ENSAM: an efficient foundation model for interactive segmentation of 3D medical images

Elias Stenhede[0009−0005−2654−4553],
Agnar Martin Bjørnstad[0009−0005−4207−6278], and
Arian Ranjbar[0000−0002−0422−2255]

Medical Technology & E-health, Akershus University Hospital,
1478 Lørenskog, Norway
arian.ranjbar@medisin.uio.no

Abstract. We present ENSAM (Equivariant, Normalized, Segment Anything Model), a lightweight and promptable model for universal 3D medical image segmentation. ENSAM combines a SegResNet-based encoder with a prompt encoder and mask decoder in a U-Net-style architecture, using latent cross-attention, relative positional encoding, normalized attention, and the Muon optimizer for training. ENSAM is designed to achieve good performance under limited data and computational budgets, and is trained from scratch on under 5,000 volumes from multiple modalities (CT, MRI, PET, ultrasound, microscopy) on a single 32 GB GPU in 6 hours. As part of the CVPR 2025 Foundation Models for Interactive 3D Biomedical Image Segmentation Challenge, ENSAM was evaluated on hidden test set with multimodal 3D medical images, obtaining a DSC AUC of 2.404, NSD AUC of 2.266, final DSC of 0.627, and final NSD of 0.597, outperforming two previously published baseline models (VISTA3D, SAM-Med3D) and matching the third (SegVol), surpassing its performance in final DSC but trailing behind in the other three metrics. In the coreset track of the challenge, ENSAM ranks 5th of 10 overall and best among the approaches not utilizing pretrained weights. Ablation studies confirm that our use of relative positional encodings and the Muon optimizer each substantially speed up convergence and improve segmentation quality.

Keywords: Medical Imaging · Multimodal · Interactive Segmentation

1 Introduction

Accurate segmentation of three-dimensional (3D) or two-dimensional plus time (2D+t) medical images has become fundamental for numerous clinical tasks, including diagnosis, treatment planning, and disease monitoring. While 3D images provide much richer spatial context compared to 2D images, technical challenges arise in processing and storing large amounts of high-resolution 3D data. In the last decade, specialized deep learning model have shown success in automating segmentation tasks when trained using high-quality pixel-level labels [22].

More recently, the advent of large pretrained foundation models in natural language processing has demonstrated that models trained on massive and diverse datasets can generalize effectively to downstream tasks [7, 36, 6, 39], surpassing the performance of specialized models.

Motivated by this, foundation models for natural image segmentation has been developed, most notably SAM [20] for 2D and subsequently SAM2 [37] for 2D+t. While providing impressive segmentations for natural images, they do not immediately provide useful segmentations when applied to medical images. To address this performance gap, the 2025 CVPR Workshop on Foundation Models for Medical Vision was established, including a challenge aimed at improving segmentation accuracy on medical modalities. In this paper, we present our contribution to that effort: an efficient SAM model for medical 3D imaging.

1.1 Related work

Several previous attempts at building a SAM for medical imaging exist. MedSAM [26] adapts SAM to the medical domain by fine-tuning on medical segmentation datasets, achieving improved performance across several imaging modalities, but is limited by only supporting an initial bounding box and no subsequent clicks. It further only supports 2D slices and lacks volumetric consistency. MedSAM2 [28] adapts SAM2 for and supports segmentation of 2D+t and 3D medical images, but also lacks support for iterative segmentation refinement.

Models inspired by but not directly utilizing weights from SAM or SAM2 have also been proposed. SAM-Med3D [40] demonstrates the feasibility of training using medical data only and supports iterative refinement, but fails to generalize well to unseen classes and requires many interactions to match the performance of task-specific baselines such as nnU-Net [13]. SegVol [8] introduces more prompt types, including text, boxes, and clicks, and is trained on Computed Tomography (CT) images only.

Other models build on established segmentation architectures to enhance performance in medical imaging tasks. SegResNet [29], a U-Net-based architecture augmented with a variational decoder head for regularization, has proven highly effective, winning multiple 3D medical image segmentation challenges [32, 31, 30]. It serves as the backbone in VISTA3D [11], a model trained on both CT and MRI data, which leverages supervoxels generated by SAM during training in addition to conventional segmentation masks.

The model nnInteractive [9] advances the state-of-the-art further by supporting bounding box, click, lasso, and scribble prompts, with the latter two providing stronger guidance. It is trained on a multimodal set of 3D medical datasets using both traditional segmentation masks and supervoxels from SAM and SAM2. Unlike the other models that rely on cross-attention, nnInteractive integrates user input through dense feature maps.

1.2 Objective and contribution

As the field of interactive 3D medical image segmentation rapidly evolves, each new model often demonstrates improvements over selected baselines. However, a unified and independent comparison of these models on a common benchmark has not yet been established. This challenge addresses that gap by evaluating participants on a standardized hidden test set comprising multiple imaging modalities, enabling a fair and unbiased comparison of methods based solely on their generalization and segmentation performance.

ENSAM (Equivariant, Normalized, SAM in 3D) is designed to improve both training efficiency and segmentation accuracy, addressing the often prohibitive computational cost of training foundation models. Our objective is to achieve performance comparable to, or surpassing, current state-of-the-art models, while training on a single GPU. The name ENSAM also subtly reflects this goal, as "ensam" (Swedish spelling) means sole/lonely in Germanic languages (cognate with the English "onesome").

Our proposed model is inspired by SAM and the SegResNet architecture, and introduces several improvements, such as relative position encodings in 3D together with a normalized attention mechanism for the encoded user input, trained by a Muon optimizer in place of Adam. All modifications are aimed at accelerating training and enhancing data efficiency, while being scalable to larger setups. Furthermore, our solution targets the coreset challenge, a sub-track of the challenge, limiting the training dataset to 10% of the full set.

2 Method

The goal of the challenge is to develop a foundation model for universal medical image segmentation; that is, given a medical image, the model should segment any anatomical or pathological structure indicated by a user prompt (specifically, a bounding box or point-based clicks). Evaluation is conducted through a simulated interactive setting, in which the model receives an image during inference, with or without an initial bounding box, followed by five simulated "clicks" representing iterative corrections made by a clinician.

To address this task, we propose a model based on three components: an image encoder, a prompt encoder, and a mask decoder, configured into a U-Net architecture. User prompts are integrated into the model via cross-attention mechanisms applied at the bottleneck layer. With the chosen structure, we are capable of simultaneously training all components, end-to-end. The following subsections detail the proposed method, and an overview of the architecture is provided in Figure 1.

2.1 Image Encoder

With the advent of vision transformers, several efforts have been made to adapt transformer-based backbones for use as image encoders in segmentation models.

