3D Cardiac Structure Segmentation with Two Stage Cascaded Neural Networks

Bryan Anenberg[1] and Peter D. Chang, MD[2]

[1] University of California Irvine, California USA banenber@uci.edu
[2] University of California Irvine, California USA changp6@hs.uci.edu

Abstract. We present a two-stage cascaded 3D convolutional neural network for semantic segmentation of atrial structures in late gadolinium enhancement (LGE) cardiac MRI, a modality widely used in the management of atrial fibrillation [5]. This task is particularly challenging due to low tissue contrast, imaging artifacts, and the anatomical ambiguity of thin-walled atrial tissue [7]. Our model segments three key cardiac regions: the *Right Atrium Cavity*, *Left Atrium Cavity*, and *Left & Right Atrial Walls*.

The proposed method achieved 3rd place in the Multi-class Bi-Atrial Segmentation (MBAS) Challenge at MICCAI 2024 (STACOM workshop) and was included in the official benchmarking study. On 5-fold cross-validation of the training dataset, our final model achieved Dice scores of 0.711 (Wall), 0.920 (Right Atrium), and 0.930 (Left Atrium), with corresponding 95th percentile Hausdorff distances (HD95) of 2.993 mm, 3.593 mm, and 3.978 mm, respectively—highlighting its robustness in delineating both thin and volumetric cardiac structures.

To further improve performance and efficiency, we conducted a comprehensive ablation study spanning more than a dozen model, training, and inference configurations. We systematically evaluated variations in architecture (nnU-Net Residual Encoder [2,3] vs. MedNeXt [6]), input resolution, receptive field size, patch sampling, loss functions, and data augmentation strategies. These experiments guided the final model configuration, refined post-processing steps, and enabled a runtime-optimized inference strategy that achieves over 50% speed-up—meeting strict competition time constraints.

1 Introduction

Cardiac magnetic resonance (CMR) is the preeminent imaging modality for evaluating cardiac function and morphology, including the measurement of the cardiac wall thickness, and atrium volume [1]. Late gadolinium enhancement (LGE) is an important CMR imaging technique for characterizing cardiac tissue. LGE refers to the hyper-enhancement of the myocardium that is observed in cardiac MRI sequences performed after the injection of a Gadolinium-based (Gd) contrast agent [1]. The enhancement is caused by the Gd lingering in the myocardial interstitium, which often indicates myocardial fibrosis or a cardiomyopathy such as myocarditis that increases the volume of the interstitial space [1].

Atrial fibrillation cardiac arrhythmias can be treated by minimally invasive catheter ablation in which radio-frequency energy or cryotherapy is used to burn the cardiac tissue to disrupt the abnormal electrical signals responsible for the arrhythmia. The injured myocardium is replaced by fibrous scar tissue, which sets the desired electrical conductivity pattern. LGE-MRI can be used to access the quality of the atrial ablation lesions, predict the success of the ablation procedure, and guide patient-specific treatment options [5].

Cardiac structures such as the atrium must be accurately segmented in order to prepare the LGE-MRI for clinical review. While the LGE-MRI may increase the visibility of fibrotic cardiac tissue, the contrast and clarity of non-diseased tissue is reduced, which increases the difficulty of automatic segmentation [7]. The need for high-quality automated segmentation methods is the motivation for the Multi-class Bi-Atrial Segmentation (MBAS) Challenge at MICCAI 2024 where participants were invited to design automatic semantic segmentation methods capable of predicting the 3D volumes of the *Right Atrium Cavity*, *Left Atrium Cavity*, and *Left & Right Atrium Walls* in LGE-MRI images. The algorithm presented in this paper was a submission to the MBAS Challenge.

1.1 Related Work

The previous iteration of the MBAS Challenge was the *Left Atrium Segmentation Challenge* organized in 2018. Algorithms submitted to that challenge were trained on a dataset of 154 3D LGE-MRI images (with 54 images reserved for the test set) to perform the task of predicting the left atrium cavity volume. The top performing methods cited in the benchmarking study achieved a dice score of 93.2% and a mean surface to surface distance of 0.7mm [7]. Many of these methods utilized two convolutional neural networks (CNNs), the first of which was used for region-of-interest localization and the second was used for refined regional segmentation [7].

nnU-Net and nnU-Net Revisisted The *nnU-Net* framework introduced a fully automated and adaptive pipeline for configuring U-Net-based models across diverse biomedical segmentation tasks [2]. Its success stems not from architectural novelty but from systematic, data-driven configuration of preprocessing, training, and inference components. Core hyperparameters (e.g., learning rate, loss function, data augmentation) are globally fixed, while others—such as image normalization, resampling, patch size, and network topology—are heuristically determined based on dataset-specific properties like voxel spacing, intensity distribution, and available GPU memory.

In line with this approach, we adopt nnU-Net as a strong baseline for the MBAS dataset. Rather than using the default U-Net, we follow recommendations from *nnU-Net Revisited* [3] and train a variant with residual connections in the encoder (*ResEnc*), which outperformed the standard U-Net across six public benchmarks (BTCV, ACDC, LiTS, BraTS, KiTS, and AMOS). These findings reinforce that well-optimized CNN architectures continue to outperform newer transformer-based and Mamba-style models in many medical imaging tasks. We also include MedNeXt [6], another top-performing CNN variant, in our architectural comparison.

ConvNeXt and MedNeXt *ConvNeXt* [4] demonstrates that a carefully modernized CNN can outperform hierarchical Vision Transformers such as Swin Transformer on major computer vision benchmarks, including ImageNet classification, COCO object detection, and ADE20K semantic segmentation. Built on ResNet, ConvNeXt incorporates several Transformer-inspired design elements—such as a patchified stem, depthwise separable convolutions, inverted bottlenecks, larger kernel sizes (7×7), GELU activations, Layer Normalization, and streamlined normalization and activation patterns—resulting in significantly improved performance while retaining the efficiency of convolutional models.

