

Coronary Artery Segmentation with SegMamba: A Practical Task-Transfer Evaluation Under Single-GPU Constraints

Lars Talian Stangebye-Hansen and Oliver Sanchez
Department of Computer Science
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway

Abstract—This paper reports a focused task-transfer evaluation of coronary artery segmentation using modern Mamba-based models. The work was motivated by the recent release of SegMamba and the question of whether it improves segmentation quality in this specific task under realistic constraints. We compare three model families on the same ASOCA-style workflow: a U-Net baseline, a Mamba-encoder variant, and the SegMamba architecture as released by the original authors, trained through our task-specific pipeline. Mean validation Dice was 0.7844 (U-Net), 0.7734 (Mamba-encoder), and 0.7673 (SegMamba). The key finding is practical: in our setup, optimization stability and patch strategy were more decisive than architectural novelty. Experiments were compute-limited to a single A100 GPU, which constrained sweep depth and prevented exhaustive task-specific hyperparameter tuning.

Index Terms—coronary segmentation, SegMamba, Mamba, 3D medical imaging, reproducibility

I. INTRODUCTION

Coronary artery segmentation is a challenging 3D medical imaging problem because vessels are thin, branching, and highly sensitive to local contrast and global continuity errors. Our project goal was not to produce a benchmark claim, but to evaluate a timely technical question: after SegMamba’s release, does it provide practical gains for coronary segmentation in a constrained engineering setting?

The intent was explicitly exploratory and model-centric: test SegMamba on this specific problem, compare against strong alternatives, and document where performance was won or lost.

II. RELATED WORK

ImageCAS established a large-scale benchmark context for coronary artery segmentation from CTA volumes [1]. SegMamba proposed long-range sequential modeling for 3D medical segmentation [2]. LightM-UNet and related Mamba-assisted segmentation work provide relevant architectural baselines for encoder-side sequence modeling [3].

III. EXPERIMENTAL SETUP

A. Data and Split

We used ASOCA-style coronary CTA data with voxel-level labels across normal and diseased cohorts. All comparisons used the same fixed 80/20 train-validation split. For context,

SegMamba reports 70/10/20 train-validation-test splits on its benchmark datasets [2].

B. Metrics

Primary metric was mean validation Dice. HD95 was tracked as a secondary boundary-quality metric.

C. Compute Budget

The project was limited to a single A100 GPU. This directly constrained full-volume training feasibility, total hyperparameter sweep depth, and the number of long training runs that could be executed. In contrast, SegMamba reports training on four NVIDIA A100 GPUs for 1000 epochs with batch size 2 per GPU [2].

IV. MODEL VARIANTS

U-Net baseline: convolutional encoder-decoder with skip connections.

Mamba-encoder: U-Net-style network with Mamba blocks in the encoder [3].

SegMamba: released SegMamba architecture, trained using our custom task-specific pipeline [2].

V. TRAINING DECISIONS THAT MATTERED

Three implementation decisions were first-order:

- Foreground sampling to reduce background-only updates.
- Patch-wise optimization due to memory limits on 3D full-volume training.
- Stability controls (mixed precision, gradient clipping, deterministic split/seed, periodic validation, checkpointing).

These steps were necessary to make SegMamba training behavior stable enough for meaningful comparison. Our training recipe also differs from SegMamba’s reported defaults (cross-entropy loss, SGD optimizer, polynomial LR decay) [2].

VI. RESULTS

U-Net was strongest in this regime. Mamba-based models were competitive but did not surpass the baseline under the available budget.