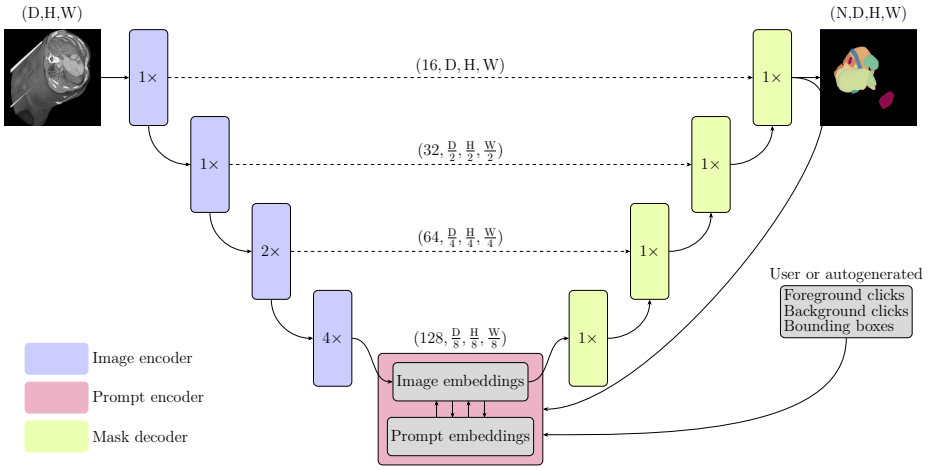


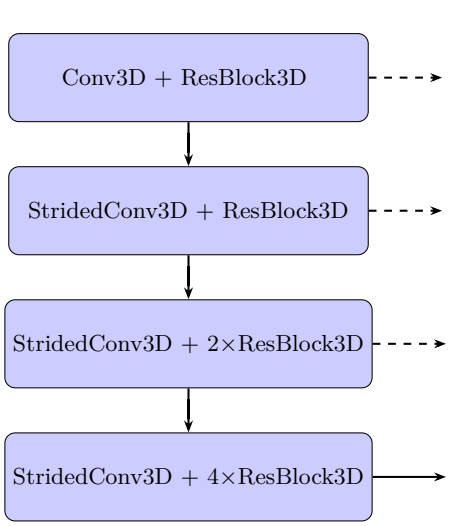
Fig. 1: Network architecture of the proposed interactive segmentation model, consisting of three main components: image encoder, prompt encoder, and mask decoder. During an interactive segmentation session on a single sample, the image encoder runs once, while the decoder updates the segmentation with each new user input.

However, most benchmarks continue to show that CNN-based encoders outperform their transformer-based counterparts [3]. We adopt an architecture based on the SegResNet model, [29], which comprises a cascade of residual blocks interleaved with downsampling layers. An overview of the image encoder is illustrated in Figure 2.

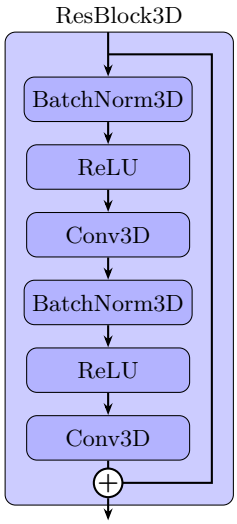
2.2 Prompt Encoder

The prompt encoder is responsible for taking input from the user and encoding it in a format compatible with the rest of the model. The prompt encoder currently supports 3D bounding boxes as well as foreground and background clicks. All user interactions are represented as unit vectors with the same dimension d as the image embeddings at the bottom latent space of the U-Net. In its current implementation, boxes are represented by a pair of unit vectors and clicks by a single vector, one for foreground and another for background clicks.

Each prompt and image embedding is associated with a 3D coordinate. This coordinate information is essential for the mask decoder to reason about spatial relations between prompts and image content. Traditionally, absolute positional information has been added element-wise to embedding vectors [20], [26]. However, absolute encoding breaks equivariance. Using methods that instead encode relative positional information has been shown to improve training efficiency and final model performance both in 1D [38] as well as 2D and 3D tasks [34].



(a) The image encoder consists of four blocks. The initial Conv3D layer transforms the image from single-channel to 16 channels. The StridedConv3D layers reduce the spatial dimensions by a factor of 2 and double the number of channels.



(b) Each residual block contains convolutional layers with skip connections. Through all layers, the channel dimension is kept constant, allowing for element-wise addition of the residual activations.

Fig. 2: Architecture overview: (a) image encoder; (b) residual block used within the encoder.

Lie Rotational Positional Encoding. To include relative positional information when computing attention between embedding vectors, the attention blocks are given pairs of embedding vectors and coordinates. Technically, positional information is encoded by applying position-dependent rotation matrices to key and query vectors. As previously noted in [34], Lie algebras provide a suitable framework in this setting, as the group of rotations SO(n) can be generated from the Lie algebra $\mathfrak{so}(n)$. In other words, for an embedding vector e_i with coordinate $p_i = (x_i, y_i, z_i)$, the corresponding rotation matrix can be written as

$$R(p_i) = \exp\left(A_x x_i + A_y y_i + A_z z_i\right). \tag{1}$$

where A_x, A_y, $A_z \in \mathfrak{so}(d)$ are learnable, skew-symmetric matrices of size $d \times d$, where d is the embedding dimension. Each matrix is parameterized by $d(d-1)/2$ values due to the skew-symmetry ($A^\top = -A$).

Letting q_i and k_j be key and query vectors with positions $p_i = (x_i, y_i, z_i)$ and $p_j = (x_j, y_j, z_j)$ respectively. The attention scores between q_i and k_j are then

calculated as

$$\begin{aligned}
\text{AttnScore}(q_i, k_j) &= (R(p_i)q_i)^\top (R(p_j)k_j) \\
&= q_i^\top R(p_i)^\top R(p_j)k_j \\
&= q_i^\top \exp\left(A_x x_i + A_y y_i + A_z z_i\right)^\top \exp\left(A_x x_j + A_y y_j + A_z z_j\right) k_j \\
&= q_i^\top \exp\left(-A_x x_i - A_y y_i - A_z z_i\right) \exp\left(A_x x_j + A_y y_j + A_z z_j\right) k_j \\
&= q_i^\top \exp\left(A_x(x_j - x_i) + A_y(y_j - y_i) + A_z(z_j - z_i)\right) k_j \\
&= q_i^\top R(p_j - p_i)k_j,
\end{aligned}$$

which shows that the attention scores depend only on the relative position $p_j - p_i$ and coincides with the standard attention calculation when $p_j = p_i$, as $\exp(0) = I$.

Normalized Attention. Recent work by Loshchilov et al. [24] introduces a normalized transformer architecture, capable of converging in 4-20 times fewer training steps compared to the standard transformer, primarily demonstrated on 1D natural language tasks. In ENSAM, we extend this approach to the 3D medical image domain, combining it with LieRE. We hypothesize that their benefits, namely faster convergence and improved numerical stability, can generalize to volumetric data.

