture-only comparison.

Table 2: **Performance comparison (DSC %) on AMOS-22.** Flumina variants are shaded in **green**. Best and second-best results are highlighted in **purple** and **red**, respectively.

AMOS-22 CT Test Split																	
Architectures	**Venue**	**Spleen**	**R. Kid**	**L. Kid**	**Gall.**	**Eso.**	**Liver**	**Stom.**	**Aorta**	**IVC**	**Panc.**	**RAG**	**LAG**	**Duo.**	**Blad.**	**Pros.**	**Avg.**
U-Net	MICCAI'15[†]	94.78	94.81	93.56	77.68	83.89	97.29	90.01	94.27	89.46	83.23	77.25	77.65	77.51	86.61	80.52	86.57
CoTr	MICCAI'21	94.44	93.97	92.41	78.81	81.73	97.08	88.26	93.21	87.94	81.25	77.08	75.84	75.44	83.97	76.34	85.19
SwinUNETR	MICCAIW'21	93.85	93.84	92.62	72.47	79.21	96.77	86.45	93.07	87.16	78.38	74.13	72.29	71.05	78.95	71.76	82.80
UNETR	WACV'22	90.14	85.86	83.75	57.74	64.58	94.73	75.74	87.26	79.45	65.89	61.66	55.62	54.88	67.26	60.98	72.37
MedNeXt	MICCAI'23	95.21	95.17	**94.93**	80.20	84.12	97.52	90.28	94.49	89.95	84.08	77.67	77.78	79.11	86.53	82.31	87.29
UNETR++	TMI'24	92.82	92.65	91.38	67.47	73.49	96.09	83.85	91.73	84.83	76.78	69.49	67.38	67.42	75.80	71.09	80.15
SegMamba	MICCAI'24	95.04	95.14	93.71	77.89	83.33	97.34	89.20	94.14	89.36	83.44	77.26	77.16	77.69	86.53	80.18	86.49
Zig-RiR	TMI'25	92.05	89.30	88.22	68.41	75.31	95.69	80.41	90.86	79.68	72.31	65.99	65.42	64.94	78.02	69.16	78.39
SegMamba-V2	TMI'25	95.24	95.12	93.86	80.38	83.89	97.53	90.49	94.49	89.77	83.97	78.72	77.92	78.26	87.21	79.46	87.09
Primus	TMLR'26	96.11	94.71	93.26	77.99	82.29	97.45	89.41	93.98	89.22	83.68	75.44	75.47	78.04	90.26	82.90	86.68
CROWn	CVPR'26	94.67	94.58	93.56	76.29	81.66	97.12	88.00	93.75	88.68	81.86	75.99	75.46	75.80	83.95	78.70	85.34
Primus-V2	TMLR'26	**96.68**	94.91	94.10	79.86	84.24	97.51	90.10	94.42	90.04	84.35	76.50	76.67	80.47	91.16	82.69	87.58
Flumina-S	Ours	95.92	95.45	94.00	82.26	85.36	97.70	90.48	94.93	90.31	84.79	79.28	79.00	**80.66**	89.57	83.00	88.18
Flumina-B	Ours	95.97	95.56	94.09	**82.99**	**85.61**	97.75	90.55	**95.04**	90.48	**85.26**	**79.86**	**79.59**	**80.81**	**91.54**	**83.17**	**88.55**
Flumina-M	Ours	**96.55**	**95.89**	**94.44**	82.40	85.32	**97.76**	**90.78**	94.91	**90.52**	**85.32**	79.86	79.50	80.38	91.26	82.67	88.50
Flumina-L	Ours	96.03	**95.75**	94.43	**83.16**	**85.67**	**97.79**	**90.81**	**95.07**	**90.55**	85.16	**79.88**	**79.62**	80.20	**91.92**	**83.90**	**88.66**
AMOS-22 MRI Test Split																	
Architectures	**Venue**	**Spleen**	**R. Kid**	**L. Kid**	**Gall.**	**Eso.**	**Liver**	**Stom.**	**Aorta**	**IVC**	**Panc.**	**RAG**	**LAG**	**Duo.**	**Blad.**	**Pros.**	**Avg.**
U-Net	MICCAI'15	96.45	96.11	95.96	71.73	75.08	97.39	88.44	92.41	90.60	84.20	62.28	50.67	67.38	–	–	82.21
CoTr	MICCAI'21	94.84	94.85	94.82	75.98	71.89	97.25	86.64	91.33	89.26	80.85	61.87	50.31	65.88	–	–	81.21
SwinUNETR	MICCAIW'21	95.07	94.72	95.18	66.65	68.65	96.99	83.46	91.63	88.87	78.43	60.34	45.42	61.01	–	–	78.96
UNETR	WACV'22	90.36	90.20	90.59	50.98	54.03	95.32	73.82	85.57	79.92	63.47	48.03	32.05	43.97	–	–	69.10
MedNeXt	MICCAI'23	96.48	95.53	95.07	72.82	76.02	97.68	89.01	92.58	90.90	84.93	62.75	50.26	68.00	–	–	82.46
UNETR++	TMI'24	93.87	93.47	94.52	65.42	64.59	96.72	82.76	91.15	86.64	78.70	55.92	45.63	56.98	–	–	77.41
SegMamba	MICCAI'24	95.35	95.99	95.67	72.15	75.20	97.66	88.02	92.83	90.76	83.05	63.44	51.05	67.10	–	–	82.18
Zig-RiR	TMI'25	86.57	88.76	90.34	60.12	64.15	95.49	69.25	88.26	81.06	70.23	47.71	35.93	51.69	–	–	71.50
SegMamba-V2	TMI'25	96.75	96.20	96.03	72.82	74.62	97.63	88.58	92.75	90.69	84.45	64.06	51.52	69.09	–	–	82.71
Primus	TMLR'26	96.69	95.99	95.72	74.80	73.55	97.77	89.73	92.71	90.67	83.95	63.83	50.42	69.64	–	–	82.73
CROWn	CVPR'26	95.33	95.71	95.50	72.98	72.15	97.29	86.85	92.27	89.62	81.30	62.25	49.14	65.77	–	–	81.24
Primus-V2	TMLR'26	95.95	96.21	95.75	76.01	76.19	97.83	**90.43**	92.98	91.08	85.28	63.86	52.53	**71.19**	–	–	83.48
Flumina-S	Ours	97.05	96.35	96.28	75.10	76.39	97.91	89.56	93.42	91.48	84.37	64.53	**54.95**	69.30	–	–	83.59
Flumina-B	Ours	97.20	**96.51**	96.35	**84.12**	**76.99**	**97.99**	90.20	**93.47**	91.48	85.25	**65.77**	54.33	69.62	–	–	**84.56**
Flumina-M	Ours	**97.22**	**96.54**	**96.35**	**82.45**	76.76	**98.00**	89.88	93.29	**91.63**	**85.30**	64.72	53.03	**70.24**	–	–	**84.26**
Flumina-L	Ours	**97.23**	95.39	**96.40**	74.79	**77.05**	97.98	**90.40**	**93.58**	**91.66**	**85.42**	**64.82**	**54.42**	68.29	–	–	83.65

Note. [†] Venue refers to the original U-Net paper. We evaluate a fixed 3D variant; implementation details are provided in Appendix E. All four Flumina variants achieve statistically significant improvements over the baselines (Wilcoxon signed-rank test on per-case DSC, $p < 0.05$).

Table 3: **Performance comparison (DSC %) across multiple datasets.** Flumina variants are shaded in **green**. Best and second-best results are highlighted in **purple** and **red**, respectively. Params (millions) and MACs (trillions) are reported for a 128^3 input crop.

Architectures	**Complexity**		**LiTS**			**KiTS-23**				**BraTS-25-GLI**							**MAMA-MIA**	**Avg.**
	Params	**MACs**	**Liver**	**Tumor**	**Mean**	**Kid.&Mass.**	**Mass.**	**Tumor**	**Mean**	**WT**	**TC**	**ET**	**NETC**	**SNFH**	**RC**	**Mean**	**Tumor**	
U-Net	90.31	1.91	95.72	65.81	80.77	96.41	79.02	72.60	82.68	90.46	79.39	78.71	73.44	86.75	76.97	80.95	74.17	79.64
CoTr	41.9	0.70	95.50	66.24	80.87	96.07	76.89	72.35	81.77	90.42	79.43	78.34	74.27	86.53	77.09	81.01	73.57	79.30
SwinUNETR	62.2	0.77	94.93	64.12	79.53	96.41	77.10	71.33	81.61	90.25	80.01	79.21	**76.67**	86.54	76.26	81.49	74.60	79.31
UNETR	102.5	0.18	92.89	56.09	74.49	95.15	69.29	63.69	76.04	88.48	75.29	74.69	69.34	84.32	71.73	77.31	69.63	74.37
MedNeXt	10.5	0.17	**96.29**	71.47	83.88	96.63	80.29	75.50	84.14	90.40	80.53	79.74	**75.90**	86.71	76.74	81.67	76.24	81.48
UNETR++	42.6	0.07	89.43	47.83	68.63	95.37	72.43	68.42	78.74	89.61	77.53	76.54	72.65	85.54	74.74	79.43	64.49	72.82
SegMamba	67.4	1.48	95.68	64.65	80.16	96.45	78.67	72.57	82.56	90.18	80.34	79.62	73.89	86.55	75.82	81.07	72.77	79.14
Zig-RiR	33.7	0.21	94.46	59.95	77.21	94.38	67.93	62.59	74.96	86.80	75.27	73.60	68.72	81.95	70.78	76.19	71.40	74.94
SegMamba-V2	138.8	1.87	94.93	65.81	80.37	96.34	78.68	73.20	82.74	90.14	79.43	78.75	74.08	86.20	75.59	80.70	74.84	79.66
Primus	150.4	1.07	95.52	64.99	80.26	96.40	78.29	75.65	83.45	89.85	77.80	76.60	73.04	85.91	76.38	79.93	73.86	79.38
CROWn	23.8	0.48	95.87	66.04	80.95	96.34	77.03	71.32	81.56	89.68	77.98	77.53	72.88	85.75	74.34	79.69	75.28	79.37
Primus-V2	149.6	1.15	95.91	72.43	84.17	96.32	80.91	77.43	84.88	90.16	79.84	78.47	72.94	86.17	**78.21**	80.96	75.58	81.40
Flumina-S	27.3	0.42	96.01	73.78	84.89	96.86	81.83	78.02	85.57	90.56	80.88	80.23	74.54	86.95	77.66	81.80	76.31	82.14
Flumina-B	66.1	0.69	96.10	75.28	85.69	97.06	81.85	**78.55**	**85.82**	**90.71**	81.32	**80.87**	74.61	87.08	77.72	**82.05**	76.45	82.50
Flumina-M	147.9	1.15	**96.12**	**76.68**	**86.40**	**97.18**	**82.12**	78.12	85.81	90.67	**81.37**	80.81	73.49	**87.16**	**77.76**	81.88	**76.55**	**82.66**
Flumina-L	213.5	1.56	95.95	**77.29**	**86.62**	**97.10**	**82.52**	**79.12**	**86.25**	**90.93**	**81.62**	**80.93**	74.47	**87.28**	77.50	**82.12**	**76.83**	**82.95**

Note. All four Flumina variants achieve statistically significant improvements over the baselines (Wilcoxon signed-rank test on per-case DSC, $p < 0.05$).

Across both evaluation settings, Flumina delivers competitive performance. Scaling Flumina generally improves performance, but the gains vary across datasets and are not monotonic across model sizes. Further evaluation details are provided in Appendix G.

4.3 Ablation Study

In the following, we organize our ablation study around four questions:

- **Q1:** Does Flumina reconcile representation learning with structural preservation? (Table 4, Figure 5 and Figure 6)
- **Q2:** Does multi-scale state initialization outperform replication and early fusion? (Table 5)
- **Q3:** How does the choice of residual mapping affect segmentation performance? (Table 5)
- **Q4:** How do normalization and read/write gating affect segmentation performance? (Table 5 and Figure 7)

To limit computational cost, all ablation experiments are conducted with a single random seed. Figure 5 illustrates the architectural variants, while Table 4 reports the corresponding performance. Figure 6 complements these results with t-SNE visualizations.

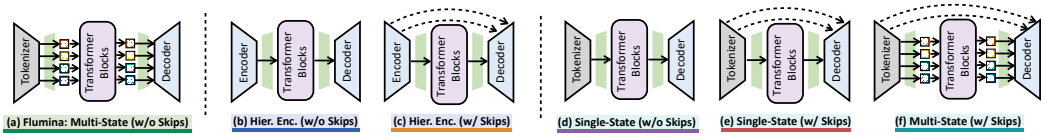

Figure 5: **Architectural variants for the representation–structure ablation.** Label backgrounds and underline colors correspond to the ablation blocks and variant markers in Table 4, respectively.

Table 4: **Component-wise analysis of multi-state modeling.** Flumina is highlighted with a **green** background. Variants (a)–(f) correspond to the architectures in Figure 5. Red annotations denote decreases in percentage points relative to Flumina.

Variant	Components				Average Dataset DSC (%)					Overall
	Iter. Tok.	Skips	Multi-State	Tr. → Id.	AMOS-22	LiTS	KiTS-23	BraTS-25-GLI	MAMA-MIA	Mean
(a) Flumina	•	○	•	○	**88.19**	**85.30**	**86.32**	**82.10**	**77.03**	**83.79**
(b) w/ Tr.	○	○	○	○	80.20 – 7.99	79.94 – 5.36	80.62 – 5.70	75.67 – 6.43	71.00 – 6.03	77.48 – 6.31
(c) w/ Tr.	○	•	○	○	85.08 – 3.11	81.65 – 3.65	82.52 – 3.80	80.95 – 1.15	75.18 – 1.85	81.08 – 2.71
(d) w/ Tr.	•	○	○	○	86.28 – 1.91	83.31 – 1.99	83.29 – 3.03	80.10 – 2.00	74.27 – 2.76	81.45 – 2.34
(e) w/ Tr.	•	•	○	○	85.28 – 2.91	80.81 – 4.49	81.77 – 4.55	80.66 – 1.44	75.26 – 1.77	80.76 – 3.03
(f) w/ Tr.	•	•	•	○	87.81 – 0.38	82.75 – 2.55	85.50 – 0.82	81.42 – 0.68	76.30 – 0.73	82.76 – 1.03
(a) w/ Id.	•	○	•	•	78.34 – 9.85	76.17 – 9.13	72.97 – 13.35	80.94 – 1.16	71.55 – 5.48	75.99 – 7.80
(b) w/ Id.	○	○	○	•	78.12 – 10.07	79.49 – 5.81	80.39 – 5.93	75.26 – 6.84	71.61 – 5.42	76.97 – 6.82
(c) w/ Id.	○	•	○	•	84.78 – 3.41	83.51 – 1.79	82.24 – 4.08	81.07 – 1.03	75.06 – 1.97	81.33 – 2.46
(d) w/ Id.	•	○	○	•	76.54 – 11.65	76.84 – 8.46	71.86 – 14.46	78.66 – 3.44	70.50 – 6.53	74.88 – 8.91
(e) w/ Id.	•	•	○	•	77.50 – 10.69	77.31 – 7.99	72.66 – 13.66	79.95 – 2.15	72.33 – 4.70	75.95 – 7.84
(f) w/ Id.	•	•	•	•	78.11 – 10.08	78.11 – 7.19	74.56 – 11.76	80.20 – 1.90	72.47 – 4.56	76.69 – 7.10

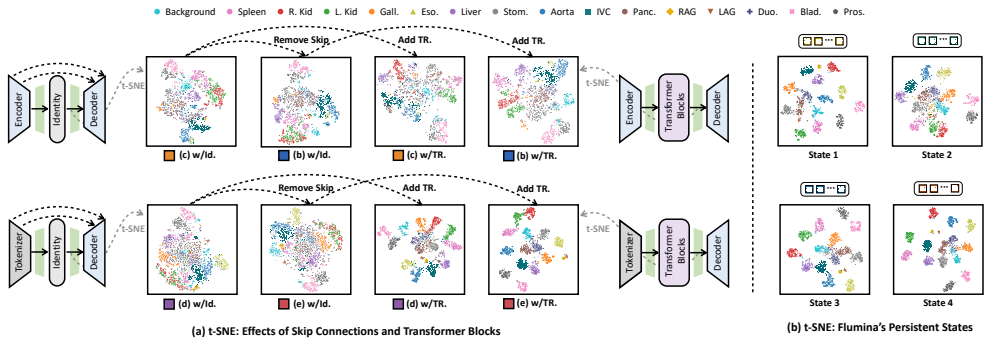

Figure 6: **t-SNE visualization of feature representations.** **(a)** Architectural variants with Transformer blocks or identity mappings, labeled as in Figure 5. **(b)** Flumina's four persistent states. Colors indicate anatomical classes and background.