MedNeXt [6] adapts ConvNeXt blocks within a fully convolutional U-Net architecture, scaling them effectively for volumetric medical image segmentation. It achieves state-of-the-art results on multiple CT and MRI benchmarks, including BTCV, AMOS22, KiTS19, and BraTS21, demonstrating the effectiveness of ConvNeXt-style CNNs in medical domains.

1.2 Contribution

In this work, we explore a range of architectural, training, and inference configurations within the nnU-Net framework for 3D semantic segmentation in the MBAS Challenge. We evaluate variants based on the *nnU-Net ResEnc* and *MedNeXt* architectures, ultimately converging on a two-stage cascaded pipeline composed of two *nnU-Net ResEnc* models. Our final model, trained on the 70-case MBAS training set with 5-fold cross-validation, achieved Dice scores of 0.711 (Wall), 0.920 (Right Atrium), and 0.930 (Left Atrium), and was submitted to the MBAS Challenge at MICCAI 2024, where it placed 3rd overall.

2 Experimental Design

2.1 Configurations, Implementation, and Baselines

Most experiments were conducted using default settings from the nnU-Net framework. Models were trained for 1000 epochs using stochastic gradient descent (SGD) with momentum 0.99, weight decay of 3×10^{-5}, and an initial learning rate of 1×10^{-2}. The learning rate was decayed according to an exponential schedule:

$$lr_t = lr_0 \cdot (1 - t/T)^{0.9}, \tag{1}$$

where t is the current iteration and T is the total number of iterations.

While 5-fold cross-validation is preferred to ensure each sample serves as validation at least once, most experiments were limited to a single fold due to computational constraints. Despite increased variance from this setup, observed trends remained consistent across configurations. Final models submitted to the challenge were trained on the full 70-case training set without cross-validation.

2.2 Dataset

The MBAS dataset comprises multi-center bi-atrial LGE-MRI scans acquired using a standardized imaging protocol. All volumes have dimensions of either $44 \times 640 \times 640$ or $44 \times 576 \times 576$, with voxel spacing of 2.5 mm (slice thickness) and 0.625 mm (in-plane resolution). The dataset includes 70 training images, 30 validation images, and 100 held-out test cases.

3 nnU-Net ResEnc Baseline Models

The nnU-Net ResEnc models were trained as a baseline. Although the ResEnc (L) model achieved the best performance of the group, the (M) model demonstrated very competitive results with a significantly shorter training time. Most of the subsequent experiments are performed with the ResEnc (M) sized models to expedite the experimental process.

Model	Patch	Wall		Right Atrium		Left Atrium		GPU
	Size	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Hours
ResEnc (M)	$20 \times 256 \times 256$	0.724	2.878	0.923	3.400	0.931	3.761	11.39
ResEnc (L)	$32 \times 384 \times 384$	0.725	2.763	0.926	3.200	0.930	3.688	37.64
ResEnc (XL)	$40 \times 448 \times 448$	0.718	2.913	0.924	3.248	0.930	3.658	90.61

Table 1: Evaluation results are on $fold_0$ of the training dataset. Training time was benchmarked on a NVIDIA RTX 4090.

Further experiments demonstrated that training for 1000 epoch is better than training for fewer (250) epochs, and that using 3D convolutions is better than 2D convolutions applied to the z-axis slices.

Model	Patch Size	Epochs	Wall		Right Atrium		Left Atrium	
			Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc (M)	$20 \times 256 \times 256$	1000	0.714	3.036	0.921	3.557	0.930	4.035
ResEnc (M)	$20 \times 256 \times 256$	250	0.715	3.406	0.919	3.646	0.928	4.430
ResEnc (M) 2D	512×512	250	0.688	3.584	0.910	3.971	0.921	4.719

Table 2: Evaluation results on 5-fold cross-validation of the MBAS training dataset.

4 Two-Stage Cascaded 3D Segmentation Architecture

The motivation behind our two-stage cascaded architecture is to decouple the tasks of foreground localization and fine-grained multi-class segmentation. The first stage focuses on high-recall binary segmentation of the atrial region, while the second stage performs precise multi-class segmentation of the three MBAS structures: *Right Atrium Cavity*, *Left Atrium Cavity*, and *Left & Right Atrial Walls*.

In stage one, a 3D CNN is trained to predict a binary mask representing the union of all atrial labels. In stage two, a separate network classifies each voxel within this predicted foreground into one of the three target classes or background. The binary mask is used to gate the deep supervision losses, restricting computation to the atrial region. This targeted loss formulation enables the second-stage model to focus solely on resolving fine-grained intra-atrial boundaries.

We apply deep supervision by aggregating losses from intermediate decoder stages, combining region-based *Soft Dice Loss* and *Cross Entropy Loss.* The Soft Dice Loss is defined as:

$$\mathcal{L}_{\text{Dice}} = \frac{1}{C} \sum_{k=1}^{C} \frac{2 \cdot TP_k + \epsilon}{2 \cdot TP_k + FP_k + FN_k + \epsilon}, \tag{2}$$

where C is the number of classes, and TP_k, FP_k, and FN_k represent true positives, false positives, and false negatives for class k, with $\epsilon = 1 \times 10^{-5}$ for numerical stability. The binary mask is applied element-wise to each tensor before reduction.

Cross Entropy Loss is computed using PyTorch's `torch.nn.CrossEntropyLoss`, with the same binary mask applied prior to averaging.

4.1 Comparison with nnU-Net Cascaded Architecture

Our two-stage cascaded design differs from the nnU-Net style cascade, in which both stages perform full multi-class segmentation. In nnU-Net, the stage-one output is converted to a one-hot tensor (e.g., shape $(B, 3, X, Y, Z)$ for three classes) and concatenated with the original input image $(B, 1, X, Y, Z)$, yielding a combined input of shape $(B, 4, X, Y, Z)$ for the second stage.