The normalized transformer replaces traditional layer normalization (e.g., RMSNorm [42] or LayerNorm [2]) and weight decay with ℓ_2 normalization applied to all weight matrices after each optimization step. Additional ℓ_2 normalization of activations is also performed. As a result, attention and MLP outputs are constrained to lie on a unit hypersphere, which requires a modified residual update strategy. Specifically, the standard residual addition:

$$x \leftarrow x + \text{Block}(x) \tag{2}$$

is replaced by

$$x \leftarrow \text{Norm}\left(\text{Norm}(x) + \lambda(\text{Norm}(\text{nBlock}(x)) - \text{Norm}(x))\right). \tag{3}$$

In eq. (3), Norm denotes ℓ_2 normalization and $\lambda \in \mathbb{R}_+^d$ is an eigen learning rate that is learned for each block in the model. Block and nBlock denote the blocks in a transformer architecture and a normalized transformer architecture, respectively. Steps performed by cross-attention and MLP layers are performed using the same logic. This update rule can be interpreted as a constrained optimization step on the unit hypersphere, which empirically stabilizes training and accelerates convergence.

For initialization of e.g. λ, we used the values recommended by the original authors. For such implementation, specific details and theoretical justifications, we refer readers to the original study [24].

Image-Prompt Interaction. The interaction between user prompts and image embeddings builds on the original SAM model, with modifications to the attention mechanism, positional encoding, and postprocessing. Besides the image embeddings, the prompt encoder processes user inputs and, when available, segmentation logits from the previous step. These segmentation logits are downsampled using strided convolutions to align with the image embeddings and are added element-wise. The interaction between user input and the modified image embeddings follows a four-step process, using normalized attention and relative positional encoding as core components.

1. Normalized self-attention is applied to the prompt embeddings.
2. The prompt embeddings attend to the image embeddings.
3. The updated prompt embeddings are passed through a multi-layer perceptron (MLP) with ReLU activation and a hidden dimension of $2d$.
4. The image embeddings attend to the updated prompt embeddings.

All four steps incorporate residual connections on the hypersphere using Equation (3). This four-step process is repeated twice and is illustrated in Figure 3.

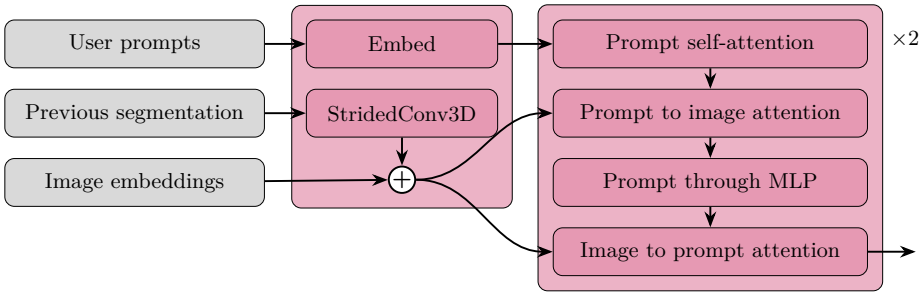


Fig. 3: The prompt encoder module encodes user input and modifies the image embeddings before it is passed to the mask decoder. If a previous segmentation exists, it is incorporated into the image embeddings via strided convolutions, allowing for iterative refinement.

2.3 Mask Decoder

The mask decoder mirrors the image encoder, except for single residual blocks per upsampling layer. The activations from skip-connections are concatenated along the channel dimension and processed by a single ResBlock3D, followed by trilinear upsampling. The final layer outputs logits with the same shape as the input, containing the segmentation mask.

2.4 Model Training and interaction simulation

During training, user interactions are simulated. If provided, initial prompts are given as bounding boxes, calculated using the ground truth labels with an added random offset, as to mimic human generation. An iterative refinement click is then placed in the middle of the largest error region. In case the largest error region is an undersegmentation, a foreground click is placed; otherwise, a background click is used. In total, a bounding box and five clicks are provided per training step.

To provide supervision for the model, we use the sum of generalized dice loss and cross-entropy, as compound loss functions have been proven to be robust in various medical image segmentation tasks [25]. Specifically, the cross-entropy carries double the weight,

$$\mathcal{L} = \mathcal{L}_{\text{Dice}} + 2 \cdot \mathcal{L}_{\text{CE}} = 1 - \frac{2\sum_i p_i g_i}{\sum_i p_i^2 + \sum_i g_i^2} - \frac{2}{N}\sum_i g_i \log p_i, \tag{4}$$

where N is the number of voxels, p denotes the prediction and g the ground-truth. The loss is averaged across all iterative steps.

To train the model in as high a resolution as possible without exceeding the GPU memory constraints, gradient accumulation and a batch size of one are used. The usage of a batch size of one is also partly motivated by the varying data shapes. A standard torch dataset/dataloader setup was used with 32 worker threads for parallel data loading. Data preprocessing and augmentation were performed on the fly within the dataset, as it did not serve as a bottleneck for the training pipeline.

Muon optimizer Following its recent success in speedrunning training of image classification and language models, [17, 23], we investigate if the Muon optimizer [18] is effective for segmentation models. The Muon optimizer has, to the best of our knowledge, not been benchmarked for segmentation tasks at the time of writing. Unlike traditional optimizers like Adam or SGD, Muon operates on 2-dimensional weight matrices. To apply Muon to ENSAM, all weights with dimension ≥ 2 is flattened beyond the first dimension. For example, 3D convolutional kernels are 5-dimensional, and needs flattening. For parameters with dimension 1, Adam is used as per usual.

Muon replaces the conventional gradient descent update with a step along $UV^\top$, where $U\Sigma V^\top$ is the singular value decomposition (SVD) of the gradient matrix. Rather than computing the full SVD, Muon employs an efficient approximation [4, 5], which has been shown to achieve similar performance [23] while significantly reducing computational cost.

2.5 Coreset selection strategy

The coreset track in the challenge required the use of no more than 10% of the total training data, corresponding to a maximum of 4,471 samples. To ensure

diverse representation under this constraint, we aimed to select an approximately equal number of samples from each dataset. In cases where a dataset contained fewer than $4471/N$ samples (where N is the number of eligible datasets), all available samples from that dataset were included.

Some datasets were excluded from the coreset selection process. The CT Aorta dataset was omitted due to apparent issues with image normalization. In addition, the microscopy datasets were excluded as many of them had issues in the provided format. Since the ground truth annotations were stored using the `uint8` format, this led to instance merging due to label value overflow.

2.6 Post-processing

Although the model is trained to segment one instance at a time, multiple instance prompts are typically provided during inference. To handle this, only one encoder pass is needed for the input, while the prompt encoder and decoder can be run in parallel for each instance. The final segmentation assigns each voxel to the instance with the highest output logit, or to the background if no logit exceeds a predefined threshold such as 0.

To ensure that each instance is assigned at least one voxel, the logits are adjusted by adding a constant inside the bounding box of any instance that initially has no assigned voxels. If every instance already has at least one voxel assigned, no changes are made to the logits.