Representation–structure dilemma. To further examine the representation–structure dilemma observed in Figure 1, we remove skips from semantic encoder–ViT variants and observe greater Transformer gains but lower overall performance, as shown in Table 4. These observations motivate replacing the semantic encoder with an iterative tokenizer to give the ViT a greater role in representation learning, while maintaining multi-scale structure as persistent states. The t-SNE visualization in Figure 6 provides complementary evidence that removing skips improves class separation and increases the gains from Transformer blocks. Compared with the semantic encoder, the tokenizer shows weaker class separation without Transformer blocks but a larger improvement when they are included. Besides, all states in Flumina exhibit clear and complementary class-separation patterns.

Structure as persistent states. We next examine whether persistent states can provide an alternative to encoder–decoder skip connections. Interestingly, with the iterative tokenizer and a single state, removing skips improves average performance (Table 4). Yet skips become beneficial when Transformer blocks are replaced with identity mappings.

Table 5: **Component-wise analysis of Flumina.** Flumina is highlighted with a **green** background. Colored cells highlight the design choices under ablation. Green and red annotations denote increases and decreases, respectively, in percentage points relative to Flumina.

Study	Configuration			Average Dataset DSC (%)					Overall
	State Path	H_{res}	Local Design	AMOS-22	LiTS	KiTS-23	BraTS-25-GLI	MAMA-MIA	Mean
Flumina	**Scale-grounded**	**Id.**	**Full**	**88.19**	**85.30**	**86.32**	**82.10**	**77.03**	**83.79**
Q2	Replicated	Id.	Full	87.93 – 0.26	81.00 – 4.30	85.35 – 0.97	81.51 – 0.59	75.61 – 1.42	82.28 – 1.51
	Early-fused	Id.	Full	86.65 – 1.54	82.85 – 2.45	82.93 – 3.39	80.12 – 1.98	74.88 – 2.15	81.49 – 2.30
Q3	Scale-grounded	HC	Full	87.98 – 0.21	83.76 – 1.54	84.96 – 1.36	81.50 – 0.60	75.58 – 1.45	82.75 – 1.04
	Scale-grounded	mHC	Full	88.05 – 0.14	85.46 + 0.16	85.47 – 0.85	81.89 – 0.21	75.72 – 1.31	83.32 – 0.47
Q4	Scale-grounded	Id.	No P-Norm	87.95 – 0.24	84.74 – 0.56	84.36 – 1.96	81.71 – 0.39	75.34 – 1.69	82.82 – 0.97
	Scale-grounded	Id.	Uniform Read	88.03 – 0.16	82.40 – 2.90	85.85 – 0.47	82.00 – 0.10	76.18 – 0.85	82.89 – 0.90
	Scale-grounded	Id.	Shared Write	87.92 – 0.27	84.36 – 0.94	85.88 – 0.44	81.49 – 0.61	76.41 – 0.62	83.21 – 0.58

We propose the following interpretation: tokenizer features are useful for prediction, but routing them directly to the decoder creates a competing pathway that may reduce reliance on ViT representations. Without skips, introducing multi-state modeling improves performance across all five datasets. Adding skips to Flumina, however, lowers performance on each. These results support preserving structural information as persistent states within the shared backbone, allowing it to participate in representation learning without a separate encoder–decoder bypass.

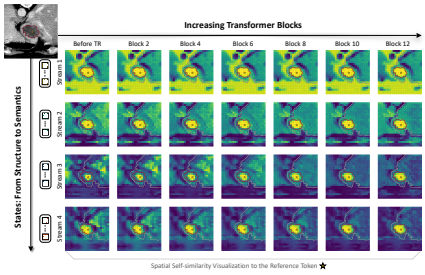

Figure 7: **Cosine similarity of features without projector RMSNorm.** Cosine similarity changes little across depth, especially in shallow-feature states.

Transformer contribution. Consistent with these observations, Table 4 shows that Transformer blocks substantially improve Flumina, whereas they offer little benefit or even reduce performance in semantic encoder variants, which highlights the shared ViT's role in learning from multi-scale persistent states.

Multi-state initialization. Flumina initializes states from different tokenizer stages to retain scale-specific structural and semantic information. To assess this design, we compare two alternatives: replication copies the element-wise mean of four projected features into four states, while early fusion uses it as one. As shown in Table 5, both alternatives reduce performance, supporting the preservation of scale-specific information in separate states throughout representation learning.

Identity residual mapping. To evaluate the identity residual mapping in Eq. 2, we replace it with unconstrained HC or doubly stochastic mHC mixing. Table 5 shows that mHC outperforms HC across all five datasets, yet identity performs best on average. Identity also avoids residual mixing and mHC's Sinkhorn–Knopp iterations. These results favor preserving states through identity paths while learning cross-state interactions through shared ViT updates.

Micro design. Removing projector RMSNorm or replacing the gated read with uniform averaging reduces performance across all five datasets. Fixing write coefficients to one causes a smaller average drop. These results validate Flumina's micro-design choices. Moreover, Figure 7 shows cosine similarity evolution without projector RMSNorm, highlighting its role for effective ViT updates.

Further analyses and visualizations are provided in Appendix G and Appendix I.

5 Conclusion

In this work, we revisit the representation–structure dilemma in 3D medical image segmentation: reconciling semantic representation with structural detail is not simply a matter of retaining or removing skip connections. Building on this insight, we propose Flumina, which treats features as persistent states within a shared ViT backbone. Across five benchmarks, Flumina achieves competitive performance, while ablations support the complementary roles of persistent multi-state modeling, identity residual mapping, and Transformer updates. Our findings highlight that structural preservation can be integrated into representation learning without encoder-decoder skip connections.

AI USE STATEMENT

The main mathematical derivations and proofs were developed and written by the authors, while large language models (LLMs) were used to assist with simplifying notation and mathematical expressions to improve the clarity, conciseness, and flow of their presentation. LLMs and LLM-based coding agents were also used to assist with writing code, including code comments and the implementation of selected functions, as well as drafting technical documentation describing code functionality and computational workflows. In addition, LLMs were used for language editing to improve the clarity, conciseness, and readability of the manuscript. The authors manually reviewed the AI-assisted code and documentation to check their correctness and consistency with the intended methods and implementation, and reviewed all AI-assisted manuscript revisions to ensure that they preserved the intended mathematical and scientific meaning. The LLMs used in this work were primarily GPT 5.5 (OpenAI, 2026a), GPT 5.6 Sol (OpenAI, 2026b) and GPT 6 Astra (OpenAI, 2026c). OpenAI Codex (OpenAI, 2025) was also used as an LLM-based coding agent to assist with the implementation and technical documentation tasks described above.

ETHICS STATEMENT

This study uses only publicly available medical imaging datasets and involves no new data collection or participant recruitment. No additional ethics approval was required for this study.

REPRODUCIBILITY STATEMENT

We describe our method in Section 3 and the experimental setup in Section 4. We provide further architectural details in Appendix B and document dataset splits and preprocessing in Appendix D. Training configurations are detailed in Appendix E, and inference settings and evaluation metrics in Appendix F. Our implementation is available through the anonymous repository at `https://anonymous.4open.science/r/Flumina`.

REFERENCES

Mariam Aboian, Udunna Anazodo, Ujjwal Baid, Spyridon Bakas, Evan Calabrese, Gian Marco Conte, Anahita Kazerooni, Yannick Kirchhoff, Florian Kofler, Dominic LaBella, Hongwei Li, Marius George Linguraru, Bjoern Menze, Jeff Rudie, Philipp Vollmuth, and Benedikt Wiestler. MICCAI 2025 Lighthouse Challenge: Brain Tumor Segmentation Cluster of Challenges (BraTS), 2025. URL https://doi.org/10.5281/zenodo.15094823.

Ekaterina Alimaskina, Gleb Molodtsov, and Aleksandr Beznosikov. Analyzing stream collapse in hyper-connections: From diagnosis to mitigation. *arXiv preprint arXiv:2606.03483*, 2026.

Pedro R Bassi, Wenxuan Li, Yucheng Tang, Fabian Isensee, Zifu Wang, Jieneng Chen, Yu-Cheng Chou, Saikat Roy, Yannick Kirchhoff, Maximilian Rokuss, et al. Touchstone benchmark: Are we on the right way for evaluating ai algorithms for medical segmentation? *Advances in Neural Information Processing Systems*, 37:15184–15201, 2024.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 3319–3327. IEEE, 2017.

Cenk Baykal, Dylan Cutler, Nishanth Dikkala, Nikhil Ghosh, Rina Panigrahy, and Xin Wang. Alternating updates for efficient transformers. *Advances in Neural Information Processing Systems*, 36:76718–76736, 2023.

Patrick Bilic, Patrick Christ, Hongwei Bran Li, Eugene Vorontsov, Avi Ben-Cohen, Georgios Kaissis, Adi Szeskin, Colin Jacobs, Gabriel Efrain Humpire Mamani, Gabriel Chartrand, et al. The liver tumor segmentation benchmark (lits). *Medical image analysis*, 84:102680, 2023.

Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

Tianxiang Chen, Xudong Zhou, Zhentao Tan, Yue Wu, Ziyang Wang, Zi Ye, Tao Gong, Qi Chu, Nenghai Yu, and Le Lu. Zig-rir: Zigzag rwkv-in-rwkv for efficient medical image segmentation. *IEEE Transactions on Medical Imaging*, 44(8):3245–3257, 2025.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pp. 424–432. Springer, 2016.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yuchen Duan, Weiyun Wang, Zhe Chen, Xizhou Zhu, Lewei Lu, Tong Lu, Yu Qiao, Hongsheng Li, Jifeng Dai, and Wenhai Wang. Vision-rwkv: Efficient and scalable visual perception with rwkv-like architectures. In *International Conference on Learning Representations*, volume 2025, pp. 83166–83182, 2025.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024.

Anahita Fathi Kazerooni, Nastaran Khalili, Xinyang Liu, et al. BraTS-PEDs: Results of the multi-consortium international pediatric brain tumor segmentation challenge 2023. *Machine Learning for Biomedical Imaging*, 2025:72–87, 2025. doi: 10.59275/j.melba.2025-f6fg.

Lidia Garrucho, Kaisar Kushibar, Claire-Anne Reidel, Smriti Joshi, Richard Osuala, Apostolia Tsirikoglou, Maciej Bobowicz, Javier Del Riego, Alessandro Catanese, Katarzyna Gwoździewicz, et al. A large-scale multicenter breast cancer dce-mri benchmark dataset with expert segmentations. *Scientific data*, 12(1):453, 2025.

Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

Ali Hatamizadeh, Vishwesh Nath, Yucheng Tang, Dong Yang, Holger R Roth, and Daguang Xu. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *International MICCAI brainlesion workshop*, pp. 272–284. Springer, 2021.

Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *2022 IEEE/CVF winter conference on applications of computer vision (WACV)*, pp. 1748–1758. IEEE, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Nicholas Heller, Fabian Isensee, Dasha Trofimova, Resha Tejpaul, Zhongchen Zhao, Huai Chen, Lisheng Wang, Alex Golts, Daniel Khapun, Daniel Shats, Yoel Shoshan, Flora Gilboa-Solomon, Yasmeen George, Xi Yang, Jianpeng Zhang, Jing Zhang, Yong Xia, Mengran Wu, Zhiyang Liu, Ed Walczak, Sean McSweeney, Ranveer Vasdev, Chris Hornung, Rafat Solaiman, Jamee Schoephoerster, Bailey Abernathy, David Wu, Safa Abdulkadir, Ben Byun, Justice Spriggs, Griffin Struyk, Alexandra Austin, Ben Simpson, Michael Hagstrom, Sierra Virnig, John French, Nitin Venkatesh, Sarah Chan, Keenan Moore, Anna Jacobsen, Susan Austin, Mark Austin, Subodh Regmi, Nikolaos Papanikolopoulos, and Christopher Weight. The kits21 challenge: Automatic segmentation of kidneys, renal tumors, and renal cysts in corticomedullary-phase ct, 2023.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Xingru Huang, Shuanghua Ye, Zhao Huang, Wenwen Tang, Huiyu Zhou, Zhiwen Zheng, Jin Liu, and Xiaoshuai Zhang. Crown: A unified framework for anti-aliased downsampling and phase-calibrated fusion in 3d medical segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8514–8524, 2026.

Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

Yuanfeng Ji, Haotian Bai, Chongjian Ge, Jie Yang, Ye Zhu, Ruimao Zhang, Zhen Li, Lingyan Zhang, Wanling Ma, Xiang Wan, et al. Amos: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation. *Advances in neural information processing systems*, 35:36722–36732, 2022.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9992–10002. IEEE, 2021.

Zhaoyi Liu, Haichuan Zhang, and Ang Li. Beyond the birkhoff polytope: Spectral-sphere-constrained hyper-connections. *arXiv preprint arXiv:2603.20896*, 2026.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11966–11976. IEEE, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Lena Maier-Hein, Annika Reinke, Patrick Godau, et al. Metrics reloaded: recommendations for image analysis validation. *Nature Methods*, 21(2):195–212, 2024. doi: 10.1038/s41592-023-02151-z.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pp. 565–571. IEEE, 2016.

Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493:626–646, 2022.

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

OpenAI. Introducing Codex. OpenAI, May 2025. URL https://openai.com/index/introducing-codex/. Accessed: 2026-09-13.

OpenAI. GPT-5.5 System Card. OpenAI Deployment Safety Hub, April 2026a. URL https://deploymentsafety.openai.com/gpt-5-5. Accessed: 2026-09-13.

OpenAI. GPT-5.6 System Card. OpenAI Deployment Safety Hub, July 2026b. URL https://deploymentsafety.openai.com/gpt-5-6. Accessed: 2026-09-13.

OpenAI. GPT-6 Astra System Card. OpenAI Deployment Safety Hub, September 2026c. URL https://deploymentsafety.openai.com/gpt-6-astra. Accessed: 2026-09-13.

PyTorch Contributors. torch.nn.functional.scaled_dot_product_attention. https://docs.pytorch.org/docs/2.12/generated/torch.nn.functional.scaled_dot_product_attention.html, 2026. PyTorch 2.12 documentation. Accessed September 15, 2026.

Annika Reinke, Minu D. Tizabi, Michael Baumgartner, et al. Understanding metric-related pitfalls in image analysis validation. *Nature Methods*, 21(2):182–194, 2024. doi: 10.1038/s41592-023-02150-0.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Saikat Roy, Gregor Koehler, Constantin Ulrich, Michael Baumgartner, Jens Petersen, Fabian Isensee, Paul F Jaeger, and Klaus H Maier-Hein. Mednext: transformer-driven scaling of convnets for medical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pp. 405–415. Springer, 2023.

Natalia Salpea, Paraskevi Tzouveli, and Dimitrios Kollias. Medical image segmentation: A review of modern architectures. In *European Conference on Computer Vision*, pp. 691–708. Springer, 2022.

Abdelrahman Shaker, Muhammad Maaz, Hanoona Rasheed, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Unetr++: delving into efficient and accurate 3d medical image segmentation. *IEEE Transactions on Medical Imaging*, 43(9):3377–3390, 2024.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

Sam Shleifer, Jason Weston, and Myle Ott. Normformer: Improved transformer pretraining with extra normalization. *arXiv preprint arXiv:2110.09456*, 2021.

Peibo Song, Zihao Wang, Jinshuo Zhang, Shujun Fu, Yunfeng Zhang, Wei Wu, and Fangxun Bao. Su-rmt: Toward bridging semantic representation and structural detail modeling for medical image segmentation. *Information Fusion*, pp. 104182, 2026a.

Peibo Song, Xiaotian Xue, Jinshuo Zhang, Zihao Wang, Jinhua Liu, Shujun Fu, Fangxun Bao, and Si Yong Yeo. Uni-encoder meets multi-encoders: Representation before fusion for brain tumor segmentation with missing modalities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15627–15638, June 2026b.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 32–42. IEEE, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Tassilo Wald, Saikat Roy, Fabian Isensee, Constantin Ulrich, Sebastian Ziegler, Dasha Trofimova, Raphael Stock, Michael Baumgartner, Gregor Köhler, and Klaus H. Maier-Hein. Primus: Enforcing attention usage for 3d medical image segmentation. *Transactions on Machine Learning Research*, May 2026. URL https://openreview.net/forum?id=x4vZE4PDEu.

Haonan Wang, Peng Cao, Jinzhu Yang, and Osmar Zaiane. Narrowing the semantic gaps in u-net with learnable skip connections: The case of medical image segmentation. *Neural Networks*, 178: 106546, 2024.

Shufang Xie, Huishuai Zhang, Junliang Guo, Xu Tan, Jiang Bian, Hany Hassan Awadalla, Arul Menezes, Tao Qin, and Rui Yan. Residual: Transformer with dual residual connections. *Artificial Intelligence*, pp. 104602, 2026.