In contrast, our approach uses a binary mask for foreground gating and restricts the second stage's supervision to the atrial region only. As shown in Appendix C, nnU-Net style cascades consistently underperformed compared to both our two-stage design and even the single-stage baseline.

4.2 nnU-Net ResEnc vs. MedNeXt Architectures

We evaluated two backbone architectures: *nnU-Net ResEnc* and a *MedNeXt*-inspired variant. While performance was generally comparable, the ResEnc model consistently outperformed MedNeXt across most metrics. Accordingly, our final submission to the MBAS Challenge used a two-stage cascade composed of two ResEnc networks

5 First Stage Binary Segmentation Model

The goal of the first-stage model is to localize the atrial region using a high-recall binary segmentation. We evaluated multiple configurations to maximize overlap with the ground truth while maintaining a competitive Dice score. The overlap score and Dice coefficient are defined as:

$$\mathrm{Overlap}(A,B) = \frac{|A \cap B|}{|B|}, \quad \mathrm{Dice}(A,B) = \frac{2|A \cap B|}{|A| + |B|} \tag{3}$$

To mitigate overfitting on the limited 70-case dataset, all models were evaluated under 5-fold cross-validation (56 training, 14 validation cases per split).

Through extensive experimentation (see Appendix A), we identified an optimal configuration based on a 6-stage *nnU-Net Residual Encoder* with reduced maximum feature channels (96 vs. 320), 50% dropout, and 25% foreground sampling. Input volumes were downsampled from $(2.5, 0.625, 0.625)$ mm to $(2.5, 0.9737, 0.9737)$ mm voxel spacing. Full architectural and training parameters are detailed in Table 27.

5.1 Post-processing: Non-Largest Component Suppression

Since the union of the three MBAS labels always forms a single contiguous atrial region, we apply a post-processing step to retain only the largest connected component in the predicted binary mask. This removes spurious false positives produced by the first-stage model.

Empirically, this step reduces HD95 by eliminating outlier predictions. As shown in Figure 1, it can suppress large disconnected regions—reducing the HD95 from 140.026 to 2.828.

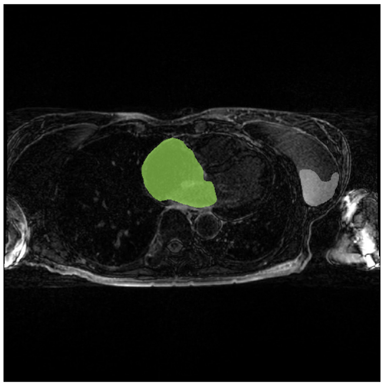
Further support for this step comes from second-stage model results with and without post-processing (Table 3), where omitting it consistently degraded performance.

Model	1st Stage Mask	Wall		Right Atrium		Left Atrium	
	Post-proc.	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc	Yes	0.723	2.727	0.925	3.215	0.931	3.714
ResEnc	No	0.720	2.840	0.924	3.359	0.931	3.819

Table 3: Second-stage performance with and without post-processing on fold 0.

5.2 Post-processing: Binary Dilation

To expand the predicted foreground region, we apply 3D morphological binary dilation to the first-stage binary mask. This operation uses a spherical structuring element to add a margin around the predicted boundary: each voxel at (i, j, k) is set to 1 if any voxel within the structuring element radius is active.



(a) Post-processing removes a false positive (white), preserving only the correct atrial prediction (green).

Model	Post-proc.	Dice[↑]	HD95[↓]
ResEnc	No	0.916	6.714
ResEnc	Yes	0.917	4.716

(b) First-stage performance with and without post-processing.

Fig. 1: Effect of non-largest component suppression. (a) Visual correction of a false positive. (b) Quantitative improvement in segmentation metrics (5-fold cross-validation).

Larger dilation radii increase overlap with the ground truth but degrade Dice and HD95 metrics by introducing over-segmentation (Table 4). Full results across dilation radii are provided in Appendix B.3. The final MBAS submission used a dilation radius of 2.

Model	Radius	Overlap[↑]	Dice[↑]	HD95[↓]
ResEnc	–	0.961	0.917	4.716
ResEnc	1	0.980	0.874	5.416
ResEnc	2	0.988	0.825	6.570
ResEnc	3	0.992	0.772	7.987

Table 4: Impact of dilation radius on first-stage performance (5-fold cross-validation).

6 Second Stage Semantic Segmentation Model

6.1 Motivation

To evaluate the potential benefit of the two-stage cascaded design, we conducted an experiment where the second-stage model was trained using a binary mask derived from the ground truth—effectively simulating a perfect first-stage prediction. As shown in Table 5, this setup substantially improved segmentation metrics across all structures.

To simulate less accurate first-stage outputs, we progressively dilated the binary mask. Although segmentation accuracy decreased with increasing dilation (indicating reduced precision), all configurations still outperformed the baseline model trained without any mask—particularly in terms of HD95. This demonstrates that even imperfect localization provides useful spatial priors for improving boundary precision.

Model	Binary	Radius	Wall		Right Atrium		Left Atrium	
	Mask		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc	No	–	0.724	2.878	0.923	3.400	0.930	3.761
ResEnc	Yes	0	0.801	2.601	0.945	2.374	0.950	2.525
ResEnc	Yes	1	0.738	2.537	0.934	2.566	0.940	2.602
ResEnc	Yes	2	0.729	2.549	0.930	2.677	0.937	2.793

Table 5: Second-stage performance using perfect and dilated binary masks as spatial priors (evaluated on $fold_0$).

6.2 Empirical Results

Based on a series of experiments (Appendix B), we selected a 7-stage *nnU-Net Residual Encoder* architecture for the second stage, using 100% foreground patch sampling. The model is structurally identical to the baseline *ResEnc (M)*—with 320 features per stage—and operates directly on native-resolution LGE-MRI volumes (voxel spacing: $2.5 \times 0.625 \times 0.625$ mm). During training and inference, predictions outside the first-stage binary mask are ignored, and non-largest component suppression is applied independently to each output class.