3 Experiments

3.1 Dataset and evaluation metrics

The development set is compiled by the organizers of the CVPR 2025 Foundation Models for Interactive 3D Biomedical Image Segmentation Challenge. This includes data normalization. The development set is an extension of the CVPR 2024 MedSAM on Laptop Challenge [27], including more 3D cases from public datasets [1] and covering commonly used 3D modalities including CT, MRI, Positron Emission Tomography (PET), Ultrasound (US), and microscopy images. The hidden test set is created by a community effort where all the cases are unpublished. The annotations are either provided by the data contributors or annotated by the challenge organizer with 3D Slicer [19] and MedSAM2 [28]. In addition to using all training cases, the challenge contains a coreset track, where participants can select 10% of the total training cases for model development. The solution proposed in this paper specifically targets the latter corset track.

Each interactive segmentation is evaluated using the Dice Similarity Coefficient (DSC) and Normalized Surface Distance (NSD), which measure segmentation region overlap and boundary accuracy, respectively.

[1] A complete list is available at https://medsam-datasetlist.github.io/

Ranking of participants is performed using four metrics: Area Under the Curve (AUC) for DSC and NSD, as well as final AUC and NSD. Denoting AUC_i as the DSC after the i-th user interaction, the AUC is calculated as

$$\text{AUC\_DSC} = \frac{1}{2}\left(\text{DSC}_1 + 2 \cdot \text{DSC}_2 + 2 \cdot \text{DSC}_3 + 2 \cdot \text{DSC}_4 + \text{DSC}_5\right). \quad (5)$$

The initial bounding box prediction is excluded from this calculation, as it is optional. The same formula is used for computing AUC_NSD, mutatis mutandis. The four metrics intend to capture both the iterative refinement and final predictions.

Finally, to ensure practical applicability, inference time is capped at 90 seconds per class. Any submission exceeding this limit receives a score of zero for both DSC and NSD on the corresponding test case.

3.2 Implementation details

Preprocessing Following the practice in MedSAM [26], all images were preprocessed by the challenge organizers into `.npz` format with an intensity range normalized to $[0, 255]$. For CT images, the Hounsfield units were normalized using standard window width and level settings: soft tissue (W:400, L:40), lung (W:1500, L:-160), brain (W:80, L:40), and bone (W:1800, L:400). Subsequently, the intensity values were rescaled to the range of $[0, 255]$. For other images, the intensity values were clipped to the range between the 0.5th and 99.5th percentiles before rescaling them to the range of $[0, 255]$. If the original intensity range is already in $[0, 255]$, no preprocessing was applied.

Environment settings The development environments and requirements are presented in Table 1.

Table 1: Development environments and hardware.

Component	Specification
System	Debian 12
CPU	Intel(R) Core(TM) i9-14900KF
RAM	2×48 GB; 4800 MT/s
GPU	NVIDIA GeForce RTX 5090 32 GB
CUDA version	12.8
Programming language	Python 3.12
Deep learning framework	PyTorch 2.7.0, Torchvision 0.22.0

Training protocols Training was performed after coreset selection, which involved nearly uniform sampling across datasets. Therefore, oversampling was not

employed. Each training epoch consisted of sampling every data instance once in randomized order and generating boxes or clicks for one randomly selected labelled instance from the label data.

Part of the datasets used during training included irrelevant regions surrounding the areas of interest. To focus computational resources on relevant structures, training volumes were randomly cropped around the labelled regions with a variable margin of 1 to 64 voxels in each spatial dimension. After cropping, volumes that exceeded a predefined size threshold were downscaled via max pooling to fit within GPU memory constraints. The shapes of training volumes varied across samples. However, to ensure compatibility with the network architecture, all spatial dimensions were adjusted to be divisible by 8 by zero padding.

Following the spatial augmentations, the volumes were converted from `uint8` format to a range between $[0, 1]$, and an intensity augmentation was applied with a probability of 0.5. Specifically, one of the following was randomly applied: bias field distortion, Gaussian smoothing, or histogram shift.

Table 2: Parameters used during model training. FLOPs were calculated for one forward pass with the maximum patch volume and only one user interaction.

Parameter	**Value**
Batch size	1
Gradient accumulation steps	4
Patch size	Variable
Maximum patch volume	4,194,304 $\approx 161^3$
Simulated clicks per step	5
Total epochs	15
Optimizer	Muon and AdamW
Muon momentum	0.95
Initial learning rate	10^{-3}
Learning rate scheduler	Halved at epochs 2, 5, 10
Training time	6 hours
Loss function	Soft Dice + 2·BCE
Number of model parameters	5.5 M
Number of FLOPs	368 G

3.3 Ablations

To evaluate the contributions of LieRE and Muon, we conducted an ablation study. First, ENSAM was trained using absolute position encoding with the Adam optimizer. Next, we replaced absolute encoding with LieRE while retaining the Adam optimizer. Finally, ENSAM was trained using both LieRE and the Muon optimizer. The results can be found in Figure 4, and we note that relative

position encoding and switching optimizer improve training speed, making the model fit faster to the training data. For all experiments

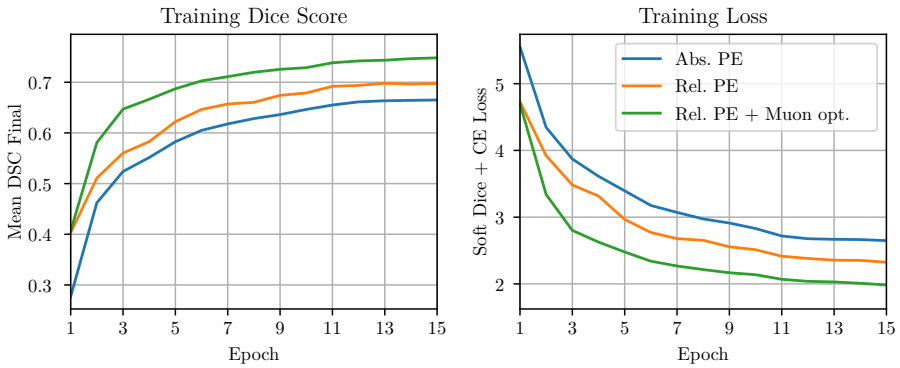


Fig. 4: Three variants of ENSAM trained on the same coreset using the same seed. Relative position encodings improve training efficiency over absolute position encodings. The Muon optimizer further improves upon the relative position encodings. Besides speeding up training, the model trained with Muon also ends up at a better final loss.

4 Results and discussion

4.1 Quantitative results on validation set

Our proposed model is benchmarked against four previously published interactive segmentation models across all five modalities, and the results are shown in Table 3. On all five modalities, either VISTA3D or SegVol obtains the highest score. Among the five modalities, ENSAM is second in ultrasound, third in MRI and microscopy, and fourth in CT.