Yutong Xie, Jianpeng Zhang, Chunhua Shen, and Yong Xia. Cotr: Efficiently bridging cnn and transformer for 3d medical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pp. 171–180. Springer, 2021.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Kuai Yu, et al. mhc: Manifold-constrained hyper-connections. *arXiv preprint arXiv:2512.24880*, 2025.

Zhaohu Xing, Tian Ye, Yijun Yang, Guang Liu, and Lei Zhu. SegMamba: Long-range Sequential Modeling Mamba For 3D Medical Image Segmentation . In *proceedings of Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, volume LNCS 15008. Springer Nature Switzerland, October 2024.

Zhaohu Xing, Tian Ye, Yijun Yang, Du Cai, Baowen Gai, Xiao-Jian Wu, Feng Gao, and Lei Zhu. Segmamba-v2: Long-range sequential modeling mamba for general 3d medical image segmentation. *IEEE Transactions on Medical Imaging*, 2025.

Ted Zadouri, Markus Hoehnerbach, Jay Shah, Vijay Thakkar, and Tri Dao. Flashattention-4: Algorithm and kernel pipelining co-design for asymmetric hardware scaling. *Proceedings of Machine Learning and Systems*, 8:912–926, 2026.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Jintao Zhang, Jia wei, Haoxu Wang, Pengle Zhang, Xiaoming Xu, Haofeng Huang, Kai Jiang, Jianfei Chen, and Jun Zhu. Sageattention3: Microscaling fp4 attention for inference and an exploration of 8-bit training. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 53901–53931. Curran Associates, Inc., 2025a. doi: 10.52202/085713-1799. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/4db397e0f760cc573c681e81a01a3dba-Paper-Conference.pdf.

Jintao Zhang, Xiaoming Xu, Jia wei, Haofeng Huang, Pengle Zhang, Chendong Xiang, Jun Zhu, and Jianfei Chen. Sageattention2++: A more efficient implementation of sageattention2. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*, 2025b. URL https://openreview.net/forum?id=Eev5rXcgng.

Xiangdong Zhang, Xiaohan Qin, Sunan Zou, Tuo Dai, Xiaoming Shi, Huaijin Wu, Yebin Yang, Zhuo Xia, Shaofeng Zhang, Lin Yao, et al. xhc: Expanded hyper-connections. *arXiv preprint arXiv:2607.14530*, 2026a.

Xiao Zhang, Ruoxi Jiang, William Gao, Rebecca Willet, and Michael Maire. Residual connections harm generative representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39669–39679, June 2026b.

Zheyuan Zhang and Ulas Bagci. Dynamic linear transformer for 3d biomedical image segmentation. In *International Workshop on Machine Learning in Medical Imaging*, pp. 171–180. Springer, 2022.

Sihang Zhou, Dong Nie, Ehsan Adeli, Jianping Yin, Jun Lian, and Dinggang Shen. High-resolution encoder–decoder networks for low-contrast medical image segmentation. *IEEE Transactions on Image Processing*, 29:461–475, 2019.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. In *International Conference on Learning Representations*, volume 2025, pp. 97183–97219, 2025.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62429–62442. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhu24f.html.

APPENDIX

Table of contents.

- A: More Discussion on representation–structure dilemma
- B: Details of the Flumina Architecture
- C: Theoretical Analysis of Identity Residual Mapping
- D: Datasets and Preprocessing
- E: Training Configurations
- F: Inference Details and Evaluation Metrics
- G: Additional Results and Statistical Analysis
- H: Computational Cost and Inference Efficiency
- I: Additional Experiments and Analysis
- J: Limitations and Future Work

A MORE DISCUSSION ON REPRESENTATION–STRUCTURE DILEMMA

We first elaborate on the representation–structure dilemma introduced in Section 1. We examine its architectural origins, and revisit the underlying assumption carried over from the convolutional era into the Transformer era: preserving structure requires skip connections.

A.1 THE SUCCESS OF THE U-SHAPED PARADIGM IN THE CONVOLUTIONAL ERA

Typical CNNs follow a hierarchical design, using local convolutions and progressive downsampling to integrate information over increasingly large receptive fields, thereby transforming shallow structural features into deep semantic representations (Zeiler & Fergus, 2014; LeCun et al., 2015; Bau et al., 2017). The U-shaped paradigm naturally fits this design: skip connections pass multi-scale encoder features directly to the decoder. By combining shallow spatial detail with deeper semantic information, the decoder supports the recognition and precise localization required for medical image segmentation (Ronneberger et al., 2015; Çiçek et al., 2016; Wang et al., 2024).

However, skip connections can still constrain representation learning in U-Net's CNN encoder. As illustrated in Figures 8 and 6(a), removing skip connections reduces similarity outside the reference region and yields clearer class separation in t-SNE projections. Notably, the cosine-similarity differences are most pronounced in the final stage. This is consistent with the hierarchical design of CNNs, where semantic understanding is more explicit in the deeper stages. But when semantic representation is concentrated in a few deep layers, its contribution to prediction may be limited relative to that of the multi-scale structural features supplied through skip connections.

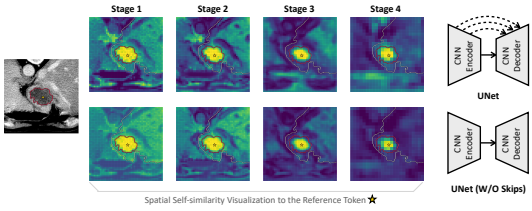

Figure 8: **Cosine similarity across four encoder stages.** *Top:* U-Net with skip connections. *Bottom:* U-Net without skip connections. Removing skip connections forms a better representation.

As shown in Table 4, removing skip connections deprives the decoder of direct access to structural detail, reducing segmentation performance. In summary, the U-shaped paradigm succeeded partly because it matched CNNs' local, hierarchical design. Skip connections preserve structural information across scales, while the strongest semantic abstraction emerges near the end of the encoder.

A.2 THE LIMITATIONS OF THE U-SHAPED PARADIGM IN THE TRANSFORMER ERA

Unlike CNNs, ViTs enable global interactions at every layer, without relying on hierarchical downsampling to expand their receptive field (Dosovitskiy et al., 2020). Semantic representations can

emerge naturally during training across the network, rather than being confined to a few deep layers. Therefore, the U-shaped paradigm's constraints on semantic representation learning can become more apparent in ViTs. This limitation appears in several architectural forms. We examine this limitation in three representative U-shaped designs:

- **Sequence-enhanced encoders.** These encoders add sequence-modeling modules to CNNs while retaining skip connections, as in TransUNet (Chen et al., 2021).
- **Hierarchical sequence encoders.** These encoders combine sequence modeling, progressive downsampling, and multi-scale skip connections, as in SwinUNETR (Hatamizadeh et al., 2021) and SegMamba (Xing et al., 2024).
- **Plain ViT encoders.** These encoders construct multi-scale features from selected ViT layers for decoder skip connections, as in UNETR (Hatamizadeh et al., 2022).

In the **sequence-enhanced** design in Figure 1, the Transformer mainly refines CNN features rather than serving as the primary representation learner. Cosine-similarity patterns change little across most Transformer blocks, suggesting that the representations remain largely inherited from the CNN encoder. Skip connections help explain this behavior: the decoder can directly access CNN features without relying on the added Transformer components.

Hierarchical sequence encoders incorporate sequence modeling across multiple stages while retaining the U-shaped skip pathways. Compared with U-Net in Figure 8, SegMamba and SwinUNETR in Figure 9 show more spatially extensive responses in early stages, consistent with earlier integration of global context. However, these broader responses are not followed by increasingly clear separation between the target and its surroundings. In Stages 3 and 4, high similarity extends over large regions outside the target, whereas U-Net's deeper responses are more localized. We propose the following interpretation of these observations. Within the U-shaped paradigm, skip connections pass intermediate features directly to the decoder, preserving access to multi-scale information. However, these pathways may also reduce the reliance on deeper semantic representations.

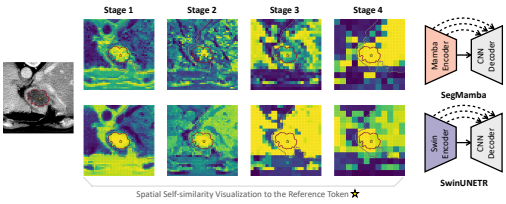

Figure 9: **Cosine similarity in hierarchical sequence encoders.** *Top:* SegMamba. *Bottom:* SwinUNETR. Spatial responses appear in early stages but remain diffuse beyond the outlined target in deeper stages.

Plain ViT encoders provide another route. Methods such as UNETR construct multi-scale features from selected Transformer layers and pass them to the decoder through skip connections. Although these encoders support global interactions at every layer, their representations may still evolve similarly to those of U-Net. In UNETR, for example, Figure 10 shows only small, gradual changes through Block 8, with more substantial changes concentrated in the final blocks. This suggests that semantic abstraction remains concentrated in the final layers, a pattern that we interpret as a possible consequence of the U-shaped skip connections.

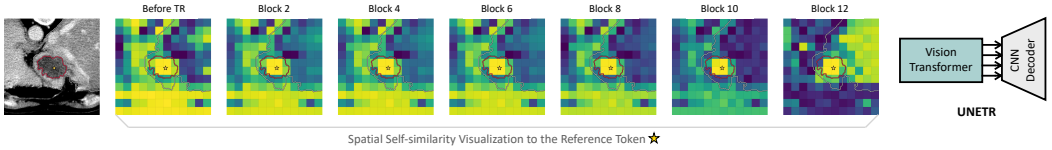

Figure 10: **Cosine similarity across UNETR's encoder.** Similarity patterns remain largely unchanged through Block 8, with more pronounced changes in the final blocks. The final response remains diffuse beyond the target region.

Skip connections give intermediate layers a direct role in supplying spatial detail to the decoder. This role may encourage these layers to retain structural information during training, leaving semantic

abstraction largely to the final blocks. Such a division of labor could explain why a plain ViT exhibits a U-Net-like progression despite supporting global interactions at every layer.

A.3 Recent Methods for Enforcing Transformer Usage

To enforce Transformer usage, Wald et al. (2026) introduce Primus (denoted Primus-V1 in this subsection) and Primus-V2. Both omit encoder-decoder skips and refine the ViT block design, although the study does not explicitly analyze the relationship between skips and Transformer usage. Their overall architectural designs broadly resemble skip-free UNETR and TransUNet, respectively.

Their Transformer-removal experiments demonstrate the contribution from the ViT, while our cosine-similarity maps (Figure 11) illustrate how its representations evolve. Primus-V2 is particularly informative: its tokenizer produces more semantically organized initial features, but this organization appears to weaken slightly before stronger semantic patterns emerge in the final blocks. Compared with U-Net's target-localized responses in Figure 8, Primus-V2's final responses also extend to regions belonging to other segmentation classes, suggesting broader anatomical associations.

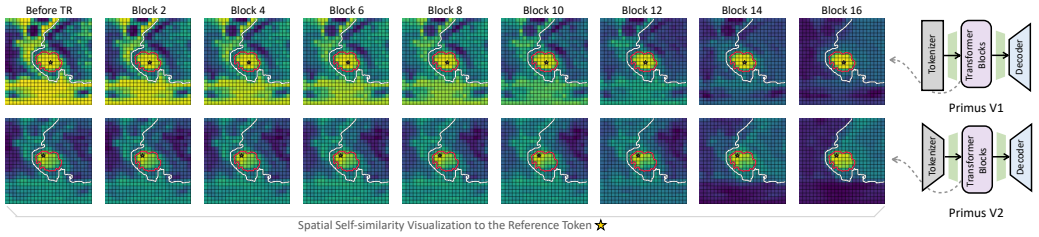

Figure 11: **Layer-wise cosine similarity in Primus-V1 and Primus-V2.** *Top:* Primus-V1. *Bottom:* Primus-V2. Maps show cosine similarity to the starred reference token in each row, before the Transformer and after every two blocks. The corresponding architectures are shown on the right.

These designs strengthen Transformer-centric segmentation, with Primus-V2 performing particularly well on multi-organ segmentation (Table 2). However, they do not consistently outperform strong convolutional baselines across diverse segmentation tasks (Table 3). From our perspective, removing skips without an alternative pathway for preserving structural detail may partly limit the translation of their semantic representations into segmentation performance.

A.4 Revisiting the Representation–Structure Dilemma in the Transformer Era

These observations point to a common architectural challenge: preserving structural detail while making the Transformer responsible for representation learning. Nevertheless, as discussed in Section 1, it is not inherent to medical image segmentation. Instead, it arises from the U-shaped paradigm, which ties structural preservation to skip connections, although dense prediction requires the former, not necessarily the latter. Motivated by this distinction, we propose a third approach: *structure as state, not skip*. Following this principle, we introduce Flumina, which maintains multi-scale features as persistent states within a shared ViT backbone.

B Details of the Flumina Architecture

Here, we supplement Section 3.2 with the implementation details shared by all four Flumina configurations. Batch dimensions are omitted for clarity.

Multi-Scale State Initialization. Each tokenizer stage contains one residual block with two $3\times3\times3$ convolutions, each followed by channel-wise LayerNorm and GELU. The four projectors use kernel sizes and strides of $(8, 4, 2, 1)$, respectively. For the i-th stage, the initial state is

$$\boldsymbol{X}_0^{(i)} = \mathrm{RMSNorm}\left(\mathcal{P}_i\left(\mathbf{F}_i\right)\right) + \mathbf{E}, \tag{7}$$

where $\mathcal{P}_i$ denotes the i-th projector, $\mathbf{F}_i$ is the feature map from the i-th tokenizer stage, and $\mathbf{E}$ is a shared learnable positional embedding. The four states originate from different feature scales and share the same $(D/8) \times (H/8) \times (W/8)$ token grid.

Details of the Read and Write Mechanisms. Each block first reads the four states into a single sequence, applies attention, and adds the gated output to each state. The FFN repeats this process on the updated states, using separate read and write parameters. Following the notation of our theoretical analysis, the block update is

$$\boldsymbol{X}_t = \boldsymbol{X}_{t-1} + \boldsymbol{\Delta}_t, \tag{8}$$

where $\boldsymbol{\Delta}_t$ collects the gated attention and FFN updates within the t-th block. Each incoming state therefore retains a direct identity path through the block.

For the read, we normalize each state independently and concatenate the results along the feature dimension. Two bias-free linear layers with a bottleneck width of $C/8$ produce the feature-wise gates. The first layer's output is divided by $k = 4$ before SiLU, and the second is followed by sigmoid. For the write, RMSNorm operates jointly over the kC features of the concatenated states. Both gates are computed from the states entering the corresponding sublayer.

3D RoPE. Attention uses 3D RoPE to encode spatial position. Within each head, query and key vectors are divided equally among the depth, height, and width axes. Each axis receives $2m$ real-valued components, where $m = d_{\text{head}}/6$ and d_{head} is the head dimension. Let q_j and k_j denote the query and key components within one axis. The r-th complex query and key entries are

$$q = q_{2r} + \mathrm{i}q_{2r+1}, \quad k = k_{2r} + \mathrm{i}k_{2r+1}, \quad r = 0, \ldots, m-1. \tag{9}$$

Let $\boldsymbol{p} = (p_d, p_h, p_w)$ and $\boldsymbol{s} = (s_d, s_h, s_w)$ denote the query and key token positions on the three-dimensional grid, with nonnegative integer coordinates. For axis $a \in \{d, h, w\}$, RoPE applies

$$q \mapsto q\, e^{\mathrm{i}p_a\theta_r}, \quad k \mapsto k\, e^{\mathrm{i}s_a\theta_r}, \quad \theta_r = 10000^{-r/m}. \tag{10}$$

In this way, each complex entry contributes to the query–key inner product as

$$\mathrm{Re}\left\langle qe^{\mathrm{i}p_a\theta_r}, ke^{\mathrm{i}s_a\theta_r}\right\rangle = \mathrm{Re}\Big[\langle q, k\rangle e^{\mathrm{i}(s_a - p_a)\theta_r}\Big], \tag{11}$$

where the complex inner product conjugates its first argument. Taking the real part of the result recovers the dot product of the two rotated real-valued feature pairs.

It is worth noticing that the phase factor $e^{\mathrm{i}(s_a - p_a)\theta_r}$ in Eq. 11 depends on the relative coordinate along axis a. By combining these phase factors across the three axes, 3D RoPE therefore makes attention sensitive to the relative spatial displacement $\boldsymbol{s} - \boldsymbol{p}$ between tokens.