Our final submission to the MBAS Challenge, denoted *ResEnc final* in Table 6, achieved results comparable to the baseline *ResEnc (M)*. The key distinction is in how each model handles false positives: *ResEnc (M)* depends heavily on post-processing, whereas *ResEnc final* inherently suppresses outliers through its binary mask prior. Notably, *ResEnc final* maintains strong HD95 performance even without post-processing, unlike *ResEnc (M)*, whose HD95 degrades significantly when post-processing is removed.

7 Inference Runtime Speed-up

The final model was submitted to the MBAS Challenge as a Docker container containing a `predict.py` script for processing the 100 held-out test cases.

By default, nnU-Net performs inference by sliding a patch-sized window over the 3D volume with a stride of 0.5, causing overlap between patches. While this

Model	1st Stage	Postproc.	Wall		Right Atrium		Left Atrium	
	Mask		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc final	Yes	Yes	0.711	3.026	0.920	3.609	0.930	3.996
ResEnc final	Yes	No	0.711	2.993	0.920	3.593	0.930	3.978
ResEnc (M)	No	Yes	0.714	3.036	0.921	3.556	0.929	4.035
ResEnc (M)	No	No	0.713	3.016	0.921	3.567	0.929	6.302

Table 6: Second-stage performance under different masking and post-processing conditions (5-fold cross-validation).

overlap reduces edge artifacts from padding, it increases the number of forward passes and overall runtime.

To accelerate inference, we increased the stride to 1.0, eliminating patch overlap. For an input volume of size 44 × 640 × 640 and a patch size of 20 × 256 × 256, this reduced the number of steps from 64 to 27. As a result, the average runtime per volume decreased from 20.80 seconds to 11.81 seconds on an NVIDIA RTX 4090 GPU. As shown in Table 7, segmentation performance was minimally affected by this change.

Model	Step	Wall		Right Atrium		Left Atrium	
	Size	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc final	0.5	0.711	3.026	0.920	3.609	0.930	3.996
ResEnc final	1.0	0.709	3.217	0.919	3.594	0.929	4.087

Table 7: Effect of patch stride on inference accuracy (5-fold cross-validation).

8 Conclusion

We presented a two-stage cascaded 3D segmentation framework for the semantic segmentation of three cardiac structures in LGE-MRI: the *Right Atrium Cavity*, *Left Atrium Cavity*, and *Left & Right Atrial Walls*. The architecture separates foreground localization and fine-grained multi-class segmentation using two *nnU-Net Residual Encoder* models in series.

The final model achieved strong 5-fold cross-validation results, with Dice scores of 0.711 (Wall), 0.920 (Right Atrium), and 0.930 (Left Atrium), and HD95 scores below 4.1 mm across all structures. Submitted to the MBAS Challenge at MICCAI 2024, our method placed 3rd overall and was included in the official benchmarking study.

While the cascaded design improves robustness by isolating the segmentation task to the foreground region, its overall performance is bounded by the quality of the first-stage binary mask. Experiments with oracle and perturbed masks (Table 5) suggest that improved localization can yield further gains. Accurately segmenting the thin and ambiguous atrial wall regions remains a key challenge and opportunity for future work.

Appendix

This appendix provides additional experimental details, including the development of the first-stage binary segmentation model (Table 27) and the second-stage multi-class segmentation model (Table 28). It also includes a comparison between the proposed two-stage cascade and the default *nnU-Net* cascaded architecture (Appendix C).

A First Stage Binary Segmentation Model Experiments

A.1 Training on Low-Resolution Downsampled Images

In cascaded architectures, the first-stage model is often trained on downsampled images to increase the receptive field and improve prediction consistency, albeit at the cost of fine detail. We compared a *nnU-Net ResEnc (M)* model trained at the original voxel spacing (2.5 × 0.625 × 0.625 mm) with one trained using nnU-Net's recommended *lowres* configuration (2.5 × 0.9737 × 0.9737 mm), which downscales input volumes from 44 × 638 × 638 to 44 × 410 × 410.

As shown in Table 8, the low-resolution model achieved slightly better overlap and HD95 scores while using fewer parameters.

Model	Patch Size	Voxel Spacing (mm)	Overlap[↑]	Dice[↑]	HD95[↓]	#Params	FLOPs
ResEnc lowres	28 × 256 × 224	2.5 × 0.97 × 0.97	0.935	0.934	3.399	102M	839G
ResEnc	20 × 256 × 256	2.5 × 0.625 × 0.625	0.931	0.933	3.514	140M	805G

Table 8: Evaluation results on $fold_0$ comparing high- vs. low-resolution training for the first-stage ResEnc model.

A.2 Foreground Sampling Likelihood

Since the atrial region occupies only a small portion of each image volume (about 270k of 18 million voxels), randomly sampled patches often contain no foreground. To address this, we varied the probability of sampling patches centered on the atrium—referred to as the foreground sampling likelihood.

Across multiple architectures, including *MedNeXt* and a lightweight *MedNeXt slim96* variant, a 25% foreground sampling rate consistently yielded the best balance of overlap and HD95. Models trained with 0% foreground sampling (i.e., fully random patches) underperformed due to underexposure to the target region. On the other hand, 100% sampling degraded performance, likely due to reduced exposure to background context and increased overfitting.

As a result, a 25% foreground sampling likelihood was adopted as the default in subsequent experiments.

A.3 Residual Encoder U-Net vs. MedNeXt U-Net

We compared *nnU-Net ResEnc* and *MedNeXt* models configured with identical architecture depth, channel widths, filter sizes, and strides. As shown in Table 9, *ResEnc* slightly outperformed *MedNeXt* in all metrics, but at the cost of a 4× increase in parameters and FLOPs. These results highlight the performance–efficiency trade-off between the two architectures.