4.2 Fair comparison and reporting standards

Common pitfalls in evaluating segmentation models include confounding performance boosters, lack of well-configured baselines, insufficient testing data, and inconsistent use of evaluation metrics [14]. In this work, the same evaluation data and metrics are used across all methods, providing a transparent and accurate depiction of each model's performance. That being said, our approach is trained from scratch, only using 10% of the full challenge dataset, without relying on pretrained model weights. Further, our method is trained on a single GPU with 32 GB of VRAM for 6 hours as opposed to the baseline methods that were originally trained using 100-1000 times more computational resources. Lastly,

Table 3: Quantitative evaluation results on the validation set for the coreset track. Note that the maximum for value for DSC AUC and NSD AUC is 4, while the maximum value for DSC Final and NSD Final is 1.

Modality	Method	DSC AUC	NSD AUC	DSC Final	NSD Final
CT	SegVol	2.98	3.12	0.75	0.78
	VISTA3D	2.81	2.84	0.72	0.73
	SAM-Med3D	2.28	2.27	0.57	0.57
	ENSAM (ours)	2.03	1.90	0.50	0.47
MRI	SegVol	2.67	3.15	0.67	0.79
	VISTA3D	2.53	2.82	0.65	0.73
	ENSAM (ours)	1.84	2.07	0.45	0.51
	SAM-Med3D	1.76	1.81	0.45	0.46
Microscopy	SegVol	2.04	3.47	0.51	0.87
	VISTA3D	1.72	2.71	0.45	0.69
	ENSAM (ours)	1.27	1.74	0.34	0.45
	SAM-Med3D	0.30	0.02	0.08	0.00
PET	SegVol	2.97	2.86	0.74	0.71
	VISTA3D	2.39	2.10	0.61	0.54
	ENSAM (ours)	2.16	1.94	0.51	0.45
	SAM-Med3D	2.13	1.82	0.53	0.46
Ultrasound	VISTA3D	2.60	2.61	0.71	0.72
	ENSAM (ours)	2.10	2.41	0.55	0.62
	SAM-Med3D	1.36	1.81	0.39	0.51
	SegVol	1.24	1.80	0.31	0.45

we do not ensemble predictions or perform augmentations during evaluation, to ensure performance is not artificially inflated in comparison to the other methods. Thus, any observed performance gains should stem from methodological advancements, and not increased compute budget, training data, or test-time tricks.

4.3 Qualitative results on validation set

In this section, we provide examples of relatively successful segmentations, as well as interesting failure cases for images in each of the five modalities. For each modality, we compare ENSAM's outputs against the all-data submissions from SAM-Med3D, VISTA3D, and SegVol. Each of which was trained on roughly ten times more annotated volumes than ENSAM.

Figure 5 presents two AbdomenAtlas CT examples: In the first, ENSAM accurately delineates the liver, spleen, and kidneys. In the second, ENSAM oversegments some parts in the left and middle parts of the slice.

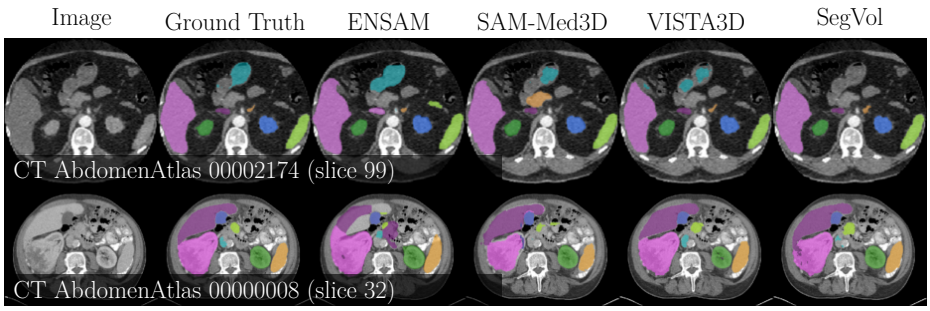


Fig. 5: **Top row:** Shows an example from the AbdomenAtlas dataset where ENSAM successfully segments the liver, spleen, and kidneys. **Bottom row:** Shows an example from the same dataset where ENSAM oversegments some parts.

Figure 6 presents two MRI slices: a slice from the Spider dataset where ENSAM captures the central vertebral bodies but fails at posterior elements; a slice from the TotalSeg dataset illustrating correct localization of large structures but failure on smaller structures.

Figure 7 illustrates PET segmentation: both ENSAM and the baseline models provide similar outputs but are often over- or undersegmented, likely due to intensity clipping during preprocessing.

Figure 8 highlights microscopy challenges: first, a slice from a microscopy volume, dense, small regions where some baseline models oversegment the image. Second, a slice with vessels where all models fail to detect thin vasculature. This might reflect the absence of microscopy data in the coreset when training

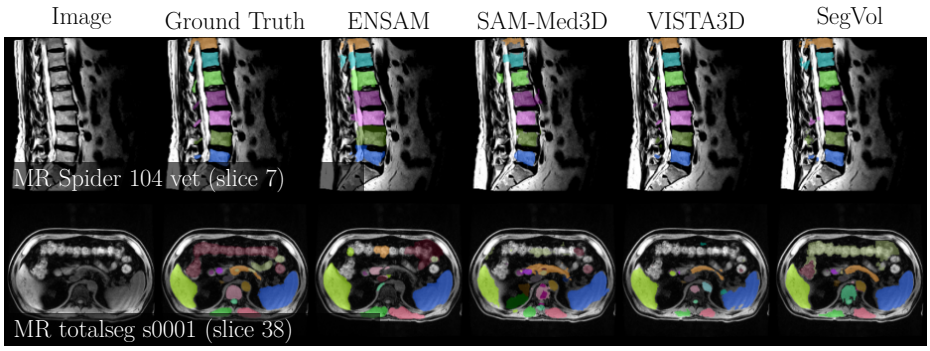


Fig. 6: **Top row:** Shows an example from the MR Spider dataset, where ENSAM successfully segments the body of the vertebra but then fails to segment the posterior. **Bottom row:** shows an example where ENSAM is relatively successful in segmenting the bottom part, whereas all models fail in segmenting the top part of the slice.

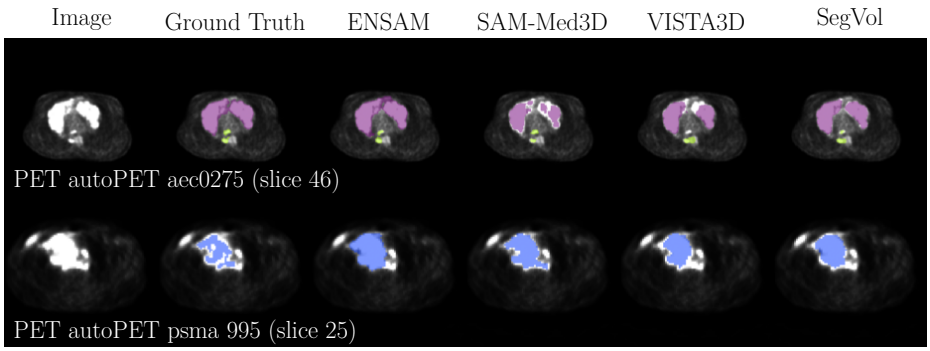


Fig. 7: **Top row:** Shows a PET image where ENSAM successfully segments high-uptake regions. **Bottom row:** Shows a PET image where ENSAM oversegments a high-uptake region. Due to the preprocessing of the PET images, all values in its neighbourhood are maximally bright, possibly making it difficult for ENSAM to distinguish the borders.