Multi-Scale State Decoding. After the final block, each state is RMS-normalized and reshaped into a 3D feature map. An independent transposed convolution, using the kernel size and stride of its corresponding projector, restores each state to its tokenizer resolution. The restored channel widths are $(32, 64, 128, 256)$. The decoder starts from the coarsest state. Each stage performs trilinear upsampling, matches channels with a $1 \times 1 \times 1$ convolution, and adds the next finer restored state. A $3 \times 3 \times 3$ convolution refines the sum. Both convolutions are followed by channel-wise LayerNorm and GELU. A final $1 \times 1 \times 1$ convolution produces segmentation logits at the input resolution. Thus, the decoder receives multi-scale information entirely through the final ViT states.

C Theoretical Analysis of Identity Residual Mapping

In this section, we examine how the residual mapping affects state propagation in multi-state architectures and explain why Flumina fixes it to the identity matrix.

- How does mHC stabilize propagation across depth? (Eq. 19)
- How do HC initialization and mHC residual mixing affect state diversity? (Eq. 29)
- Why does Flumina instead use an identity residual mapping? (Eq. 31)

Residual propagation across depth. We first isolate the direct residual path from the learned branch updates. This separation allows us to characterize what $\boldsymbol{H}_t^{\text{res}}$ preserves or suppresses without making assumptions about the complete nonlinear dynamics. For notational convenience, we flatten the token and channel dimensions of each state and write

$$\boldsymbol{X}_t = \begin{bmatrix} \boldsymbol{x}_{t,1}^\top \\ \vdots \\ \boldsymbol{x}_{t,k}^\top \end{bmatrix} \in \mathbb{R}^{k \times d}, \tag{12}$$

where $\boldsymbol{x}_{t,i}$ denotes the i-th state at depth t and d denotes the flattened feature dimension. We absorb the learned branch output in Eq. 2 into

$$\boldsymbol{\Delta}_t = \boldsymbol{H}_t^{\text{post}} f_t(\boldsymbol{H}_t^{\text{pre}} \boldsymbol{X}_{t-1}), \tag{13}$$

which gives the compact recurrence

$$\boldsymbol{X}_t = \boldsymbol{H}_t^{\text{res}} \boldsymbol{X}_{t-1} + \boldsymbol{\Delta}_t. \tag{14}$$

For $b \geq a$, define the residual transition product as

$$\boldsymbol{P}_{b:a} = \boldsymbol{H}_b^{\text{res}} \boldsymbol{H}_{b-1}^{\text{res}} \cdots \boldsymbol{H}_a^{\text{res}}, \tag{15}$$

and let $\boldsymbol{P}_{b:a} = \boldsymbol{I}_k$ when $b < a$. Recursively expanding Eq. 14 over T blocks gives

$$\boldsymbol{X}_T = \underbrace{\boldsymbol{P}_{T:1} \boldsymbol{X}_0}_{\text{carried component}} + \underbrace{\sum_{t=1}^{T} \boldsymbol{P}_{T:t+1} \boldsymbol{\Delta}_t}_{\text{learned updates}}. \tag{16}$$

The first term describes how the initial states are carried through the residual path. The second collects the updates produced by the shared transformations. We focus first on the carried component, whose behavior is governed entirely by the product of the residual mappings.

Stability under doubly stochastic mixing. The manifold constraint in Eq. 3 makes each $\boldsymbol{H}_t^{\text{res}}$ nonnegative with unit row and column sums. Consequently,

$$\|\boldsymbol{H}_t^{\text{res}}\|_1 = \|\boldsymbol{H}_t^{\text{res}}\|_\infty = 1. \tag{17}$$

The induced spectral norm therefore satisfies

$$\|\boldsymbol{H}_t^{\text{res}}\|_2 \leq \sqrt{\|\boldsymbol{H}_t^{\text{res}}\|_1 \, \|\boldsymbol{H}_t^{\text{res}}\|_\infty} = 1. \tag{18}$$

By submultiplicativity,

$$\|\boldsymbol{P}_{T:1}\|_2 \leq \prod_{t=1}^{T} \|\boldsymbol{H}_t^{\text{res}}\|_2 \leq 1. \tag{19}$$

The same bound holds for $\boldsymbol{P}_{T:1}^\top$. Thus, the direct residual mapping is non-expansive in both forward and backward propagation. This rules out amplification along this path, but it does not imply that differences among the states are preserved.

Contraction of state differences. To characterize these differences, define the state diameter

$$\operatorname{diam}(\boldsymbol{X}) = \max_{i,j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2. \tag{20}$$

For a row-stochastic matrix $\boldsymbol{A} \in \mathbb{R}^{k \times k}$, its row-wise Dobrushin coefficient is

$$\tau(\boldsymbol{A}) = \frac{1}{2} \max_{p,q} \sum_{j=1}^{k} |A_{pj} - A_{qj}| = 1 - \min_{p,q} \sum_{j=1}^{k} \min\{A_{pj}, A_{qj}\}. \tag{21}$$

This coefficient controls how strongly $\boldsymbol{A}$ contracts the state diameter:

$$\operatorname{diam}(\boldsymbol{A}\boldsymbol{X}) \leq \tau(\boldsymbol{A}) \operatorname{diam}(\boldsymbol{X}). \tag{22}$$

To see this, consider two rows p and q of $\boldsymbol{A}$. Their coefficient differences sum to zero because $\boldsymbol{A}$ is row-stochastic. Separating the positive and negative coefficients therefore expresses the corresponding output difference as a scaled difference between two convex combinations of the input states. The scale is at most $\tau(\boldsymbol{A})$, while the distance between the two convex combinations is bounded by $\operatorname{diam}(\boldsymbol{X})$, yielding Eq. 22.

We now consider a sufficient condition for repeated mixing to make the states increasingly similar. Suppose there exists a layer-independent constant $0 < \delta \leq 1/k$ such that

$$[\boldsymbol{H}_t^{\text{res}}]_{ij} \geq \delta, \qquad \forall\, t, i, j. \tag{23}$$

For any two rows p and q,

$$\sum_{j=1}^{k} \min\left\{ [\boldsymbol{H}_t^{\text{res}}]_{pj}, [\boldsymbol{H}_t^{\text{res}}]_{qj} \right\} \geq k\delta. \tag{24}$$

Substituting this bound into Eq. 21 gives

$$\tau\left(\boldsymbol{H}_t^{\text{res}}\right) \leq 1 - k\delta < 1. \tag{25}$$

Applying Eq. 22 successively across depth yields

$$\begin{aligned} \operatorname{diam}\left(\boldsymbol{P}_{T:1}\boldsymbol{X}_0\right) &\leq \prod_{t=1}^{T} \tau\left(\boldsymbol{H}_t^{\text{res}}\right) \operatorname{diam}(\boldsymbol{X}_0) \\ &\leq (1-k\delta)^T \operatorname{diam}(\boldsymbol{X}_0) \xrightarrow[T\to\infty]{} 0. \end{aligned} \tag{26}$$

Because each residual mapping is also column-stochastic, the mean of the carried states remains unchanged. Let

$$\bar{\boldsymbol{x}}_0 = \frac{1}{k}\mathbf{1}_k^\top \boldsymbol{X}_0. \tag{27}$$

Then

$$\frac{1}{k}\mathbf{1}_k^\top \boldsymbol{P}_{T:1}\boldsymbol{X}_0 = \bar{\boldsymbol{x}}_0, \tag{28}$$

and Eq. 26 further gives

$$\max_i \left\| \left(\boldsymbol{P}_{T:1}\boldsymbol{X}_0\right)_i - \bar{\boldsymbol{x}}_0 \right\|_2 \leq (1-k\delta)^T \operatorname{diam}(\boldsymbol{X}_0). \tag{29}$$

Thus, under uniform positivity, repeated doubly stochastic mixing drives the carried component of each state toward the common average of the initial states. Besides, these approaches also provide no initial state diversity: both HC and mHC initialize $\boldsymbol{X}_0$ with k copies of the same state, yielding $\operatorname{diam}(\boldsymbol{X}_0) = 0$ and leaving the states without distinct roles at initialization.

Here, we clarify two points about this contraction result:

1. **Uniform positivity in mHC.** Uniform positivity is a sufficient condition rather than a property of doubly stochastic matrices in general. Nevertheless, mHC produces strictly positive residual mappings through exponentiation and Sinkhorn–Knopp normalization (Xie et al., 2025). Exponentiation makes every entry strictly positive, and the subsequent normalization preserves positivity. For any fixed finite forward pass, the smallest entry across these mappings therefore defines some $\delta > 0$.
2. **Contraction of the carried component.** Eq. 29 concerns only the carried component; the learned updates $\boldsymbol{\Delta}_t$ may introduce new differences in the full states. However, they do not alter the repeated mixing of the carried component, which attenuates the multi-scale differences in $\boldsymbol{X}_0$ and conflicts with maintaining structurally distinct states across depth.

Identity residual mapping in Flumina. Flumina instead fixes the residual mapping at every depth to

$$\boldsymbol{H}_t^{\text{res}} = \boldsymbol{I}_k, \qquad t = 1, \ldots, T. \tag{30}$$

Consequently, the residual transition product remains the identity and preserves the state diameter:

$$\boldsymbol{P}_{T:1} = \boldsymbol{I}_k, \qquad \left\|\boldsymbol{P}_{T:1}\right\|_2 = 1, \qquad \operatorname{diam}\left(\boldsymbol{P}_{T:1}\boldsymbol{X}_0\right) = \operatorname{diam}(\boldsymbol{X}_0). \tag{31}$$

Substituting this mapping into Eq. 16 gives

$$\boldsymbol{X}_T = \boldsymbol{X}_0 + \sum_{t=1}^{T} \boldsymbol{\Delta}_t. \tag{32}$$

Each initial multi-scale structural state therefore retains an unmodified, unit-gain path across depth. Interaction remains in $\boldsymbol{\Delta}_t$, where the learned read and write mappings exchange information through the shared transformation. The identity residual mapping thus separates state preservation from state interaction while avoiding residual averaging.

D DATASETS AND PREPROCESSING

To validate the effectiveness of the proposed method, we conducted experiments on 5 publicly available datasets. Descriptions of these datasets are provided below, and a summary is given in Table 6.

Table 6: **Datasets and settings used in our experiments.** Anatomical Label and Evaluation Area denote foreground classes and evaluated regions, respectively. Splits report labeled volumes in our internal partitions. Spacing (mm) and crop size follow the input-axis order.

Dataset	Dataset Configuration				Preprocessing Setting			Split		
	Modality	Region	Anatomical Label	Evaluation Area	Spacing (mm)	Normalization	Crop Size	Train	Valid	Test
AMOS-22	CT, MRI	Abdomen	15 organs	15 / 13 organs	(2.00, 0.71, 0.71)	z-score (instance)	64×160×192	252	36	72
LiTS	CT	Liver	Liver, Tumor	Liver, Tumor	(1.00, 0.77, 0.77)	z-score (dataset)	128×128×128	92	13	26
BraTS-25-GLI	MRI	Brain	NETC, SNFH, ET, RC	WT, TC, ET, NETC, SNFH, RC	(1.00, 1.00, 1.00)	z-score (instance)	128×128×128	2,014	280	578
KiTS-23	CT	Kidney	Kid., Tumor, Cyst	Kid.&Mass., Mass., Tumor	(1.00, 0.78, 0.78)	z-score (dataset)	128×128×128	342	49	98
MAMA-MIA	MRI	Breast	Tumor	Tumor	(2.00, 0.70, 0.70)	z-score (instance)	64×160×192	1,054	151	301

AMOS-22: AMOS-22 (Ji et al., 2022) is a multi-modality benchmark for abdominal multi-organ segmentation. We use 300 CT and 60 MRI scans with annotations for 15 organs. The scans are stratified by modality, with each group split into training, validation, and test sets at a 7:1:2 ratio. We train models on the combined CT and MRI training set and report results separately on the corresponding test subsets. Evaluation covers all 15 organs for CT and 13 for MRI, excluding bladder and prostate/uterus from MRI evaluation because neither is annotated in the MRI test subset.

We reorient the scans to RAS, crop them to the bounding box of the nonzero region, and apply z-score normalization to each cropped volume. The scans are then resampled to the voxel spacing in Table 6. All models use the same input crop size of $64 \times 160 \times 192$ voxels.

LiTS: LiTS (Bilic et al., 2023) is a benchmark for liver and liver-tumor segmentation in contrast-enhanced abdominal CT. We use 131 annotated scans, divided into 92 training, 13 validation, and 26 test scans, approximating a 7:1:2 ratio. Models predict liver and tumor as separate classes. For evaluation, the Liver region includes both classes, whereas the Tumor region includes only tumor.

We crop each scan to the bounding box of the nonzero region, clip intensities to $[-15, 197]$ HU, and apply dataset-level z-score normalization using the same foreground-intensity mean and standard deviation for all scans. The scans are then resampled to the voxel spacing in Table 6. All models use the same input crop size of $128 \times 128 \times 128$ voxels.

KiTS-23: KiTS-23 (Heller et al., 2023) is a benchmark for kidney and renal-mass segmentation in contrast-enhanced abdominal CT. We use 489 annotated scans, divided into 342 training, 49 validation, and 98 test scans, approximating a 7:1:2 ratio. Models predict kidney, tumor, and cyst as separate classes. These classes define three evaluation regions: Kidney & Masses includes all three classes, Masses includes tumor and cyst, and Tumor includes only tumor.

We reorient the scans to RAS, crop them to the bounding box of the nonzero region, and clip intensities to $[-58, 302]$ HU. Dataset-level z-score normalization uses the same foreground-intensity mean and standard deviation for all scans. The scans are then resampled to the voxel spacing in Table 6. All models use the same input crop size of $128 \times 128 \times 128$ voxels.

BraTS-25-GLI: BraTS-25-GLI (Aboian et al., 2025) is a benchmark for glioma segmentation in multi-parametric brain MRI before and after treatment. Each examination contains four co-registered sequences: FLAIR, contrast-enhanced T1 (T1ce), T1, and T2. Models predict four annotated subregions: non-enhancing tumor core (NETC), surrounding non-enhancing FLAIR hyperintensity (SNFH), enhancing tumor (ET), and resection cavity (RC). Evaluation covers these four subregions and two composite regions: whole tumor (WT), comprising NETC, SNFH, and ET; and tumor core (TC), comprising NETC and ET. RC is excluded from both composites. All six regions are evaluated for each test examination.

We first divide the full available cohort of 3,279 examinations into training, validation, and test sets at approximately a 7:1:2 ratio. Patients are stratified by treatment status: pre-treatment only, post-treatment only, or both. All examinations from the same patient or an identified duplicate-input

group remain in the same split. Retaining only annotated examinations then yields 2,014, 280, and 578 examinations from 1,286, 178, and 366 patients, respectively. The final counts therefore reflect both patient grouping and annotation availability.

We retain the provided co-registration and 1-mm isotropic spacing, reorient the images to RAS, and stack FLAIR, T1ce, T1, and T2 as input channels. Each channel is z-score normalized separately within each examination, using the union of nonzero voxels across channels and preserving zeros outside this region. The volumes are then cropped using the nonzero support of the raw images, with each spatial dimension extended to at least 128 voxels when needed. No additional resampling is applied. All models use the same input crop size of $128 \times 128 \times 128$ voxels.

MAMA-MIA: MAMA-MIA (Garrucho et al., 2025) is a multi-center benchmark for breast tumor segmentation in dynamic contrast-enhanced MRI. We use 1,506 examinations with expert primary-tumor annotations, taking only the first post-contrast T1-weighted image from each examination as input. The primary tumor is the single foreground class for both prediction and evaluation. Examinations are stratified by source cohort—DUKE, ISPY1, ISPY2, and NACT—and divided into 1,054 training, 151 validation, and 301 test examinations. This internal split approximates a 7:1:2 ratio within each cohort and replaces the released training/test split.

We reorient the scans to RAS, crop them to the bounding box of the nonzero region, and apply z-score normalization to each cropped volume. The scans are then resampled to the voxel spacing in Table 6. All models use the same input crop size of $64 \times 160 \times 192$ voxels.

E Training Configurations

Table 7 summarizes the training settings for all models.

Flumina training settings. We use the same training settings for all four Flumina variants. We optimize Flumina with AdamW (Loshchilov & Hutter, 2019), using a base learning rate of 3×10^{-4}, weight decay of 0.05, $(\beta_1, \beta_2) = (0.9, 0.98)$, and $\epsilon = 10^{-8}$. To stabilize early training, we linearly increase the learning rate from 10^{-5} to the base learning rate over the first 12,500 updates (5% of training). After warmup, we apply cosine decay to gradually reduce the learning rate to 10^{-6}. To limit unusually large gradients during optimization, we clip the global gradient ℓ_2 norm to 1.