Model	Overlap[↑]	Dice[↑]	HD95[↓]	#Params	FLOPs
ResEnc	0.935	0.934	3.427	102M	839G
MedNeXt	0.931	0.933	3.677	24.5M	254G

Table 9: Performance comparison on $fold_0$ of the training set. Both models trained for 1000 epochs with a patch size of $28 \times 256 \times 224$ and 25% foreground sampling.

A.4 Increasing the Receptive Field Size

To evaluate the effect of a larger receptive field, we modified the *ResEnc* model by increasing the convolutional kernel size from $3 \times 3 \times 3$ to $5 \times 5 \times 5$ in the early encoder stages:

```
stage1: [1,3,3] → [1,5,5]
stage2: [3,3,3] → [3,5,5]
stage3: [3,3,3] → [5,5,5]
```

As shown in Table 10, the larger kernels increased model complexity (118M parameters, 2302G FLOPs) but yielded slightly worse segmentation performance than the default $3 \times 3 \times 3$ configuration. Additional experiments with dilated convolutions also failed to improve results and are omitted here for brevity.

Model	Overlap[↑]	Dice[↑]	HD95[↓]	#Params	FLOPs
ResEnc (3×3×3)	0.935	0.934	3.427	102M	839G
ResEnc (5×5×5)	0.932	0.933	3.472	118M	2302G

Table 10: Comparison of *ResEnc* models with different kernel sizes, trained on $fold_0$ with patch size $28 \times 256 \times 224$ and 25% foreground sampling.

A.5 Effect of Dropout

We evaluated the impact of dropout on the *ResEnc* model by varying dropout probabilities from 0% to 50%. As shown in the table below, increasing dropout improved the overlap score but led to slight declines in Dice and HD95 performance, suggesting a trade-off between coverage and boundary precision.

Model	Dropout %	Overlap[↑]	Dice[↑]	HD95 [↓]
ResEnc	0	0.935	0.934	3.427
ResEnc	25	0.946	0.931	3.467
ResEnc	50	0.957	0.924	3.719

Table 11: Performance of *ResEnc* with varying dropout rates on $fold_0$ (patch size: $28 \times 256 \times 224$, 25% FG sampling).

A.6 Reducing Model Size

To reduce model complexity, we limited the maximum number of feature channels in the later network stages. For instance, setting the feature dimension to 128 changes the channel progression from 32, 64, 128, 256, 320, 320 to 32, 64, 128, 128, 128, 128. As shown in Table 12, the *ResEnc (feature dim = 96)* model achieved the highest overlap while significantly reducing parameter count and FLOPs, with only a minor drop in Dice and HD95.

Model	Feature Dim.	Overlap[↑]	Dice[↑]	HD95[↓]	#Params	FLOPs
ResEnc	96	0.961	0.917	4.073	11.3M	234G
ResEnc	128	0.958	0.924	3.818	23.1M	665G
ResEnc	320	0.957	0.924	3.719	102M	839G

Table 12: Performance of *ResEnc* models with varying feature dimensions on $fold_0$. All models trained with 50% dropout, patch size $28 \times 256 \times 224$, and 25% foreground sampling.

A.7 MedNeXt Architectural Variants

We explored minor architectural modifications to the *MedNeXt* model to increase its similarity to the *ResEnc* design.

Decoder Update: replaces MedNeXt's upsampling block with a single `ConvTranspose3D` layer.

Stem Update: replaces the default stem (`Conv3D`, `GroupNorm`) with a deeper `StackedConvBlock` composed of `Conv3D`, `Dropout`, `InstanceNorm`, and `LeakyReLU`, as used in *ResEnc*.

As shown in Table 13, neither modification improved segmentation quality. In fact, combining both changes led to a substantial increase in HD95.

Model Variant	Overlap[↑]	Dice[↑]	HD95[↓]
MedNeXt (base)	0.921	0.920	4.983
MedNeXt + Decoder	0.951	0.899	8.451
MedNeXt + Stem	0.914	0.894	8.223
MedNeXt + Decoder + Stem	0.946	0.880	14.876

Table 13: Performance of *MedNeXt* variants on $fold_0$, trained with patch size $28 \times 256 \times 224$, 25% foreground sampling, and 50% dropout. All models used 24.5M–28.3M parameters and 223G–254G FLOPs.

A.8 Effect of Input Patch Size and Voxel Spacing

We evaluated *ResEnc* models trained with progressively lower in-plane resolutions by increasing the voxel spacing from the default $2.5 \times 0.625 \times 0.625$ mm to coarser spacings: $2.5 \times 0.97 \times 0.97$, $2.5 \times 1.0 \times 1.0$, $2.5 \times 1.25 \times 1.25$, and $2.5 \times 1.5 \times 1.5$. These correspond to downsampled input volumes of approximately $44 \times 410 \times 410$ to $44 \times 266 \times 266$.

As shown in Table 14, segmentation accuracy remained stable at moderate downsampling levels, with only minor degradation at the coarsest resolution.

Model	Patch Size	Voxel Spacing (mm)	Overlap[↑]	Dice[↑]	HD95[↓]
ResEnc	28×256×224	2.5×0.97×0.97	0.935	0.934	3.399
ResEnc	16×256×256	2.5×1.00×1.00	0.935	0.934	3.410
ResEnc	16×256×256	2.5×1.25×1.25	0.940	0.932	3.585
ResEnc	16×256×256	2.5×1.50×1.50	0.929	0.931	3.632

Table 14: Impact of voxel spacing and patch size on *ResEnc* performance ($fold_0$).

B Second Stage Semantic Segmentation Model Experiments

B.1 Sampling Policy

The second-stage segmentation model was trained with 100% foreground patch sampling, as background loss is masked out using the first-stage binary prediction. Patches are sampled by randomly selecting a center voxel from one of three annotated regions: *Right Atrium Cavity* (25%), *Left Atrium Cavity* (25%), or *Left & Right Atrium Wall* (50%). Figure 2 illustrates this sampling strategy.