ENSAM (see Section 2.5), but also the general ambiguity in segmenting vessels using sparse prompts like single points.

Figure 9 presents two ultrasound frames: The first is a frame from a cardiac 2D+t video where ENSAM outlines the left ventricle and atrium with jagged edges from motion and speckle noise; The second shows a freehand-leg reconstruction slice segmenting three lower-leg muscles with minor boundary artifacts. For this case, SAM-Med3D's inference code crashed, and the other baseline models severely undersegmented all three muscles.

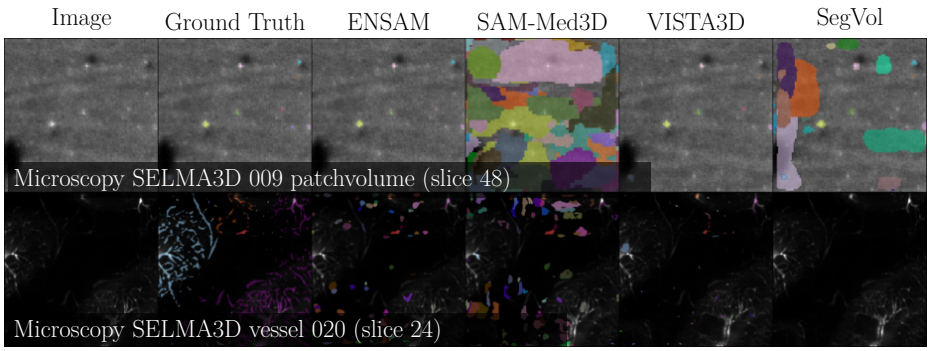


Fig. 8: **Top row:** Shows an example of a microscopy image where all labelled areas are very small, making it easy to obtain high scores for surface-distance based. It is hard to visually determine how well the model performs on this slice. In this case, SAM-Med3D and SegVol oversegment the volume. **Bottom row:** Shows an example where all models fail to segment the vessels.

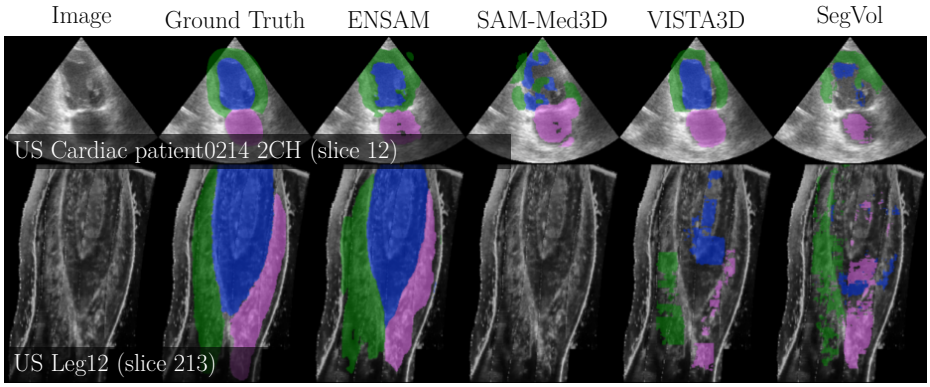


Fig. 9: **Top row:** Shows a frame in a cardiac ultrasound video, where the left atrial walls, blood volume and left atrium is annotated. The surface of ENSAM's predictions is not as smooth as the annotations. **Bottom row:** Shows a slice from a 3D volume reconstructed from 2D handheld ultrasound. Three muscles in the lower leg are annotated and segmented relatively successfully by ENSAM. For this case, the inference code provided for SAM-Med3D crashed.

4.4 Results on final testing set

Results for the hidden test set were calculated by challenge participants and baseline contributors submitting Docker containers to the challenge organizers, ensuring fair comparison.

Comparison against baseline models Qualitative results when compared to previously published baselines are summarized in Table 4. ENSAM outperforms VISTA3D and SAM-MED3D on all four metrics, and is on par with SegVol; however, it only outperforms SegVol when evaluated on the final dice score after all interactions. Notably, ENSAM performs better on the hidden test set as compared to the validation set. This likely reflects the uniform sampling strategy used during training: its broad coverage supports generalization to the diverse hidden test set but is less optimal for the unbalanced validation set.

Table 4: Quantitative evaluation results on the hidden test set for baseline submissions in the coreset track. DSC AUC and NSD AUC range from 0 to 4, higher is better.

Model	DSC AUC	NSD AUC	DSC Final	NSD Final
SegVol	2.489	2.594	0.622	0.649
ENSAM (ours)	2.404	2.266	0.627	0.597
VISTA3D	2.229	2.148	0.589	0.578
SAM-Med3D	2.110	1.914	0.536	0.487

Comparison against other challenge participants Table 5 summarizes the performance of the coreset track participants. Our submission ranked 5th out of 10 teams, placing highest among the participants not leveraging any external pretrained weights.

4.5 Limitation and future work

Limitations to ENSAM. There are several limitations to our work, pertaining to both the training and inference setup, as well as the model architecture.

First, our model was trained under constrained data and computational budgets. Leveraging the full dataset alongside increased computational resources would likely yield improved performance.

Second, while we conducted targeted ablation studies demonstrating that the Muon optimizer outperforms AdamW and that relative positional encoding is preferable to absolute positional encoding, we did not evaluate the impact of normalized attention compared to the standard attention mechanism. Additionally, due to computational constraints, we did not explore variations in model

Table 5: Quantitative evaluation results on the test set for the coreset track. The team names correspond to the names on the official leaderboard. DSC AUC and NSD AUC range from 0 to 4, higher is better.

Team	Initialization	DSC AUC	NSD AUC	DSC Final	NSD Final
aim [10]	EfficientTAM	3.104	3.232	0.801	0.838
norateam [33]	nnInteractive	2.911	2.970	0.754	0.775
yiooo [43]	VISTA3D & SAM-Med3D	2.896	2.898	0.745	0.749
lexor [1]	SegVol	2.501	2.572	0.625	0.643
ahus (ours)	-	2.404	2.266	0.627	0.597
hanglok [15]	VISTA3D	2.120	2.012	0.553	0.533
cemrg [35]	-	1.885	1.624	0.476	0.410
sail [16]	SAM-Med2D	1.654	1.593	0.413	0.398
dtftech [12]	SegVol	1.591	1.079	0.417	0.284
owwwen [21]	SAM-Med3D	0.956	0.496	0.239	0.124

size. It is therefore unlikely that the current architecture is optimal; for instance, increasing the depth of the U-Net may lead to better results.