Training settings for other baselines. For a fair comparison, we train all models for 250,000 iterations. All models use the same crop size within each dataset (Table 6) and a batch size of 2. For baselines other than U-Net, we follow the optimizer and learning-rate settings reported in the original papers and use the official code for missing details. When the two sources differ, we follow the paper. For U-Net, we follow nnU-Net's optimizer and learning-rate schedule (Isensee et al., 2021) and retain its global gradient ℓ_2 norm clipping threshold of 12. We also apply this clipping threshold to SwinUNETR and UNETR, whose official training implementations do not use gradient clipping. For CROWn (Huang et al., 2026), the paper specifies AdamW, but neither the paper nor the official code provides the remaining optimizer and learning-rate settings. Since CROWn is a CNN, we adopt these settings from MedNeXt (Roy et al., 2023).

Table 7: **Training hyperparameters for the 12 baselines and Flumina.** Columns correspond to models and rows to hyperparameters. The Flumina column applies to all four model variants. **Gray** cells indicate settings omitted from the original sources or added in our implementation.

Hyperparameter	U-Net	CoTr	SwinUNETR	UNETR	MedNeXt	UNETR++	SegMamba	Zig-RiR	SegMamba-V2	Primus	CROWn	Primus-V2	Flumina
Total updates	250,000	250,000	250,000	250,000	250,000	250,000	250,000	250,000	250,000	250,000	250,000	250,000	250,000
Optimizer	SGD	SGD	AdamW	AdamW	AdamW	SGD	SGD	SGD	SGD	AdamW	AdamW	AdamW	AdamW
Base learning rate	1×10^{-2}	1×10^{-2}	8×10^{-4}	1×10^{-4}	1×10^{-3}	1×10^{-2}	1×10^{-2}	1×10^{-2}	1×10^{-2}	3×10^{-4}	1×10^{-3}	3×10^{-4}	3×10^{-4}
Weight decay	3×10^{-5}	3×10^{-5}	1×10^{-5}	1×10^{-5}	3×10^{-5}	3×10^{-5}	1×10^{-5}	3×10^{-5}	3×10^{-5}	5×10^{-2}	3×10^{-5}	5×10^{-2}	5×10^{-2}
AdamW betas	–	–	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	–	–	–	–	(0.9, 0.98)	(0.9, 0.999)	(0.9, 0.98)	(0.9, 0.98)
AdamW ϵ	–	–	1×10^{-8}	1×10^{-8}	1×10^{-4}	–	–	–	–	1×10^{-8}	1×10^{-4}	1×10^{-8}	1×10^{-8}
SGD momentum	0.99	0.99	–	–	–	0.99	0.99	0.99	0.99	–	–	–	–
Nesterov	Yes	Yes	–	–	–	Yes	Yes	Yes	Yes	–	–	–	–
LR decay	Poly	Poly	Cosine	Cosine	Poly	Poly	Poly	Poly	Poly	Poly	Poly	Poly	Cosine
Warmup updates	0	0	41,666	2,500	0	0	0	0	0	12,500	0	12,500	12,500
Warmup ratio	0%	0%	16.7%	1%	0%	0%	0%	0%	0%	5%	0%	5%	5%
Warmup start LR	–	–	0	0	–	–	–	–	–	6×10^{-6}	–	6×10^{-6}	1×10^{-5}
Minimum LR	0	0	0	0	0	0	0	0	0	0	0	0	1×10^{-6}
Clip norm (ℓ_2)	12	12	12	12	12	12	12	12	12	1	12	1	1

U-Net implementation. U-Net (Ronneberger et al., 2015) was originally developed for 2D biomedical image segmentation, and 3D U-Net (Çiçek et al., 2016) extended this design to volumetric segmentation in 2016. However, the original 3D U-Net predicts a smaller spatial volume than its input (Çiçek et al., 2016), requiring corresponding adjustments to the input and target regions. To maintain a common input–output setting across models, we adopt a fixed 3D implementation based on the PlainConvUNet architecture used in nnU-Net (Isensee et al., 2021). It uses padded 3D convolutions, instance normalization, and LeakyReLU activations, and produces predictions with the same spatial dimensions as the input. We keep the architecture fixed across datasets without dataset-specific planning and refer to this implementation as U-Net throughout the paper.

F Inference Details and Evaluation Metrics

Inference details. We perform sliding-window inference on complete preprocessed test volumes, using the dataset-specific crop sizes in Table 6 with 50% overlap. Overlapping window logits are averaged in FP32 using Gaussian weights, with a standard deviation of one eighth of each output window dimension. Volumes are padded as needed, and the padding is removed before taking the voxel-wise argmax to obtain the segmentation mask.

Overlap metrics. For each foreground evaluation region defined in Appendix D, let $\boldsymbol{P}$ and $\boldsymbol{G}$ denote the predicted and reference voxel sets. We measure their overlap using DSC and IoU:

$$\mathrm{DSC} = \frac{2|\boldsymbol{P} \cap \boldsymbol{G}|}{|\boldsymbol{P}| + |\boldsymbol{G}|}, \qquad \mathrm{IoU} = \frac{|\boldsymbol{P} \cap \boldsymbol{G}|}{|\boldsymbol{P} \cup \boldsymbol{G}|}, \tag{33}$$

where $|\cdot|$ denotes the number of voxels. Both metrics are reported as percentages.

Boundary metric. We use HD95 to evaluate boundary accuracy in millimeters. For nonempty masks, let $\partial\boldsymbol{P}$ and $\partial\boldsymbol{G}$ denote the sampled surfaces in physical coordinates. The nearest-surface distance is $d(p, \partial\boldsymbol{G}) = \min_{g \in \partial\boldsymbol{G}} \|p - g\|_2$. We compute HD95 as

$$\begin{aligned} h_{95}(\boldsymbol{P}, \boldsymbol{G}) &= Q_{95}\{d(p, \partial\boldsymbol{G}) : p \in \partial\boldsymbol{P}\}, \\ \mathrm{HD95} &= \max\{h_{95}(\boldsymbol{P}, \boldsymbol{G}),\, h_{95}(\boldsymbol{G}, \boldsymbol{P})\}, \end{aligned} \tag{34}$$

where Q_{95} denotes the area-weighted 95th percentile: the smallest distance at which cumulative surface area reaches 95%. However, HD95 is undefined when either mask is empty, so these cases require an explicit scoring rule (Reinke et al., 2024; Maier-Hein et al., 2024). We adapt the spatial-diagonal penalty used in BraTS-PEDs (Fathi Kazerooni et al., 2025) to the physical extent of each evaluation volume.

Let $\boldsymbol{n}_i$ and $\boldsymbol{s}_i$ denote the voxel dimensions and voxel spacing in millimeters of the complete preprocessed volume for case i. Its physical diagonal is

$$D_i = \|\boldsymbol{n}_i \odot \boldsymbol{s}_i\|_2, \tag{35}$$

where $\odot$ denotes element-wise multiplication. We assign HD95 a value of D_i when exactly one mask is empty, and zero when both are empty. For two nonempty masks, we compute HD95 as defined above. The same D_i applies to all models evaluated on case i. We average scores over annotated cases within each region, then equally over evaluable regions.

G Additional Results and Statistical Analysis

We provide IoU and HD95 results to complement the DSC comparisons in Table 2 and Table 3. All results are averaged over three random seeds, and the four-dataset average gives equal weight to each dataset. Table 8 reports IoU on AMOS-22. All four Flumina variants achieve higher mean IoU than the baselines on both CT and MRI, demonstrating consistent gains across modalities and model sizes. Table 9 reports results on LiTS, KiTS-23, BraTS-25-GLI, and MAMA-MIA.

Flumina achieves the highest mean IoU on each dataset, and all four variants outperform the baselines in the four-dataset average. The overlap improvements therefore extend across both multi-organ and tumor segmentation tasks.

G.1 IoU Comparison Results

Table 8: **Performance comparison (IoU %) on AMOS-22.** Higher is better. Flumina variants are shaded in **green**. Best and second-best results are highlighted in **purple** and **red**, respectively.

AMOS-22 CT Test Split																	
Architectures	**Venue**	**Spleen**	**R. Kid**	**L. Kid**	**Gall.**	**Eso.**	**Liver**	**Stom.**	**Aorta**	**IVC**	**Panc.**	**RAG**	**LAG**	**Duo.**	**Blad.**	**Pros.**	**Avg.**
U-Net	MICCAI'15†	91.59	90.67	89.51	70.05	72.97	94.77	83.85	89.36	81.57	73.36	63.98	65.01	65.88	79.23	70.86	78.84
CoTr	MICCAI'21	91.07	89.52	87.97	70.84	70.23	94.40	81.25	87.65	79.32	70.69	63.82	62.86	63.25	76.03	65.44	76.96
SwinUNETR	MICCAIW'21	90.00	89.14	87.98	63.51	66.55	93.79	78.60	87.32	78.11	66.79	60.16	58.51	57.55	69.44	59.10	73.77
UNETR	WACV'22	84.11	77.53	75.13	46.29	49.17	90.20	64.22	78.34	67.10	51.29	45.95	40.79	39.57	55.18	46.72	60.77
MedNeXt	MICCAI'23	92.35	91.13	**91.09**	72.81	73.30	95.18	84.44	89.70	82.38	74.39	64.56	65.20	68.15	79.42	72.89	79.80
UNETR++	TMI'24	88.15	87.09	85.95	57.23	59.30	92.63	74.94	84.99	74.56	63.90	54.31	52.53	52.93	64.95	58.19	70.11
SegMamba	MICCAI'24	92.06	91.08	89.75	70.19	72.23	94.87	82.65	89.16	81.52	73.24	64.01	64.40	65.76	78.89	70.20	78.67
Zig-RiR	TMI'25	86.86	82.48	81.02	58.56	61.25	91.83	70.16	83.55	67.48	58.17	50.61	50.09	50.05	67.56	55.97	67.71
SegMamba-V2	TMI'25	92.42	91.05	90.00	72.96	73.04	95.20	84.44	89.70	82.14	74.07	65.76	65.25	66.67	79.90	69.67	79.49
Primus	TMLR'26	93.02	90.39	89.09	69.66	70.45	95.04	83.46	88.74	81.15	73.40	61.23	61.70	66.06	83.04	73.24	78.65
CROWn	CVPR'26	91.35	90.31	89.47	68.18	70.03	94.45	81.01	88.43	80.43	71.25	62.53	62.17	63.55	75.80	68.07	77.14
Primus-V2	TMLR'26	**93.61**	90.66	89.99	72.22	73.28	95.16	84.46	89.53	82.43	74.48	62.48	63.12	69.21	84.44	73.74	79.92
Flumina-S	Ours	93.16	91.74	90.41	74.82	75.01	95.52	85.01	90.46	82.98	75.27	66.49	66.57	**69.50**	83.07	74.16	80.94
Flumina-B	Ours	93.27	91.90	90.52	**75.88**	**75.39**	95.62	85.20	**90.66**	83.28	**75.91**	**67.28**	**67.30**	**69.88**	**85.10**	**74.69**	**81.46**
Flumina-M	Ours	**93.85**	**92.33**	91.00	75.24	74.97	**95.63**	**85.42**	90.44	**83.33**	**75.96**	67.27	67.04	69.41	84.89	74.22	81.40
Flumina-L	Ours	93.37	**92.10**	**91.01**	**76.09**	**75.52**	**95.69**	**85.66**	**90.71**	**83.34**	75.81	**67.32**	**67.31**	69.33	**85.77**	**75.28**	**81.62**
AMOS-22 MRI Test Split																	
Architectures	**Venue**	**Spleen**	**R. Kid**	**L. Kid**	**Gall.**	**Eso.**	**Liver**	**Stom.**	**Aorta**	**IVC**	**Panc.**	**RAG**	**LAG**	**Duo.**	**Blad.**	**Pros.**	**Avg.**
U-Net	MICCAI'15	93.18	92.52	92.24	62.82	61.16	94.93	79.79	86.13	83.07	74.51	45.68	38.42	53.25	–	–	73.67
CoTr	MICCAI'21	90.52	90.43	90.31	66.66	57.56	94.67	77.18	84.52	81.02	70.44	45.34	37.69	51.26	–	–	72.12
SwinUNETR	MICCAIW'21	90.76	90.09	90.84	56.72	53.54	94.18	72.63	84.85	80.25	66.37	43.55	33.15	45.89	–	–	69.45
UNETR	WACV'22	83.42	82.77	83.27	39.51	38.68	91.11	60.25	75.51	67.42	49.24	32.53	21.05	29.56	–	–	58.02
MedNeXt	MICCAI'23	93.25	91.52	90.74	64.09	62.40	95.47	80.92	86.43	83.53	75.52	46.18	38.35	54.33	–	–	74.06
UNETR++	TMI'24	88.83	88.06	89.66	53.84	48.98	93.66	71.51	84.00	76.88	66.68	39.54	32.93	41.56	–	–	67.39
SegMamba	MICCAI'24	91.46	92.31	91.73	63.06	61.35	95.43	79.61	86.86	83.28	73.15	47.07	38.83	52.78	–	–	73.61
Zig-RiR	TMI'25	78.30	80.72	82.85	47.99	48.29	91.42	54.62	79.42	68.67	56.60	32.02	24.40	35.82	–	–	60.09
SegMamba-V2	TMI'25	93.75	92.69	92.39	63.71	60.71	95.37	80.40	86.73	83.21	74.73	47.60	39.31	54.71	–	–	74.25
Primus	TMLR'26	93.61	92.29	91.81	65.67	59.49	95.65	81.80	86.65	83.10	74.58	47.54	37.14	55.17	–	–	74.19
CROWn	CVPR'26	91.32	91.80	91.42	63.18	57.52	94.74	77.59	85.87	81.41	70.56	45.61	36.83	51.04	–	–	72.22
Primus-V2	TMLR'26	92.49	92.71	91.90	67.31	62.71	95.75	**82.95**	87.11	83.81	76.64	47.58	40.00	**57.17**	–	–	75.24
Flumina-S	Ours	94.29	92.98	92.84	66.37	62.97	95.90	81.64	87.89	84.54	75.64	48.46	**42.08**	55.34	–	–	75.46
Flumina-B	Ours	94.57	**93.27**	92.97	**75.66**	**63.77**	**96.07**	82.74	**88.00**	84.53	76.64	**49.69**	41.58	55.75	–	–	**76.56**
Flumina-M	Ours	**94.61**	**93.33**	**92.98**	**73.84**	63.52	**96.08**	82.29	87.68	**84.78**	**76.75**	**48.73**	40.32	**56.06**	–	–	**76.23**
Flumina-L	Ours	**94.62**	91.54	**93.06**	66.05	**64.02**	96.05	**82.87**	**88.18**	**84.82**	**76.82**	48.64	**41.81**	54.92	–	–	75.65

Note. † Venue refers to the original U-Net paper. We evaluate a fixed 3D variant; implementation details are provided in Appendix E.

Table 9: **Performance comparison (IoU %) across multiple datasets.** Higher is better. Flumina variants are shaded in **green**. Best and second-best results are highlighted in **purple** and **red**, respectively. Params (millions) and MACs (trillions) are reported for a 128^3 input crop.