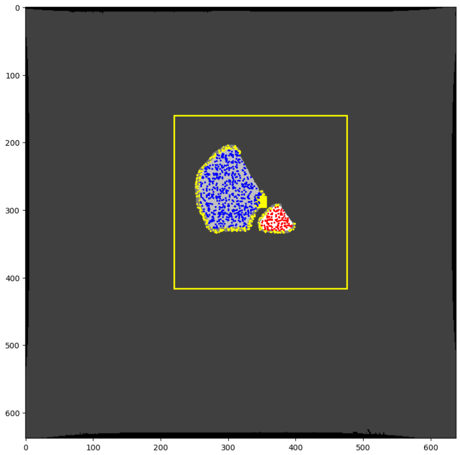


Fig. 2: Patch of size 20 × 256 × 256 centered on a voxel from the *Left & Right Atrium Wall* (yellow). Possible sampling locations are shown as colored dots: blue (Right Atrium), red (Left Atrium), yellow (Wall).

As shown in Table 15, applying this sampling strategy without masking the background leads to severe false positives—particularly in the Right Atrium—due to insufficient exposure to background voxels during training.

Model	1st Stage Mask	Wall		Right Atrium		Left Atrium	
		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc final	Yes	0.711	3.026	0.920	3.609	0.930	3.996
ResEnc	No	0.701	6.475	0.581	96.238	0.929	3.969

Table 15: Performance with and without the first-stage binary mask using 100% foreground sampling (5-fold cross-validation).

We also evaluated a depth-normalized sampling policy (*z-equal*) to counteract the spatial bias introduced by random sampling, which tends to favor central voxels in the atrial volume. As shown in Table 16, performance remained nearly unchanged, indicating that simple random sampling is sufficient.

Model	Sampling Policy	Wall		Right Atrium		Left Atrium	
		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc final	default	0.723	2.727	0.925	3.215	0.931	3.714
ResEnc	z-equal	0.722	2.740	0.925	3.240	0.932	3.690

Table 16: Comparison of default vs. z-equal sampling strategies ($fold_0$ results).

B.2 Best vs. Final Checkpoint

To assess overfitting, we compared the *ResEnc final* model's last training checkpoint (`checkpoint_final.pth`) with the best-performing checkpoint (`checkpoint_best.pth`), determined by the maximum exponential moving average of validation Dice score during training.

As shown in Figure 3, the validation loss diverges from training loss, suggesting overfitting. However, the pseudo-validation Dice score—computed from 100 validation patches—remains stable or slightly improves. The best checkpoint occurred before epoch 1000 in most folds (e.g., epochs 958, 926, 994, 950, and 284).

Despite this, Table 17 shows negligible difference in final evaluation metrics between the best and final checkpoints when tested on the full validation set.

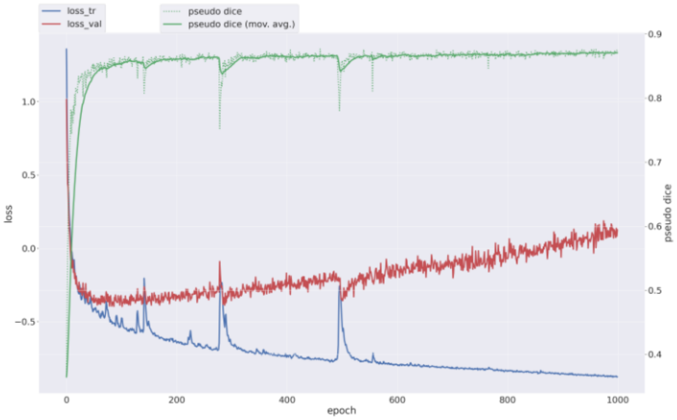


Fig. 3: Training vs. validation loss and pseudo Dice score over 1000 epochs. Despite loss divergence, Dice remains stable.

Model	Checkpoint	Wall		Right Atrium		Left Atrium	
		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc	Final	0.711	3.026	0.920	3.609	0.930	3.996
ResEnc	Best	0.713	3.052	0.921	3.565	0.930	4.083

Table 17: Performance comparison using the final vs. best checkpoint across 5-fold cross-validation.

B.3 Binary Dilation on the First-Stage Mask

We evaluated the impact of applying morphological dilation to the first-stage binary mask prior to use in second-stage training. As shown in Table 18, a dilation radius of 1–2 slightly improved Dice and HD95 metrics, while radius 3 led to marginal performance degradation. Based on these results, we selected radius 2 for the final model.

Model	Radius	Wall		Right Atrium		Left Atrium	
		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc	–	0.721	2.880	0.925	3.200	0.931	3.875
ResEnc	1	0.724	2.827	0.925	3.275	0.932	3.871
ResEnc	2	0.724	2.740	0.925	3.330	0.933	3.754
ResEnc	3	0.721	2.734	0.923	3.397	0.931	3.730

Table 18: Effect of binary dilation radius on segmentation performance ($fold_0$).

B.4 Reducing Input Patch Size

We hypothesized that a smaller spatial context might suffice for the second-stage model due to the global structure provided by the first-stage mask. As shown in Table 19, a larger patch size (20 × 256 × 256) marginally improved metrics like Wall Dice and Left Atrium HD95 but required significantly more computation. All models were trained using dilation radius 1.

Model	Patch Size	Wall		Right Atrium		Left Atrium		#Params	FLOPs
		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]		
ResEnc	16×192×192	0.723	2.772	0.925	3.266	0.931	3.893	140M	362G
ResEnc	16×256×256	0.720	2.852	0.925	3.207	0.931	3.940	140M	644G
ResEnc	20×256×256	0.724	2.827	0.925	3.275	0.932	3.871	140M	805G

Table 19: Effect of input patch size on segmentation performance and compute cost ($fold_0$). All models trained with binary dilation radius 1.