Third, as noted by Isensee et al. [9], 2D bounding boxes can be preferable to 3D ones even for 3D segmentation tasks, and click-based prompts may convey less information compared to other prompt types. However, as the evaluation protocol for this challenge relies on 3D boxes and clicks, the current version of ENSAM supports only these prompt types.

Fourth, we have not conducted latency evaluations or user studies. To robustly assess the practical utility of interactive segmentation models, simulated user interactions alone are insufficient; real user studies are essential.

Limitations in evaluation pipeline While the test set contains unpublished medical images and annotations providing a fair comparison between models, the validation set includes mostly CT and MR. In the PET modality, a single dataset is included, and for US, two datasets are used. One of the US datasets comprises 2D+t echocardiographic videos. The microscopy modality includes just eight volumes, several of which contain only a small fraction of labelled voxels. As a result, quantitative conclusions regarding model performance on US, PET, and Microscopy should be interpreted with caution due to limited sample diversity and volume.

In addition, the CT and MR subsets of the validation set are imbalanced in terms of the number of samples taken from each dataset. This imbalance may bias the evaluation and obscure insights into how well the model generalizes across datasets within a modality. A more robust assessment, especially for modalities like CT, could be achieved through a more uniform sampling strategy that considers factors such as the number of labeled instances per dataset.

Lastly, the simulation of user input could be improved, aligning better with the use case of the model. Currently, N interactions are provided between each

refinement step, with N being equal to the number of objects of interest in the image. In reality, the user would probably want an updated segmentation after each interaction.

Future directions The field is rapidly evolving, and future work should focus on improving multiple areas. Below, we outline promising directions.

User input integration via attention: The comparative effectiveness of attention-based methods versus dense feature maps to incorporate user input warrants further investigation. In particular, prompt types such as scribbles and lassos have not yet been explored in the context of attention mechanisms in 3D medical images.

Handling anisotropic spacing and physical coordinates: The impact of anisotropic voxel spacing on position encoding remains an open question. In this work, voxel spacing is disregarded, but future studies could examine whether representing coordinates in physical units improves model performance. A related challenge is the incorporation of spatiotemporal data in training and evaluation.

Handling multiple objects: Current methods, including ENSAM and all baseline models of the challenge, treat each object instance in parallel during training and inference. Incorporating interactions between instances could likely reduce the amount of required user input and improve inference efficiency. For example, a foreground click for one instance could be interpreted as a background click for others.

5 Conclusion

In this paper, we presented ENSAM, an efficient, promptable model for universal medical image segmentation. With compute restraint and training on a coreset of the challenge data, we achieved a DSC AUC of 2.404, NSD AUC of 2.266, final DSC of 0.627 and final NSD of 0.597. Our results demonstrate that the combination of relative positional encoding and the Muon optimizer significantly improves both model performance and training efficiency. Furthermore, enabling the model to handle variable-shape inputs is critical for reducing computational overhead, particularly VRAM usage, and facilitates inference at resolutions closer to the native input scale.

Acknowledgements We thank all the data owners for making the medical images publicly available and CodaLab [41] for hosting the challenge platform. We also thank Akershus University Hospital for the funding that made this study possible.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Bibliography

[1] Aher, V., Villa, E.S., Mosquera, L.V.G., Torres, L.F.T., Verma, V.K., Ordóñez, S.A.C.: MobileSeg3D: A Lightweight Framework for Multi-Modality 3D Medical Image Segmentation (Jun 2025), https://openreview.net/forum?id=eqVsBeWMbG 18

[2] Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization (2016), https://arxiv.org/abs/1607.06450 6

[3] Bassi, P.R., Li, W., Tang, Y., Isensee, F., Wang, Z., Chen, J., Chou, Y.C., Kirchhoff, Y., Rokuss, M.R., Huang, Z., 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) 4

[4] Bernstein, J., Newhouse, L.: Modular duality in deep learning. arXiv preprint arXiv:2410.21265 (2024) 8

[5] Björck, Å., Bowie, C.: An iterative algorithm for computing the best estimate of an orthogonal matrix. SIAM Journal on Numerical Analysis **8**(2), 358–364 (1971) 8

[6] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in Neural Information Processing Systems **33**, 1877–1901 (2020) 2

[7] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 2

[8] Du, Y., Bai, F., Huang, T., Zhao, B.: Segvol: Universal and interactive volumetric medical image segmentation. In: Advances in Neural Information Processing Systems. vol. 37, pp. 110746–110783 (2024) 2

[9] Fabian, I., Maximilian, R., Lars, K., Stefan, D., Ashis, R., Florian, S., Benjamin, H., Tassilo, W., Moritz, L., Constantin, U., Jonathan, D., Ralf, F., Klaus, M.H.: nninteractive: Redefining 3D promptable segmentation. arXiv preprint arXiv:2503.08373 (2025) 2, 18

[10] Friedetzki, T., Haberzettl, L., Buttman, R., Puppe, F., Krenzer, A.: iMedSTAM: Interactive Segmentation and Tracking Anything in 3D Medical Images and Videos (Jun 2025), https://openreview.net/forum?id=1tsCloFWFT 18

[11] He, Y., Guo, P., Tang, Y., Myronenko, A., Nath, V., Xu, Z., Yang, D., Zhao, C., Simon, B., Belue, M., Harmon, S., Turkbey, B., Xu, D., Li, W.: VISTA3D: A unified segmentation foundation model for 3D medical imaging. In: Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition (2024) 2

[12] Huang, C., Huang, J., Wang, L.: From Single-Round to Sequential: Building Stateful Interactive Segmentation with SegVol and GRU Corrector (Jun 2025), https://openreview.net/forum?id=45TSSNV3sJ 18

[13] Isensee, F., Jaeger, 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**(2), 203–211 (2021). https://doi.org/10.1038/s41592-020-01008-z 2

[14] Isensee, F., Wald, T., Ulrich, C., Baumgartner, M., Roy, S., Maier-Hein, K., Jäger, P.F.: nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation (2024) 12

[15] Ji, J., Lin, T., Xiong, J., Han, T.: Enhancing a 3D Foundation Model with Gaussian Sampling for Interactive Biomedical Image Segmentation (Jun 2025), https://openreview.net/forum?id=CLk0KhDXgm 18

[16] Jo, S., Choi, A., Hong, J.H.: GAMT: A Geometry-Aware, Multi-view, Training-free Segmentation Framework for Foundation Models in Medical Imaging (Jun 2025), https://openreview.net/forum?id=DeeoLKgCVU 18

[17] Jordan, K.: 94% on cifar-10 in 3.29 seconds on a single gpu. arXiv preprint arXiv:2404.00498 (2024) 8