Architectures	**Complexity**		**LiTS**			**KiTS-23**				**BraTS-25-GLI**							**MAMA-MIA**	**Avg.**
	Params	**MACs**	**Liver**	**Tumor**	**Mean**	**Kid.&Mass.**	**Mass.**	**Tumor**	**Mean**	**WT**	**TC**	**ET**	**NETC**	**SNFH**	**RC**	**Mean**	**Tumor**	
U-Net	90.31	1.91	91.86	54.83	73.35	93.32	69.09	63.48	75.29	84.28	73.16	71.74	67.73	78.82	72.71	74.74	63.12	71.63
CoTr	41.9	0.70	91.56	56.23	73.89	92.75	66.95	63.07	74.25	84.13	72.99	70.99	68.25	78.54	72.87	74.63	62.58	71.34
SwinUNETR	62.2	0.77	90.68	54.33	72.51	93.26	66.64	61.56	73.82	84.03	73.63	72.13	**71.11**	78.64	71.92	75.24	63.69	71.31
UNETR	102.5	0.18	87.51	45.56	66.54	91.04	57.86	53.01	67.30	81.41	68.23	66.94	63.26	75.45	67.11	70.40	58.23	65.62
MedNeXt	10.5	0.17	**92.89**	61.54	77.21	93.69	70.63	66.44	76.92	84.32	74.34	72.75	**70.34**	78.90	72.51	75.53	65.77	73.86
UNETR++	42.6	0.07	83.89	38.35	61.12	91.53	61.95	58.56	70.68	83.06	70.82	69.13	66.68	77.23	70.24	72.86	53.27	64.48
SegMamba	67.4	1.48	91.85	54.53	73.19	93.34	68.50	63.39	75.08	84.00	74.08	72.67	68.15	78.67	71.51	74.85	61.15	71.07
Zig-RiR	33.7	0.21	89.80	49.65	69.73	89.78	56.17	51.37	65.77	78.84	67.81	64.84	61.78	72.02	65.63	68.49	60.50	66.12
SegMamba-V2	138.8	1.87	90.57	56.36	73.46	93.22	68.65	63.83	75.23	83.89	73.02	71.66	68.22	78.25	71.28	74.39	64.09	71.79
Primus	150.4	1.07	91.64	55.50	73.57	93.31	68.97	66.80	76.36	83.23	71.03	68.88	66.88	77.51	71.98	73.25	62.92	71.53
CROWn	23.8	0.48	92.15	55.75	73.95	93.12	66.73	61.88	73.91	83.08	71.28	70.16	67.12	77.39	69.94	73.16	64.47	71.37
Primus-V2	149.6	1.15	92.30	62.44	77.37	93.15	71.71	68.93	77.93	83.61	73.24	70.71	66.59	77.79	**74.00**	74.32	65.22	73.71
Flumina-S	27.3	0.42	92.51	64.13	78.32	94.09	72.78	69.50	78.79	84.43	74.60	73.15	68.71	79.14	73.66	75.62	66.50	74.81
Flumina-B	66.1	0.69	92.60	66.23	79.41	94.44	73.25	**70.39**	**79.36**	**84.58**	75.05	73.81	68.81	79.29	73.66	**75.87**	66.52	75.29
Flumina-M	147.9	1.15	**92.63**	**67.56**	**80.09**	**94.63**	**73.31**	69.79	79.25	84.57	**75.18**	**73.86**	67.80	**79.39**	**73.82**	75.77	**66.86**	**75.49**
Flumina-L	213.5	1.56	92.36	**67.99**	**80.17**	**94.56**	**73.82**	**70.83**	**79.74**	**84.91**	**75.40**	**73.94**	68.70	**79.58**	73.48	**76.00**	**67.13**	**75.76**

G.2 HD95 Comparison Results

We next examine whether the overlap gains are accompanied by lower boundary error. In Table 10, Flumina-L and Flumina-B achieve the lowest mean HD95 on AMOS-22 CT and MRI, respectively, consistent with their IoU results. Table 11 further shows that the four-dataset mean HD95 decreases as Flumina scales from S to L. Flumina-L reaches 22.78 mm, compared with 33.07 mm for the strongest baseline, Primus-V2.

Flumina-L also achieves the highest four-dataset mean IoU. The agreement between these metrics shows that scaling improves region overlap together with boundary accuracy on average. However, this aggregate trend does not imply a uniform benefit from larger models: smaller variants remain preferable for some datasets and modalities. The results thus support scaling as a means of improving performance across tasks, while retaining the importance of task-specific model selection.

Table 10: **Performance comparison (HD95 mm) on AMOS-22.** Lower is better. Flumina variants are shaded in **green**. Best and second-best results are highlighted in **purple** and **red**, respectively.

AMOS-22 CT Test Split																	
Architectures	**Venue**	**Spleen**	**R. Kid**	**L. Kid**	**Gall.**	**Eso.**	**Liver**	**Stom.**	**Aorta**	**IVC**	**Panc.**	**RAG**	**LAG**	**Duo.**	**Blad.**	**Pros.**	**Avg.**
U-Net	MICCAI'15†	33.31	4.57	31.49	77.25	7.44	5.74	18.31	3.21	5.04	9.69	3.48	14.33	12.28	24.06	44.83	19.67
CoTr	MICCAI'21	23.10	14.11	26.70	63.41	7.88	9.68	21.59	4.85	6.77	11.26	3.41	13.48	19.79	17.97	52.41	19.76
SwinUNETR	MICCAIW'21	24.48	12.63	29.93	100.42	8.72	10.93	29.64	7.70	11.45	16.52	8.25	7.23	18.79	38.91	77.48	26.87
UNETR	WACV'22	55.60	58.42	91.21	123.05	16.15	22.26	53.56	15.42	19.33	29.65	15.01	32.04	34.96	50.56	87.70	46.99
MedNeXt	MICCAI'23	17.33	6.00	11.45	73.52	6.84	4.08	14.80	3.01	3.71	6.91	3.26	8.45	12.63	21.75	29.30	14.87
UNETR++	TMI'24	30.15	13.78	31.72	97.23	8.75	13.34	29.12	5.07	6.80	12.36	8.83	8.11	16.98	34.61	53.09	24.66
SegMamba	MICCAI'24	18.96	10.90	24.62	79.28	6.57	8.39	20.04	5.78	4.00	7.53	3.33	4.31	12.33	19.74	50.45	18.42
Zig-RiR	TMI'25	43.32	43.09	56.16	108.89	11.05	16.78	39.77	9.27	11.41	17.92	16.91	12.92	22.63	42.32	81.25	35.58
SegMamba-V2	TMI'25	17.14	7.35	25.82	65.45	7.00	4.37	18.03	6.47	6.28	7.27	3.05	5.81	11.86	16.90	57.53	17.36
Primus	TMLR'26	7.75	6.48	16.66	79.34	6.09	3.76	10.71	2.98	3.36	6.43	3.02	3.87	10.41	5.82	31.91	13.24
CROWn	CVPR'26	22.91	7.56	27.62	81.88	7.30	8.84	24.08	4.92	4.21	12.55	10.96	6.71	16.03	32.53	51.08	21.28
Primus-V2	TMLR'26	1.73	4.92	13.03	72.97	5.71	4.22	9.20	2.73	2.97	6.11	2.67	3.41	9.88	3.96	33.80	11.82
Flumina-S	Ours	12.19	5.38	15.77	57.01	5.67	4.73	9.36	2.36	3.10	6.32	2.78	3.76	9.83	7.22	31.71	11.81
Flumina-B	Ours	10.92	5.36	16.23	60.33	5.53	3.94	9.96	2.07	3.09	5.99	2.69	8.04	9.51	5.29	31.98	12.06
Flumina-M	Ours	6.33	5.17	15.68	65.72	5.68	3.62	9.52	2.29	3.14	6.00	2.65	3.74	9.61	4.93	37.10	12.08
Flumina-L	Ours	10.89	4.89	15.63	53.90	5.76	4.39	8.72	2.62	3.09	6.30	2.67	3.58	10.12	3.77	32.07	11.23
AMOS-22 MRI Test Split																	
Architectures	**Venue**	**Spleen**	**R. Kid**	**L. Kid**	**Gall.**	**Eso.**	**Liver**	**Stom.**	**Aorta**	**IVC**	**Panc.**	**RAG**	**LAG**	**Duo.**	**Blad.**	**Pros.**	**Avg.**
U-Net	MICCAI'15	11.96	2.36	6.27	40.83	15.00	6.08	10.62	9.21	3.13	5.20	5.47	13.24	15.65	–	–	11.16
CoTr	MICCAI'21	20.93	11.81	15.23	39.36	12.22	10.76	17.36	8.86	4.24	15.32	7.12	13.23	16.34	–	–	14.83
SwinUNETR	MICCAIW'21	39.23	14.47	12.54	61.24	29.91	12.63	19.33	17.72	3.33	15.65	6.65	13.74	14.94	–	–	20.11
UNETR	WACV'22	68.18	60.08	51.39	83.71	31.56	39.23	38.08	36.87	29.73	25.62	12.81	40.95	33.63	–	–	42.45
MedNeXt	MICCAI'23	3.62	16.96	3.19	40.55	9.56	2.35	9.11	6.20	2.32	5.52	5.71	11.97	14.24	–	–	10.10
UNETR++	TMI'24	18.54	12.14	17.56	55.18	12.26	4.81	14.05	6.84	4.09	7.94	7.12	14.75	17.97	–	–	14.87
SegMamba	MICCAI'24	17.79	13.79	5.50	54.30	19.92	4.69	10.37	15.88	6.40	5.77	5.45	12.14	13.49	–	–	14.27
Zig-RiR	TMI'25	88.15	33.57	46.60	66.32	27.18	28.21	34.50	23.63	10.02	24.83	11.33	35.14	16.57	–	–	34.31
SegMamba-V2	TMI'25	2.60	2.46	5.40	39.30	13.55	3.29	16.92	6.65	2.56	4.51	13.10	11.91	13.56	–	–	10.45
Primus	TMLR'26	1.54	2.45	2.44	51.88	9.67	2.23	7.99	7.01	2.48	5.30	5.61	10.21	11.04	–	–	9.22
CROWn	CVPR'26	13.41	12.85	5.46	39.32	10.22	14.17	16.37	12.76	6.91	7.79	5.51	12.02	15.38	–	–	13.24
Primus-V2	TMLR'26	3.73	8.90	2.50	52.38	9.10	2.14	7.71	7.02	2.78	5.07	5.03	9.55	12.53	–	–	9.88
Flumina-S	Ours	2.61	3.45	2.33	53.08	9.11	2.14	9.03	7.10	2.52	5.38	5.18	10.14	13.69	–	–	9.68
Flumina-B	Ours	1.37	2.61	2.41	8.85	8.94	2.03	8.97	7.29	2.48	5.17	4.99	10.50	13.46	–	–	6.08
Flumina-M	Ours	1.35	2.53	2.38	22.86	9.13	2.03	10.81	7.77	2.32	5.21	5.10	11.95	14.14	–	–	7.51
Flumina-L	Ours	1.39	4.70	2.41	52.71	9.01	2.04	8.45	6.94	2.57	5.17	5.37	11.42	13.76	–	–	9.69

Note. † Venue refers to the original U-Net paper. We evaluate a fixed 3D variant; implementation details are provided in Appendix E.

Table 11: **Performance comparison (HD95 mm) across multiple datasets.** Lower is better. Flumina variants are shaded in **green**. Best and second-best results are highlighted in **purple** and **red**, respectively. Params (millions) and MACs (trillions) are reported for a 128^3 input crop.

Architectures	**Complexity**		**LiTS**			**KiTS-23**				**BraTS-25-GLI**							**MAMA-MIA**	**Avg.**
	Params	**MACs**	**Liver**	**Tumor**	**Mean**	**Kid.&Mass.**	**Mass.**	**Tumor**	**Mean**	**WT**	**TC**	**ET**	**NETC**	**SNFH**	**RC**	**Mean**	**Tumor**	
U-Net	90.31	1.91	45.75	145.82	95.78	18.37	51.90	57.08	42.45	6.22	24.40	24.79	30.93	5.93	32.24	20.75	54.92	53.48
CoTr	41.9	0.70	20.80	107.35	64.07	23.39	59.39	62.20	48.32	5.76	23.12	23.50	25.82	6.22	32.66	19.51	58.28	47.55
SwinUNETR	62.2	0.77	20.93	138.33	79.63	17.08	55.86	68.69	47.21	6.26	21.43	22.10	22.21	6.41	35.10	18.92	48.22	48.49
UNETR	102.5	0.18	50.85	165.46	108.15	40.75	102.50	108.20	83.82	9.83	28.36	28.02	32.99	9.16	43.30	25.28	63.90	70.29
MedNeXt	10.5	0.17	17.78	71.58	44.68	18.61	48.79	58.25	41.88	6.78	20.76	21.04	24.19	6.93	34.56	19.04	40.96	36.64
UNETR++	42.6	0.07	54.07	169.62	111.84	29.52	76.42	81.49	62.48	7.67	26.20	27.04	29.88	8.31	37.87	22.83	74.11	67.82
SegMamba	67.4	1.48	24.53	119.42	71.97	17.60	52.25	50.13	40.00	7.04	21.46	22.16	27.39	6.74	36.47	20.21	66.04	49.55
Zig-RiR	33.7	0.21	31.37	162.36	96.87	41.87	105.07	110.02	85.65	9.28	24.43	24.12	26.49	9.87	41.89	22.68	46.55	62.94
SegMamba-V2	138.8	1.87	25.02	124.66	74.84	18.10	57.90	61.39	45.80	5.99	22.98	23.10	27.64	6.51	36.94	20.53	46.97	47.03
Primus	150.4	1.07	14.95	91.20	53.07	10.91	39.15	43.88	31.31	6.07	24.00	24.63	25.38	6.58	33.14	19.97	47.75	38.02
CROWn	23.8	0.48	21.75	102.85	62.30	26.95	60.84	63.79	50.53	7.58	23.88	23.68	28.55	7.22	37.32	21.37	41.93	44.03
Primus-V2	149.6	1.15	11.16	70.87	41.01	13.57	40.01	48.14	33.91	5.80	20.69	21.14	26.85	6.15	30.98	18.60	38.75	33.07
Flumina-S	27.3	0.42	10.10	51.16	30.63	11.84	33.99	56.03	33.95	6.44	19.70	19.57	24.22	6.59	31.55	18.01	31.76	28.59
Flumina-B	66.1	0.69	10.70	41.87	26.29	7.90	35.27	53.86	32.34	6.33	18.44	18.00	23.57	6.51	32.41	17.54	32.56	27.18
Flumina-M	147.9	1.15	10.94	21.10	16.02	7.10	29.63	63.10	33.28	6.59	18.78	18.49	28.12	6.52	32.80	18.55	31.22	24.77
Flumina-L	213.5	1.56	12.14	18.03	15.09	7.00	31.16	52.16	30.11	6.18	17.45	18.00	24.85	6.46	33.38	17.72	28.20	22.78

G.3 Qualitative Results

We provide qualitative comparisons in Figure 12 and Figure 13 to examine how the quantitative gains appear in individual predictions. Across the five datasets, the 2D overlays show agreement with the reference annotations, while the 3D renderings reveal the extent and spatial organization of the predicted regions.

In general, Flumina's advantage is most apparent where baselines introduce spurious foreground or miss substantial parts of the target. In the LiTS example, Flumina preserves the liver segmentation while avoiding the large false-positive tumor regions produced by several baselines. The MAMA-MIA example highlights a different challenge: some baselines substantially underestimate the tumor extent, whereas Flumina more faithfully recovers the annotated region. These improvements in target coverage and false-positive suppression are consistent with the higher IoU and lower HD95 reported in Table 9 and Table 11.

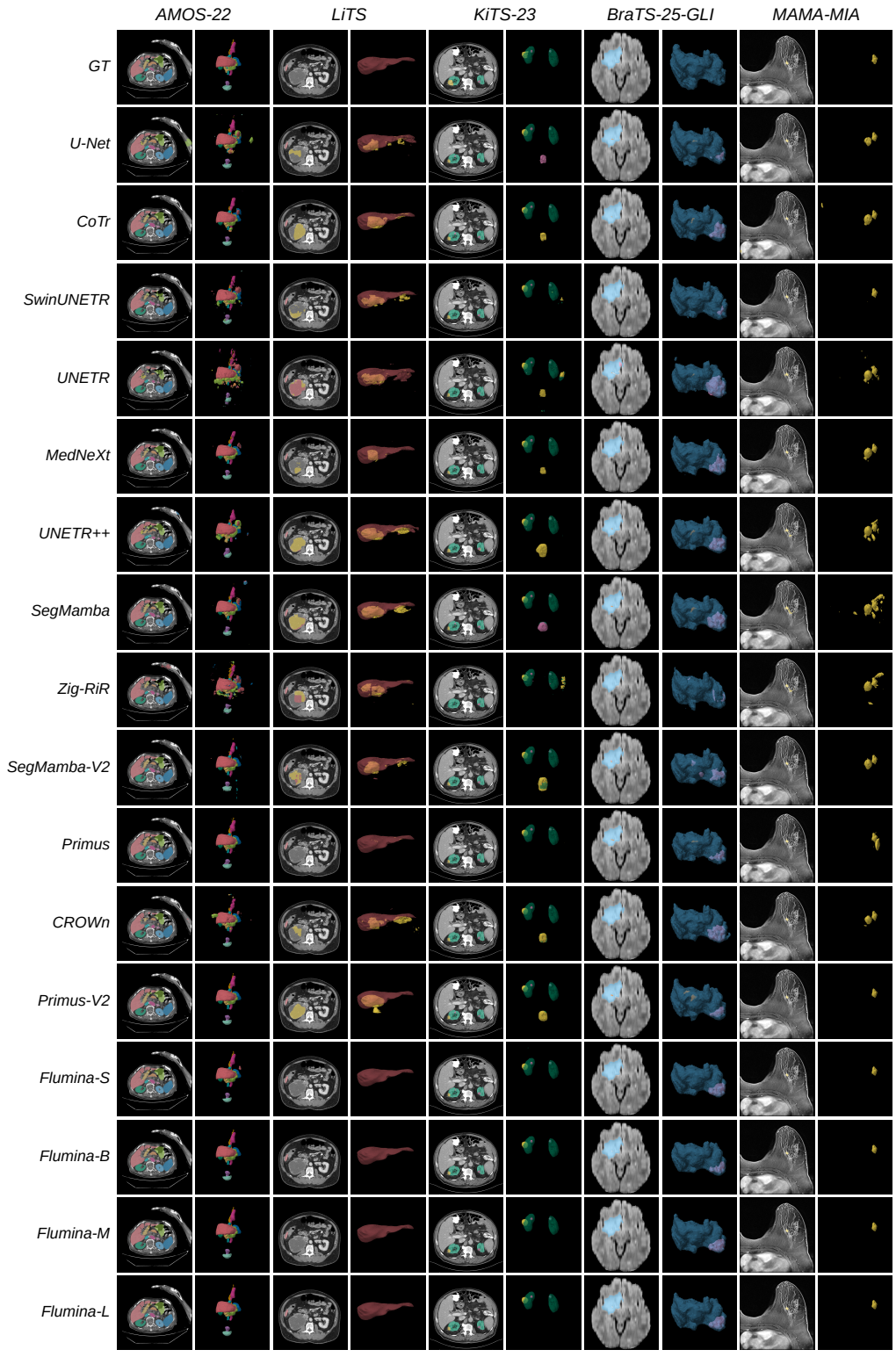

Figure 12: **Qualitative segmentation comparisons across five datasets (I).** Each dataset pairs a 2D segmentation overlay (left) with a 3D rendering (right). The top row shows the ground truth (GT), followed by predictions from the 12 baselines and four Flumina variants. Colors indicate segmentation classes within each dataset.