B.5 Reducing Model Size

We evaluated smaller *ResEnc* variants by capping the maximum number of feature channels. As shown in Table 20, the default configuration with a 320-channel cap achieved the best overall performance. All models used binary dilation radius 1.

Model	Feature Dim.	Wall		Right Atrium		Left Atrium	
		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc	96	0.720	2.859	0.926	3.171	0.931	3.929
ResEnc	128	0.722	2.837	0.925	3.319	0.932	4.167
ResEnc	256	0.721	2.722	0.924	3.339	0.932	3.768
ResEnc	320	0.724	2.827	0.925	3.275	0.932	3.871

Table 20: Effect of feature dimensionality on segmentation performance ($fold_0$).

B.6 Effect of Dropout

Adding 50% dropout to the second-stage model consistently reduced performance across all structures (Table 21). All models used binary dilation radius 1.

Model	Dropout	Wall		Right Atrium		Left Atrium	
		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc	0	0.724	2.827	0.925	3.275	0.932	3.871
ResEnc	50	0.705	2.892	0.920	3.324	0.922	4.188

Table 21: Effect of dropout on segmentation performance ($fold_0$).

B.7 Using Batch Dice

We tested the `batch_dice=True` setting in `SoftDiceLoss`, which computes loss across all batch elements as a single volume. As shown in Table 22, the effect was minimal, with slight improvement in HD95 for the Right Atrium. All models used binary dilation radius 2.

Model	Batch Dice	Wall		Right Atrium		Left Atrium	
		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc	No	0.724	2.744	0.925	3.325	0.933	3.754
ResEnc	Yes	0.723	2.727	0.925	3.215	0.931	3.714

Table 22: Effect of batch-level Dice loss computation ($fold_0$).

B.8 Increasing Input Patch Size

Using the batch dice model as baseline, we explored larger input patches to increase spatial context. Table 23 shows that performance remained flat while computational cost increased significantly.

Model	Patch Size	Wall		Right Atrium		Left Atrium		#Params	FLOPs
		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]		
ResEnc	20×256×256	0.723	2.727	0.925	3.215	0.931	3.714	140M	805G
ResEnc	32×256×256	0.723	2.860	0.922	3.394	0.931	3.815	140M	1289G
ResEnc	32×384×384	0.720	2.911	0.924	3.223	0.929	4.062	140M	2900G

Table 23: Effect of increasing patch size on segmentation and compute ($fold_0$).

B.9 Varying Data Augmentation

We evaluated several data augmentation configurations by adjusting the probabilities of rotation, scaling, Gaussian noise, and blur. Table 24 summarizes the key differences. The default augmentation settings consistently yielded better segmentation performance across all structures (Table 25).

Augmentation Parameter	Default	Config 1	Config 2	Config 3
Rotation Prob.	0.2	0.5	0.5	0.2
Scaling Prob.	0.2	0.4	0.4	0.0
Gaussian Noise Prob.	0.1	0.0	0.3	0.1
Gaussian Blur Prob.	0.2	0.0	0.3	0.2
Low-Res Transform Prob.	0.25	0.0	0.0	0.0

Table 24: Selected data augmentation parameters varied across configurations. All other settings matched the default.

Model	Aug. Config	Wall		Right Atrium		Left Atrium	
		Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
ResEnc	Default	0.723	2.727	0.925	3.215	0.931	3.714
ResEnc	Config 1	0.717	2.753	0.921	3.463	0.931	3.759
ResEnc	Config 2	0.718	2.855	0.922	3.329	0.930	3.799
ResEnc	Config 3	0.721	2.851	0.925	3.379	0.931	3.815

Table 25: Segmentation performance with different data augmentation settings ($fold_0$).

C Experiments with nnU-Net Style Cascaded Architecture

We evaluated nnU-Net's cascaded two-stage architecture using the *MedNeXt* model and found that it underperforms compared to both single-stage and first-stage-only baselines.

In this setup, the first-stage model is a full-resolution *MedNeXt* trained with $16 \times 256 \times 256$ patches. The second-stage model, *MedNeXt verA*, shares the same architecture but uses a reduced feature dimension (128) and a 25% foreground sampling rate. Table 26 summarizes the results.

Across all configurations, cascaded *MedNeXt verA* models failed to outperform the single-stage *MedNeXt verA* baseline and even underperformed the first-stage-only model. This suggests that concatenating first-stage predictions with the image volume—as done in the nnU-Net cascade—did not improve downstream segmentation.

Model	Stage	Patch Size	Wall		Right Atrium		Left Atrium	
			Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]	Dice[↑]	HD95[↓]
MedNeXt verA	–	16×256×256	0.724	2.843	0.926	3.031	0.932	3.940
MedNeXt	1st stage	16×256×256	0.723	3.259	0.923	3.173	0.927	4.596
MedNeXt verA	2nd stage	16×256×256	0.721	3.064	0.921	3.509	0.928	4.244
MedNeXt verA	2nd stage	16×96×96	0.725	3.277	0.921	3.287	0.927	4.440
MedNeXt verA	2nd stage	16×128×128	0.725	3.360	0.922	3.342	0.926	4.626

Table 26: Evaluation results ($fold_0$) for nnU-Net style cascaded *MedNeXt* models. All two-stage variants underperformed relative to their single-stage counterparts.