[18] Jordan, K., Jin, Y., Boza, V., You, J., Cesista, F., Newhouse, L., Bernstein, J.: Muon: An optimizer for hidden layers in neural networks (2024), https://kellerjordan.github.io/posts/muon/ 8

[19] Kikinis, R., Pieper, S.D., Vosburgh, K.G.: 3D Slicer: a platform for subject-specific image analysis, visualization, and clinical support, pp. 277–289. Springer (2013) 9

[20] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment anything. In: Proceedings of the International Conference on Computer Vision. pp. 4015–4026 (2023) 2, 4

[21] Lin, J., zhengdong, Ma, Z., Xiao, Y., Fu, H., Pan, Y.: Enhanced SAM-Med3D: A Robust Solution for 3D Medical Image Segmentation with Advanced Post-processing (Jun 2025), https://openreview.net/forum?id=Y3zTAf99Vr 18

[22] Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A.W.M., van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. Medical Image Analysis **42**, 60–88 (2017). https://doi.org/10.1016/j.media.2017.07.005 1

[23] Liu, J., Su, J., Yao, X., Jiang, Z., Lai, G., Du, Y., Qin, Y., Xu, W., Lu, E., Yan, J., Chen, Y., Zheng, H., Liu, Y., Liu, S., Yin, B., He, W., Zhu, H., Wang, Y., Wang, J., Dong, M., Zhang, Z., Kang, Y., Zhang, H., Xu, X., Zhang, Y., Wu, Y., Zhou, X., Yang, Z.: Muon is scalable for llm training (2025) 8

[24] Loshchilov, I., Hsieh, C.P., Sun, S., Ginsburg, B.: ngpt: Normalized transformer with representation learning on the hypersphere. In: Proceedings of the International Conference on Learning Representations (ICLR) (2025), arXiv:2410.01131 [cs.LG] 6

[25] Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., Martel, A.L.: Loss odyssey in medical image segmentation. Medical Image Analysis **71**, 102035 (2021) 8

[26] Ma, J., He, Y., Li, F., Han, L., You, C., Wang, B.: Segment anything in medical images. Nature Communications **15**, 654 (2024) 2, 4, 10

[27] Ma, J., Li, F., Kim, S., Asakereh, R., Le, B.H., Nguyen-Vu, D.K., Pfefferle, A., Wei, M., Gao, R., Lyu, D., Yang, S., Purucker, L., Marinov, Z., Staring, M., Lu, H., Dao, T.T., Ye, X., Li, Z., Brugnara, G., Vollmuth, P., Foltyn-Dumitru, M., Cho, J., Mahmutoglu, M.A., Bendszus, M., Pflüger, I., Rastogi, A., Ni, D., Yang, X., Zhou, G.Q., Wang, K., Heller, N., Papanikolopoulos, N., Weight, C., Tong, Y., Udupa, J.K., Patrick, C.J., Wang, Y., Zhang, Y., Contijoch, F., McVeigh, E., Ye, X., He, S., Haase, R., Pinetz, T., Radbruch, A., Krause, I., Kobler, E., He, J., Tang, Y., Yang, H., Huo, Y., Luo, G., Kushibar, K., Amankulov, J., Toleshbayev, D., Mukhamejan, A., Egger, J., Pepe, A., Gsaxner, C., Luijten, G., Fujita, S., Kikuchi, T., Wiestler, B., Kirschke, J.S., de la Rosa, E., Bolelli, F., Lumetti, L., Grana, C., Xie, K., Wu, G., Puladi, B., Martín-Isla, C., Lekadir, K., Campello, V.M., Shao, W., Brisbane, W., Jiang, H., Wei, H., Yuan, W., Li, S., Zhou, Y., Wang, B.: Efficient medsams: Segment anything in medical images on laptop. arXiv:2412.16085 (2024) 9

[28] Ma, J., Yang, Z., Kim, S., Chen, B., Baharoon, M., Fallahpour, A., Asakereh, R., Lyu, H., Wang, B.: Medsam2: Segment anything in 3d medical images and videos. arXiv preprint arXiv:2504.03600 (2025) 2, 9

[29] Myronenko, A.: 3d mri brain tumor segmentation using autoencoder regularization. In: International MICCAI brainlesion workshop. pp. 311–320. Springer (2018) 2, 4

[30] Myronenko, A., Siddiquee, M.M.R., Yang, D., He, Y., Xu, D.: Automated head and neck tumor segmentation from 3d pet/ct hecktor 2022 challenge report. In: 3D Head and Neck Tumor Segmentation in PET/CT Challenge, pp. 31–37. Springer (2022) 2

[31] Myronenko, A., Yang, D., He, Y., Xu, D.: Aorta segmentation from 3d ct in miccai seg. a. 2023 challenge. In: MICCAI Challenge on Segmentation of the Aorta, pp. 13–18. Springer (2023) 2

[32] Myronenko, A., Yang, D., He, Y., Xu, D.: Automated 3d segmentation of kidneys and tumors in miccai kits 2023 challenge. In: International Challenge on Kidney and Kidney Tumor Segmentation, pp. 1–7. Springer (2023) 2

[33] Ndir, T.C., Pfefferle, A., Schirrmeister, R.T.: Dynamic Prompt Generation for Interactive 3D Medical Image Segmentation (Jun 2025), https://openreview.net/forum?id=EnSf2D1blH 18

[34] Ostmeier, S., Axelrod, B., Moseley, M.E., Chaudhari, A., Langlotz, C.: LieRE: Generalizing Rotary Position Encodings (2025), arXiv:2406.10322 4, 5

[35] Qayyum, A., Mazher, M., Niederer, S.: Exploring Foundation Model Adaptations for 3D Medical Imaging: Prompt-Based Segmentation with xLSTM network (Jun 2025), https://openreview.net/forum?id=eMskbK0uSY 18

[36] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8) (2019) 2

[37] Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos. In: International Conference on Learning Representations (2025) 2

[38] Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., Liu, Y.: RoFormer: Enhanced Transformer with Rotary Position Embedding (2023), arXiv:2104.09864 4

[39] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023) 2

[40] Wang, H., Guo, S., Ye, J., Deng, Z., Cheng, J., Li, T., Chen, J., Su, Y., Huang, Z., Shen, Y., Fu, B., Zhang, S., He, J., Qiao, Y.: Sam-med3d: Towards general-purpose segmentation models for volumetric medical images. arXiv preprint arXiv:2310.15161 (2024) 2

[41] Xu, Z., Escalera, S., Pavão, A., Richard, M., Tu, W.W., Yao, Q., Zhao, H., Guyon, I.: Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. Patterns **3**(7), 100543 (2022) 19

[42] Zhang, B., Sennrich, R.: Root mean square layer normalization (2019), https://arxiv.org/abs/1910.07467 6

[43] Zhang, Z., Yu, Y., Xue, Y.: Rethinking RoI Strategy in Interactive 3D Segmentation for Medical Images (Jun 2025), https://openreview.net/forum?id=jospESnUL9 18