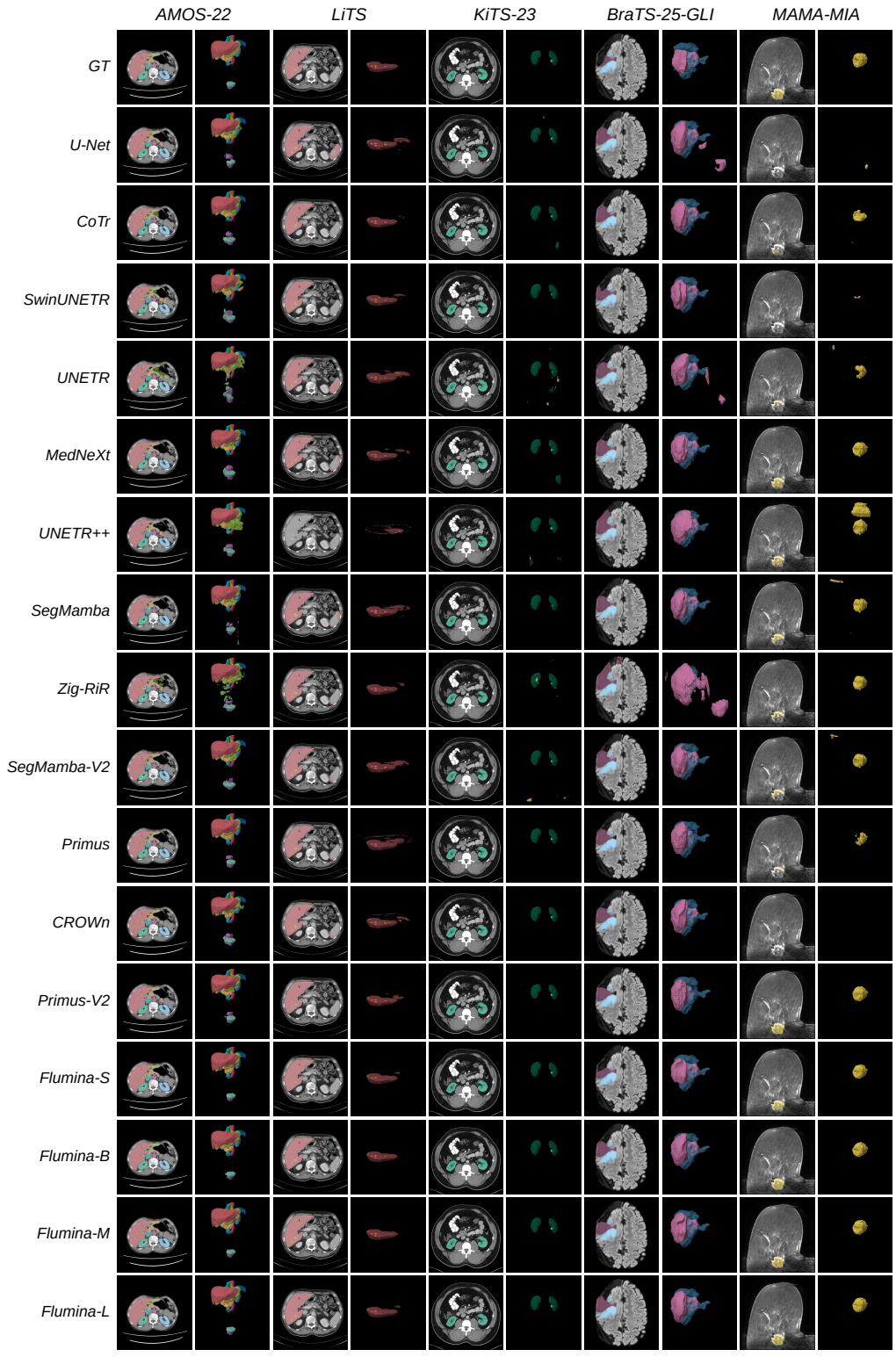

Figure 13: **Qualitative segmentation comparisons across five datasets (II).** Each dataset pairs a 2D segmentation overlay (left) with a 3D rendering (right). The top row shows the ground truth (GT), followed by predictions from the 12 baselines and four Flumina variants. Colors indicate segmentation classes within each dataset.

G.4 Comparison with nnU-Net

As discussed in Section 4, our main comparisons in Tables 2 and 3 keep architectures fixed across datasets, using the modern 3D PlainConvUNet described in Appendix E as the U-Net baseline. Here, we extend the comparison to nnU-Net (Isensee et al., 2021) with dataset-specific planning. We run the default planner on the preprocessed training and validation subsets of our fixed splits, and Table 12 lists the resulting 3D full-resolution configurations.

Table 12: **nnU-Net configurations and performance comparison (DSC %).** Spacing (mm) and crop size follow the input-axis order. , , and indicate the first-, second-, and third-best results, respectively.

Datasets	nnU-Net Configuration						Performance Comparison (DSC)				
	Spacing (mm)	Normalization	Crop Size	Kernel Sizes	Params (M)	MACs (T)	nnU-Net	Flumina-S	Flumina-B	Flumina-M	Flumina-L
AMOS-22-CT	(2.00, 0.71, 0.71)	z-score	64×160×192	[[1,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3]]	31.12	0.53	87.64	88.18	88.55	88.50	**88.66**
AMOS-22-MRI	(2.00, 0.71, 0.71)	z-score	64×160×192	[[1,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3]]	31.12	0.53	83.77	83.59	**84.56**	84.26	83.65
LiTS	(1.00, 0.77, 0.77)	CT	128×128×128	[[3,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3]]	31.20	0.48	80.23	84.89	85.69	86.40	**86.62**
BraTS-25-GLI	(1.00, 1.00, 1.00)	z-score	112×160×128	[[3,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3]]	30.79	0.53	80.94	81.80	82.05	81.88	**82.12**
KiTS-23	(1.00, 0.78, 0.78)	CT	112×128×160	[[3,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3]]	30.79	0.52	82.17	85.57	85.82	85.81	**86.25**
MAMA-MIA	(2.00, 0.70, 0.70)	z-score	56×192×192	[[1,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3],[3,3,3]]	30.70	0.55	74.20	76.31	76.45	76.55	**76.83**
Average	–						81.49	83.39	83.85	83.90	**84.02**

All plans share six encoder stages, channel widths $(32, 64, 128, 256, 320, 320)$, and two convolutions per stage. The first stage preserves resolution; later stages downsample in their first convolution. Kernels and downsampling strides default to $(3, 3, 3)$ and $(2, 2, 2)$, respectively. We follow nnU-Net's default training recipe: 1,000 epochs with 250 iterations per epoch and a batch size of 2, matching the main training budget. Optimization uses Nesterov SGD with an initial learning rate of 0.01, momentum 0.99, weight decay 3×10^{-5}, and polynomial learning-rate decay. We retain the compound Dice and cross-entropy loss.

As shown in Table 12, Flumina achieves competitive segmentation performance compared with nnU-Net. All four variants attain higher average DSC, with the largest gains on KiTS-23 and LiTS.

G.5 Details of Statistical Analysis

We detail the statistical significance tests reported above. For each method and case, we average DSC across three seed-specific checkpoints within each evaluation region, then across regions to obtain one value per case. We compare these paired values using a two-sided Wilcoxon signed-rank test. The tests in Tables 2 and 3 use the AMOS-22 test split and the pooled test splits of the other four datasets, respectively. To extend these comparisons to the full benchmark, we pool test cases from all five datasets and test DSC, IoU, and HD95 separately using the same case-level aggregation and paired test. All four Flumina variants achieve statistically significant improvements over each of the 12 baselines on all three metrics ($p < 0.05$).

H Computational Cost and Inference Efficiency

H.1 Complexity Analysis of the Transformer Block

As discussed in Section 3.2, although Flumina maintains four states within its ViT, the multi-state modeling design keeps the additional computational cost modest. More specifically, let L denote the number of tokens and C the embedding width. Flumina maintains $k = 4$ states, but reads them into one length-L sequence before each shared transformation. We now count the dominant matrix operations per sample, treating one multiply–accumulate as one MAC. For each read, two linear projections with dimensions $kC \to C/8 \to kC$ predict the feature-wise gates. Their combined cost is

$$\mathcal{M}_{\text{read}} = L \cdot kC \cdot \frac{C}{8} + L \cdot \frac{C}{8} \cdot kC = \frac{k}{4} LC^2. \tag{36}$$

Each write predicts token-wise gates through a single projection from kC to k features, requiring

$$\mathcal{M}_{\text{write}} = L \cdot kC \cdot k = k^2 LC. \tag{37}$$

Attention and SwiGLU each use a separate read and write, so both costs occur twice per block. The total is therefore

$$\begin{aligned}\mathcal{M}_{\text{block}} &= 4LC^2 + 2L^2C + 8LC^2 + 2 \cdot \left(\frac{k}{4}LC^2\right) + 2 \cdot \left(k^2LC\right) \\ &= \underbrace{4LC^2 + 2L^2C}_{\text{Attention}} + \underbrace{8LC^2}_{\text{FFN}} + \underbrace{\frac{k}{2}LC^2}_{\text{Read}} + \underbrace{2k^2LC}_{\text{Write}}.\end{aligned} \tag{38}$$

The attention term includes the query, key, value, and output projections and both attention matrix products. The SwiGLU term accounts for three projections with hidden width $8C/3$. Eq. 38 excludes normalization, nonlinearities, softmax, RoPE, and other element-wise operations; tokenizer, projector, and decoder operations are additional costs outside the backbone.

The four states contain kLC elements per sample, while attention still acts on L, rather than kL, tokens. Thus, maintaining distinct states does not multiply the quadratic attention term by k^2. For fixed $k = 4$, the block retains the $\mathcal{O}(LC^2 + L^2C)$ arithmetic complexity of a standard Transformer block. For a concrete example, consider Flumina-B with a 128^3 input, which gives $L = 4096$ and $C = 480$. Under the above counting convention, each block requires approximately 29.381 billion MACs. The two read mechanisms contribute 1.887 billion MACs, accounting for 6.42% of the block total. The two write mechanisms contribute 0.063 billion MACs, accounting for 0.21%.

H.2 Efficient Attention Kernels

Recall that Flumina uses stride-8 tokenization to retain more spatial detail in the initial states and leave more contextual modeling to the Transformer (Section 3.2). For a 128^3 crop, this increases the sequence length from 512 tokens at stride 16 to $L = 4096$. The longer sequence increases the arithmetic cost of attention, but its practical cost also depends on memory traffic and parallel execution. Here, Flumina can benefit from efficient attention kernels developed for LLMs (Dao et al., 2022; Zhang et al., 2025b;a; Zadouri et al., 2026; PyTorch Contributors, 2026), where 4096 tokens is already a relatively short sequence (Hsieh et al., 2024).

To examine how attention acceleration translates into faster Flumina inference, we compare SDPA (PyTorch Contributors, 2026), FlashAttention-4 (Zadouri et al., 2026), SageAttention2++ (Zhang et al., 2025b), and SageAttention3 (Zhang et al., 2025a). We benchmark these implementations on BraTS-25-GLI using a consumer NVIDIA GeForce RTX 5090 GPU, batch size 1, and BF16 AMP. Vanilla, an attention implementation without optimizations, serves as the reference. For each checkpoint, we keep the input crops, sliding-window settings, and model compilation settings fixed across implementations. Table 13 compares attention implementations' latency, preliminary crop timings and throughput, complete-case latency, and segmentation performance.

Table 13: **Attention kernel comparison on BraTS-25-GLI.** Speedups and score changes (percentage points) are relative to Vanilla. , , and indicate the first-, second-, and third-best efficiency results within each model size, respectively.

Model	Attention	Attention Forward		Crop Inference			Case Inference		Segmentation Performance			
		Time (ms) ↓	Speedup ↑	s/crop ↓	crops/s ↑	Speedup ↑	s/case ↓	Speedup ↑	DSC (%)	ΔDSC (%)	IoU (%)	ΔIoU (%)
Flumina-S	Vanilla	18.996	1.00×	0.0396	25.23	1.00×	0.287	1.00×	81.804	+0.000	75.615	+0.000
	SDPA	1.725	11.01×	0.0221	45.24	1.79×	0.155	1.85×	81.804	-0.000	75.615	-0.000
	FlashAttention-4	1.039	18.28×	0.0217	46.07	1.83×	0.151	1.90×	81.804	-0.000	75.615	-0.000
	SageAttention2++	**0.996**	**19.08×**	**0.0213**	**46.99**	**1.86×**	**0.150**	**1.91×**	81.804	-0.000	75.615	-0.000
	SageAttention3	1.079	17.60×	0.0213	46.90	1.86×	0.151	1.90×	81.778	-0.026	75.584	-0.031
Flumina-B	Vanilla	46.877	1.00×	0.0724	13.81	1.00×	0.530	1.00×	82.052	+0.000	75.868	+0.000
	SDPA	2.581	18.16×	0.0274	36.46	2.64×	0.195	2.71×	82.051	-0.001	75.867	-0.001
	FlashAttention-4	1.962	23.90×	0.0269	37.11	2.69×	0.190	2.79×	82.032	-0.020	75.848	-0.020
	SageAttention2++	**1.660**	**28.25×**	**0.0268**	**37.27**	**2.70×**	**0.189**	**2.80×**	82.052	+0.000	75.868	+0.000
	SageAttention3	1.815	25.83×	0.0271	36.84	2.67×	0.190	2.78×	82.023	-0.029	75.837	-0.031
Flumina-M	Vanilla	64.183	1.00×	0.0966	10.35	1.00×	0.712	1.00×	81.877	+0.000	75.769	+0.000
	SDPA	5.861	10.95×	0.0382	26.16	2.53×	0.275	2.59×	81.877	+0.000	75.769	+0.000
	FlashAttention-4	5.136	12.50×	0.0379	26.42	2.55×	0.271	2.63×	81.877	+0.000	75.769	+0.000
	SageAttention2++	3.809	16.85×	0.0365	27.37	2.64×	0.260	2.74×	81.876	-0.001	75.768	-0.001
	SageAttention3	**3.301**	**19.44×**	**0.0359**	**27.83**	**2.69×**	**0.257**	**2.77×**	81.871	-0.006	75.763	-0.006
Flumina-L	Vanilla	76.201	1.00×	0.1140	8.77	1.00×	0.838	1.00×	82.112	+0.000	75.990	+0.000
	SDPA	7.592	10.04×	0.0450	22.22	2.53×	0.323	2.59×	82.122	+0.010	76.000	+0.010
	FlashAttention-4	6.448	11.82×	0.0441	22.69	2.59×	0.316	2.66×	82.113	+0.001	75.991	+0.001
	SageAttention2++	5.105	14.93×	0.0424	23.57	2.69×	0.304	2.75×	82.113	+0.001	75.991	+0.001
	SageAttention3	**4.411**	**17.28×**	**0.0421**	**23.77**	**2.71×**	**0.300**	**2.80×**	82.053	-0.059	75.930	-0.060

For fixed k, these kernels retain a block complexity of $\mathcal{O}(LC^2 + L^2C)$. FlashAttention-4 improves memory access and GPU utilization in BF16 (Zadouri et al., 2026). The SageAttention implementations introduce quantization within attention: SageAttention2++ combines INT8 and FP8 arithmetic, while SageAttention3 uses FP4 Tensor Cores (Zhang et al., 2025b;a).