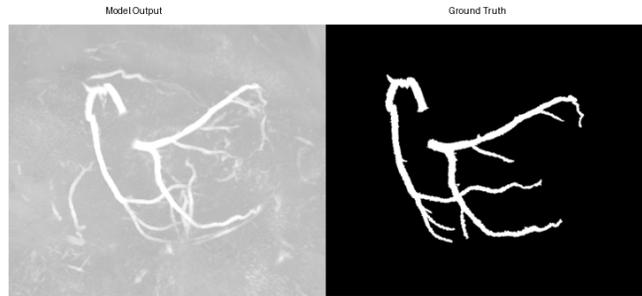
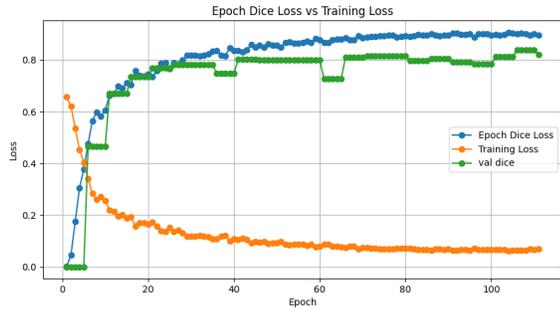


Fig. 1. Legacy evidence from the SegMamba pipeline used in this study. Left: training dynamics (loss and validation Dice) under single-GPU constraints. Right: representative SegMamba prediction versus ground-truth mask.

TABLE I
MEAN VALIDATION DICE ON THE SAME SPLIT AND WORKFLOW.

Model	Dice	Rank
U-Net baseline	0.7844	1
Mamba-encoder	0.7734	2
SegMamba	0.7673	3

VII. DISCUSSION

The negative result versus U-Net should not be interpreted as evidence against SegMamba in general. In this setup, the most likely bottlenecks were optimization and scale rather than representational ceiling: higher sensitivity to hyperparameters, patch-policy effects on continuity learning, and limited sweep breadth under a single-GPU budget. Given differences in data protocol, optimizer/loss recipe, and compute scale relative to SegMamba’s reported setting [2], this result should be treated as a constrained task-transfer evaluation rather than a direct leaderboard comparison. The legacy evidence in Fig. 1 is consistent with this interpretation: optimization remains stable, qualitative outputs are reasonable, and validation improvements flatten under a constrained search budget. In short, our intent was to explore SegMamba early on this domain; we obtained a stable and comparable result, but not a new best score.

VIII. LIMITATIONS AND NEXT STEPS

This is a validation-centric study and not a benchmark submission. We also did not run exhaustive task-specific hyperparameter searches for SegMamba due to the single-GPU compute budget. The primary run profile was: $1 \times A100$, patch size $224 \times 224 \times 96$, batch size 1, 120 epochs, validation every 5 epochs, and mixed precision training. The most valuable next steps are overlap-aware patching, structured sweeps on patch/loss parameters, threshold calibration per volume, and experiments on larger datasets where long-range modeling may provide larger gains.

IX. REPRODUCIBILITY

The repository is cleaned for external review and includes CLI-based training/inference entrypoints and `uv`-based envi-

ronment setup.

Observed failure modes under this budget:

- unstable convergence with naive random cropping,
- sensitivity to patch-policy and threshold choice for thin-vessel continuity,
- limited sweep depth relative to model complexity.

X. CONCLUSION

Under a single-A100 budget, SegMamba was stable and competitive but did not exceed a tuned U-Net baseline; the highest-leverage next steps are structured hyperparameter sweeps and overlap-aware patching.

REFERENCES

- [1] A. Zeng *et al.*, “ImageCAS: A large-scale dataset and benchmark for coronary artery segmentation based on computed tomography angiography images,” *arXiv preprint arXiv:2211.01607*, 2022.
- [2] Z. Xing, T. Ye, Y. Yang, G. Liu, and L. Zhu, “SegMamba: Long-range sequential modeling Mamba for 3D medical image segmentation,” *arXiv preprint arXiv:2401.13560*, 2024.
- [3] W. Liao, Y. Zhu, X. Wang, C. Pan, Y. Wang, and L. Ma, “LightM-UNet: Mamba assists in lightweight UNet for medical image segmentation,” *arXiv preprint arXiv:2403.05246*, 2024.