Architecture (ResEnc)	
# Stages	6
Features per stage	32, 32, 64, 96, 96, 96
Kernel sizes	(1,3,3), (3,3,3), ...
Strides	(1,1,1), (1,2,2), ...
# Blocks per stage	1, 3, 4, 6, 6, 6
# Convs per decoder stage	1, 1, 1, 1, 1, 1
Dropout probability	0.5
Normalization	InstanceNorm3d
Nonlinearity	LeakyReLU
Training Configuration	
Patch size	28×256×224
Voxel spacing (mm)	2.5×0.97×0.97 (downsampled from 2.5×0.625×0.625)
Batch size	2
Batch Dice	False
Foreground sampling likelihood	0.25
Weight init	`kaiming_normal`(1e-2)
Optimizer	SGD
Base learning rate	1e-2
Weight decay	3e-5
Momentum	0.99
Epochs	1000
Iterations per epoch	250
LR schedule	exponential decay
Loss function	SoftDiceLoss (smoothing=1e-5), CrossEntropyLoss
Data Augmentation	
Elastic deformation prob	0
Rotation prob	0.2
Rotation range	[−180°, 180°]
Scaling prob	0.2
Scaling range	[0.7, 1.4]
Gaussian noise prob	0.1
Gaussian noise variance	[0.0, 0.1]
Gaussian blur prob	0.2
Gaussian blur sigma	[0.5, 1.0]
Brightness prob	0.15
Brightness multiplier	[0.75, 1.25]
Contrast prob	0.15
Contrast range	[0.75, 1.25]
Low-res transform prob	0.25
Low-res scale	[0.5, 1.0]
Gamma transform prob	0.1
Gamma range	[0.7, 1.5]
Inverse gamma prob	0.3
Inverse gamma range	[0.7, 1.5]

Table 27: Hyperparameter and data augmentation settings for the 1st-stage ResEnc binary segmentation model. Training time: 6.25 hours on NVIDIA RTX 4090.

Architecture (ResEnc)	
# Stages	7
Features per stage	32, 32, 64, 128, 256, 320, 320, 320
Kernel sizes	(1,3,3), (1,3,3), (3,3,3), (3,3,3), (3,3,3), (3,3,3), (3,3,3)
Strides	(1,1,1), (1,2,2), (1,2,2), (2,2,2), (2,2,2), (1,2,2), (1,2,2)
# Blocks per stage	1, 3, 4, 6, 6, 6, 6
# Convs per decoder stage	1, 1, 1, 1, 1, 1, 1
Dropout probability	0.0
Normalization	InstanceNorm3d
Nonlinearity	LeakyReLU
Training Configuration	
Patch size	20×256×256
Voxel spacing (mm)	2.5×0.625×0.625
Batch size	2
Foreground sampling likelihood	1.0
Class sampling weights	Wall: 0.5, Right: 0.25, Left: 0.25
Weight init	`kaiming_normal`(1e-2)
Optimizer	SGD
Base learning rate	1e-2
Weight decay	3e-5
Momentum	0.99
Epochs	1000
Iterations per epoch	250
LR schedule	exponential decay
Loss function	SoftDiceLoss (smoothing=1e-5), CrossEntropyLoss
Batch Dice	True
Cascaded mask dilation radius	2
Data Augmentation	
Elastic deformation prob	0
Rotation prob	0.2
Rotation range	[−180°, 180°]
Scaling prob	0.2
Scaling range	[0.7, 1.4]
Gaussian noise prob	0.1
Gaussian noise variance	[0.0, 0.1]
Gaussian blur prob	0.2
Gaussian blur sigma	[0.5, 1.0]
Brightness prob	0.15
Brightness multiplier	[0.75, 1.25]
Contrast prob	0.15
Contrast range	[0.75, 1.25]
Low-res transform prob	0.25
Low-res scale	[0.5, 1.0]
Gamma transform prob	0.1
Gamma range	[0.7, 1.5]
Inverse gamma prob	0.3
Inverse gamma range	[0.7, 1.5]

Table 28: Hyperparameters and data augmentation settings for the 2nd-stage ResEnc segmentation model. The model was trained for 1000 epochs and took 11.6 hours to complete on an NVIDIA RTX 4090.

References

1. Aquaro, G.D., De Gori, C., Faggioni, L., Parisella, M.L., Cioni, D., Lencioni, R., Neri, E.: Diagnostic and prognostic role of late gadolinium enhancement in cardiomyopathies. Eur Heart J Suppl **25**(Suppl C), C130–C136 (Apr 2023)
2. Isensee, F., Faeger, P.F., Kohl, S.A.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature Methods **18**, 203–211 (2021). https://doi.org/10.1038/s41592-020-01008-z
3. Isensee, F., Wald, T., Ulrich, C., Baumgartner, M., Roy, S., Maier-Hein, K., Jaeger, P.F.: nnu-net revisited: A call for rigorous validation in 3d medical image segmentation (2024), https://arxiv.org/abs/2404.09556
4. Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. CoRR **abs/2201.03545** (2022), https://arxiv.org/abs/2201.03545
5. Mont, L., Roca-Luque, I., Althoff, T.F.: Ablation lesion assessment with mri. Arrhythmia Electrophysiology Review **11**, e02 (April 2022). https://doi.org/10.15420/aer.2021.63, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9014705
6. Roy, S., Koehler, G., Ulrich, C., Baumgartner, M., Petersen, J., Isensee, F., Jaeger, P.F., Maier-Hein, K.: Mednext: Transformer-driven scaling of convnets for medical image segmentation (2024), https://arxiv.org/abs/2303.09975
7. Xiong, Z., Xia, Q., Hu, Z., Huang, N., Bian, C., Zheng, Y., Vesal, S., Ravikumar, N., Maier, A.K., Yang, X., Heng, P., Ni, D., Li, C., Tong, Q., Si, W., Puybareau, É., Khoudli, Y., Géraud, T., Chen, C., Bai, W., Rueckert, D., Xu, L., Zhuang, X., Luo, X., Jia, S., Sermesant, M., Liu, Y., Wang, K., Borra, D., Masci, A., Corsi, C., de Vente, C., Veta, M., Karim, R., Preetha, C.J., Engelhardt, S., Qiao, M., Wang, Y., Tao, Q., Garcia, M.N., Camara, O., Savioli, N., Lamata, P., Zhao, J.: A global benchmark of algorithms for segmenting late gadolinium-enhanced cardiac magnetic resonance imaging. CoRR **abs/2004.12314** (2020), https://arxiv.org/abs/2004.12314