As shown in Table 13, efficient attention kernels achieve attention speedups of $10.04\times$ to $28.25\times$ over Vanilla. These gains translate into inference speedups of $1.79\times$ to $2.71\times$ in the preliminary crop measurements and $1.85\times$ to $2.80\times$ for complete cases. The acceleration comes with little change in aggregate segmentation performance: the reported mean DSC and IoU remain within 0.06 percentage points of Vanilla.

H.3 Fused Decoder Upsampling

As detailed in Section 3.2 and Appendix B, each decoder stage first upsamples its input, then uses a $1 \times 1 \times 1$ convolution to match the channels for fusion with the next finer state. Under BF16 AMP, the default implementation interpolates FP32 features before converting the output to BF16 for convolution. Materializing this FP32 output adds a write-read cycle over the enlarged feature map. We therefore fuse interpolation and conversion in a single Triton kernel, retaining FP32 arithmetic while writing the result directly in BF16. Algorithm 1 summarizes the fused implementation.

Algorithm 1 Decoder upsampling with a fused BF16 output store.

```
# Y is preallocated in BF16 with dense NCDHW layout.
@triton.jit
def fused_upsample(X, Y, ...):
    n, c, p, valid = output_tile(32, 64)  # parallel output tiles
    ptrs, wx, wy, wz = neighbors_2x(X, n, c, p)
    v = tl.load(ptrs, valid, other=0).to(tl.float32)
    # v[z, y, x]: eight neighbors; lerp(a,b,w) = a + (b-a)*w
    a = lerp(v[:, :, 0], v[:, :, 1], wx)  # interpolate width
    b = lerp(a[:, 0], a[:, 1], wy)        # interpolate height
    r = lerp(b[0], b[1], wz)              # interpolate depth
    tl.store(Y + offset_NCDHW(n, c, p), r.to(tl.bfloat16), valid)
```

To evaluate whether the fused decoder further improves inference efficiency, we apply Algorithm 1 to the SDPA configuration in Table 13, using the same RTX 5090 and BF16 AMP. Table 14 reports mean case latency, peak allocated GPU memory, and DSC and IoU on BraTS-25-GLI.

Table 14: **Effect of fused decoder upsampling on BraTS-25-GLI.** Memory is reported in GiB. Speedups and memory reductions are relative to Original. In each metric group, Δ denotes Triton minus Original in percentage points.

Model	Time (s/case) ↓			Peak memory (GiB) ↓			DSC (%)			IoU (%)		
	Original	Triton	Speedup ↑	Original	Triton	Reduction ↑	Original	Triton	Δ (%)	Original	Triton	Δ (%)
Flumina-S	0.1557	**0.1271**	**1.225×**	2.832	**1.202**	**57.57%**	81.804	81.804	-0.000	75.615	75.615	-0.000
Flumina-B	0.1948	**0.1660**	**1.173×**	2.993	**1.351**	**54.87%**	82.051	82.052	+0.001	75.867	75.869	+0.002
Flumina-M	0.2737	**0.2445**	**1.119×**	3.324	**1.662**	**50.01%**	81.877	81.877	+0.000	75.769	75.769	-0.000
Flumina-L	0.3220	**0.2920**	**1.103×**	3.572	**1.906**	**46.64%**	82.122	82.122	+0.000	76.000	76.000	+0.000

As shown in Table 14, avoiding the enlarged FP32 intermediate substantially reduces peak allocated memory and consistently accelerates complete-case inference across model sizes. These efficiency gains are accompanied by little change in mean DSC and IoU.

H.4 Inference Speed Comparison

We next compare Flumina's inference speed across datasets with three representative segmentation methods: U-Net, MedNeXt, and SegMamba-V2. We extend the four efficient attention configurations with Triton decoder upsampling to all five datasets, using the inference protocol above and the

dataset-specific crop sizes in Table 6. All methods use BF16 AMP by default. Table 15 reports mean case latency and total inference time over each test set.

As shown in Table 15, Flumina-S and Flumina-B achieve lower case latency than U-Net across all five datasets with each efficient attention implementation. MedNeXt-L remains slower than U-Net despite fewer parameters and MACs. With SageAttention3, Flumina-M approaches U-Net's latency, while Flumina-L is comparable to MedNeXt-L and faster than SegMamba-V2 across all five datasets. SegMamba-V2 uses linear-time state-space modeling, whereas Flumina retains global self-attention. These comparisons show that theoretical complexity alone does not fully capture inference efficiency, while extensive global attention can remain practical with efficient implementations.

Table 15: **Inference speed comparison.** We report the average inference time per case and total time over each test set. Speedups are relative to U-Net. , , and indicate the first-, second-, and third-fastest attention implementations within each Flumina model size and dataset, respectively.

Model	Attention	AMOS (72 cases)			BraTS (578 cases)			KiTS (98 cases)			LiTS (26 cases)			MAMA (301 cases)		
		s/case ↓	Total (s) ↓	Speedup ↑	s/case ↓	Total (s) ↓	Speedup ↑	s/case ↓	Total (s) ↓	Speedup ↑	s/case ↓	Total (s) ↓	Speedup ↑	s/case ↓	Total (s) ↓	Speedup ↑
U-Net	–	6.540	470.91	1.00×	0.220	126.97	1.00×	9.755	956.01	1.00×	10.955	284.82	1.00×	0.673	202.57	1.00×
MedNeXt-L	–	8.270	595.44	0.79×	0.269	155.25	0.82×	12.274	1202.86	0.79×	13.667	355.34	0.80×	0.857	258.05	0.79×
SegMamba-V2	–	9.572	689.20	0.68×	0.312	180.29	0.70×	14.251	1396.59	0.68×	15.848	412.05	0.69×	0.991	298.16	0.68×
Flumina-S	Vanilla	8.479	610.52	0.77×	0.287	165.97	0.77×	12.905	1264.73	0.76×	14.323	372.41	0.76×	0.871	262.21	0.77×
	SDPA	3.841	276.58	1.70×	0.127	73.44	1.73×	5.353	524.58	1.82×	6.160	160.16	1.78×	0.377	113.58	1.78×
	FlashAttention-4	3.680	264.92	1.78×	0.121	70.04	1.81×	5.119	501.70	1.91×	5.725	148.86	1.91×	0.361	108.59	1.87×
	SageAttention2++	**3.679**	**264.92**	**1.78×**	**0.121**	**70.01**	**1.81×**	**5.087**	**498.57**	**1.92×**	**5.696**	**148.08**	**1.92×**	**0.361**	**108.57**	**1.87×**
	SageAttention3	3.701	266.50	1.77×	0.122	70.69	1.80×	5.129	502.63	1.90×	5.723	148.79	1.91×	0.364	109.45	1.85×
Flumina-B	Vanilla	15.506	1116.44	0.42×	0.530	306.34	0.41×	24.127	2364.49	0.40×	26.800	696.79	0.41×	1.617	486.77	0.42×
	SDPA	5.016	361.15	1.30×	0.166	95.94	1.32×	7.189	704.50	1.36×	7.957	206.88	1.38×	0.503	151.26	1.34×
	FlashAttention-4	4.907	353.28	1.33×	0.162	93.58	1.36×	7.029	688.84	1.39×	7.768	201.96	1.41×	0.491	147.72	1.37×
	SageAttention2++	**4.874**	**350.93**	**1.34×**	**0.161**	**92.80**	**1.37×**	**6.966**	**682.63**	**1.40×**	**7.696**	**200.09**	**1.42×**	**0.487**	**146.52**	**1.38×**
	SageAttention3	4.895	352.43	1.34×	0.161	93.23	1.36×	7.039	689.86	1.39×	7.719	200.69	1.42×	0.489	147.22	1.38×
Flumina-M	Vanilla	21.000	1511.99	0.31×	0.712	411.36	0.31×	32.540	3188.89	0.30×	36.192	940.99	0.30×	2.195	660.77	0.31×
	SDPA	7.486	538.96	0.87×	0.245	141.34	0.90×	10.902	1068.37	0.89×	12.059	313.54	0.91×	0.766	230.44	0.88×
	FlashAttention-4	7.377	531.14	0.89×	0.240	138.98	0.91×	10.753	1053.80	0.91×	11.862	308.40	0.92×	0.754	226.92	0.89×
	SageAttention2++	7.066	508.72	0.93×	0.230	132.74	0.96×	10.265	1005.95	0.95×	11.310	294.06	0.97×	0.720	216.86	0.93×
	SageAttention3	**6.956**	**500.83**	**0.94×**	**0.226**	**130.65**	**0.97×**	**10.177**	**997.33**	**0.96×**	**11.134**	**289.47**	**0.98×**	**0.707**	**212.86**	**0.95×**
Flumina-L	Vanilla	28.823	2075.29	0.23×	0.839	484.76	0.26×	38.393	3762.55	0.25×	42.767	1111.94	0.26×	2.576	775.37	0.26×
	SDPA	8.796	633.34	0.74×	0.292	168.79	0.75×	13.221	1295.67	0.74×	14.502	377.06	0.76×	0.907	272.87	0.74×
	FlashAttention-4	8.566	616.73	0.76×	0.286	165.20	0.77×	12.853	1259.64	0.76×	14.107	366.77	0.78×	0.884	266.08	0.76×
	SageAttention2++	8.256	594.42	0.79×	0.274	158.16	0.80×	12.324	1207.71	0.79×	13.534	351.89	0.81×	0.850	255.72	0.79×
	SageAttention3	**8.098**	**583.04**	**0.81×**	**0.269**	**155.34**	**0.82×**	**11.999**	**1175.93**	**0.81×**	**13.274**	**345.13**	**0.83×**	**0.834**	**251.02**	**0.81×**

I Additional Experiments and Analysis

We examine whether skips reduce trained models' reliance on the Transformer and whether the lightweight iterative tokenizer increases it. Unlike the from-scratch ablations in Section 4.3, we replace Transformer blocks with identity mappings at inference, without retraining or fine-tuning.

The four single-state variants (b)–(e) in Figure 5 use either a semantic encoder or an iterative tokenizer, each with and without encoder–decoder skips. Figure 14 compares their mean validation DSC across five datasets before and after complete Transformer removal. As shown in Figure 14, skips reduce the mean DSC drop from 17.70 to 6.76 percentage points with the semantic encoder, and from 46.81 to 28.44 with the iterative tokenizer. Under either skip setting, the tokenizer yields larger drops than the semantic encoder. These results are consistent with skips providing an alternative path for decoding and the lightweight tokenizer actually leaving more representation learning to the Transformer.

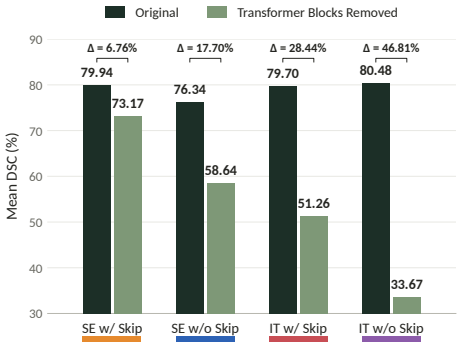

Figure 14: **Removing all Transformer blocks at inference.** Colored strips match the architectures in Figure 5. Δ denotes the DSC drop in percentage points.

We next examine individual datasets by replacing the last $r \in \{0, 3, 6, 9, 12\}$ blocks with identity mappings. Figure 15 reports the resulting DSC, while Figure 16 shows the drop relative to each unmodified model:

$$\Delta = \mathrm{DSC}(0) - \mathrm{DSC}(r), \tag{39}$$

where $\mathrm{DSC}(r)$ denotes the mean validation DSC on a given dataset after removing the last r blocks and $\mathrm{DSC}(0)$ denotes the original score. Both scores are expressed as percentages.

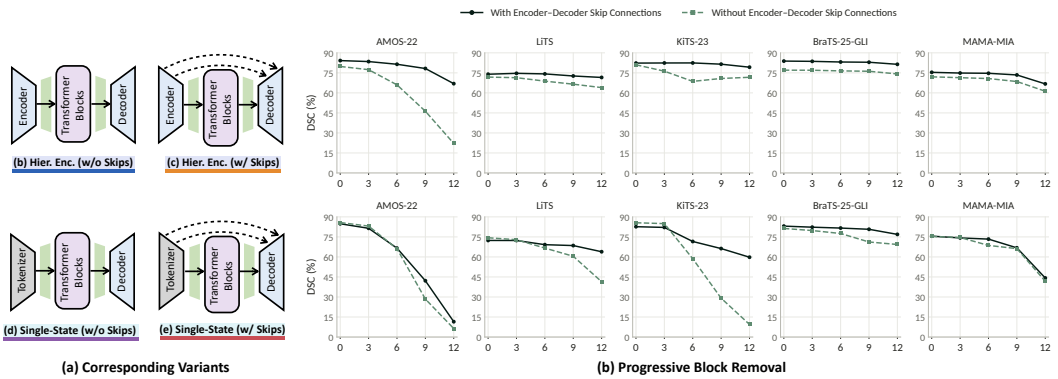

Figure 15: **Progressive Transformer block removal at inference.** (a) Variants from Figure 5. (b) DSC after progressively replacing blocks with identity mappings, without retraining. *Top:* semantic encoder. *Bottom:* iterative tokenizer.

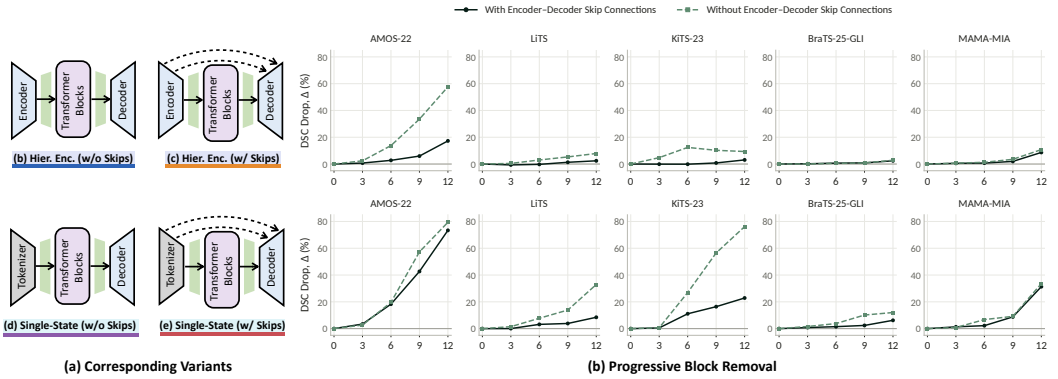

Figure 16: **DSC drops under progressive Transformer block removal.** (a) Variants from Figure 5. (b) DSC drop $\Delta = \mathrm{DSC}(0) - \mathrm{DSC}(r)$ after removing the last r blocks without retraining, measured in percentage points. *Top:* semantic encoder. *Bottom:* iterative tokenizer.

We first focus on AMOS-22. Figures 15 and 16 show that, under complete removal ($r = 12$), AMOS-22 has the largest DSC drop among the five datasets for every variant. With skips retained, the semantic-encoder and iterative-tokenizer models lose 17.28 and 73.36 percentage points, respectively. This sensitivity reflects greater demand for organ-discriminative representations in this multi-organ task. The encoder contrast suggests that semantic-encoder skips already supply such features, whereas the iterative tokenizer leaves more of this representation learning to the Transformer.

On LiTS and KiTS-23, Figure 16 shows that skips reduce the DSC drop after complete Transformer removal with both encoders, especially the iterative tokenizer. For the tokenizer, skips reduce the drop from 32.94 to 8.45 percentage points on LiTS and from 76.15 to 22.86 on KiTS-23. The smaller drops suggest that multi-scale tokenizer features remain useful for prediction when passed directly to the decoder through skips, without Transformer processing.

These results support the design motivation in Section 4.3: preserve multi-scale structural information without a separate encoder–decoder bypass that can reduce reliance on the Transformer. As described in Section 3.2, Flumina maintains and updates this information as persistent states within the shared ViT and decodes only the final states.

J LIMITATIONS AND FUTURE WORK

Flumina addresses the representation–structure dilemma in 3D medical image segmentation by maintaining multi-scale features as persistent states within a shared ViT. This keeps the Transformer responsible for representation learning while preserving structural detail. How effectively this design scales, however, remains an open question.

Under our setting, scaling from S to B improves DSC across all five datasets, whereas further scaling to M and L yields smaller and less consistent overlap gains. We hypothesize that recurring anatomical patterns may partly explain the limited gains from larger models: once common task-relevant patterns are learned, additional capacity may offer limited benefits. More scans with similar patterns may likewise provide little new supervision.

From our perspective, Flumina's design connects general-purpose Transformer architectures with the requirements of medical segmentation. We hope that adapting general-purpose methods such as latent visual reasoning to Flumina will offer a path to exploring scaling laws in medical image analysis, even with limited labeled training data.