

Supplementary file

SAMamba3D: Adapting segment anything for generalizable three-dimensional segmentation of multiphase pore-scale images

Rui Zhang^{1,2}, Xianzhi Song^{3*}, Linqi Zhu^{1*}, Branko Bijeljic¹, Gensheng Li³ and Martin J. Blunt¹

¹ *Department of Earth Science and Engineering, Imperial College London, London SW7 2AZ, United Kingdom*

² *College of Artificial Intelligence, China University of Petroleum, Beijing 102249, China*

³ *College of Petroleum Engineering, China University of Petroleum, Beijing 102249, China*

E-mail address: r.zhang1@imperial.ac.uk (R. Zhang); songxz@cup.edu.cn (X. Song);
linqi.zhu@imperial.ac.uk (L. Zhu); b.bijeljic@imperial.ac.uk (B. Bijeljic); ligs@cup.edu.cn
(G. Li); m.blunt@imperial.ac.uk (M. J. Blunt).

*Corresponding author (ORCID: 0000-0002-6963-9262 (X. Song); 0000-0001-5725-2805 (L. Zhu))

Zhang, R., Song, X., Zhu, L., Bijeljic, B., Li, G., Blunt, M. J. SAMamba3D: Adapting segment anything for generalizable three-dimensional segmentation of multiphase pore-scale images. Advances in Geo-Energy Research, 2026, 21(2): 109-124.

The link to this file is: <https://doi.org/10.46690/ager.2026.08.03>

Appendix A: Data preprocessing

(1) Denoising

Each reconstructed 3D volume was filtered using a non-local means algorithm, which suppresses photon-counting noise and ring artifacts while preserving phase-boundary sharpness. The non-local means filter exploits self-similarity across non-adjacent voxel neighborhoods, making it effective for micro-CT data where edges carry critical physical information.

(2) Percentile-based intensity alignment

Directly combining volumes from different scanners can introduce artificial domain shifts because absolute grey-level ranges are not comparable. Each source volume I_s was aligned to a fixed training-derived reference range using Eq. (S1), where I_s is the source intensity volume, I'_s is the aligned source volume, $q_s^{(p)}$ is the p-th percentile of I_s , $q_r^{(p)}$ is the p-th percentile of the reference training distribution, $p_{low} = 1$ and $p_{high} = 99$ are the lower and upper percentile bounds:

$$I'_s = \frac{I_s - q_s^{(p_{low})}}{q_s^{(p_{high})} - q_s^{(p_{low})}} \cdot (q_r^{(p_{high})} - q_r^{(p_{low})}) + q_r^{(p_{low})} \quad (S1)$$

The reference range was defined as pooled training percentiles and was reused unchanged for validation and test data.

(3) Global standardization

After percentile alignment, all volumes were standardised using Eq. (S2). In Eq. (S2), $I(v)$ is the aligned intensity at voxel v , $\hat{I}(v)$ is the standardised intensity, μ_g is the global mean computed from the training split, σ_g is the corresponding global standard deviation and $\epsilon = 10^{-8}$ is a numerical stability constant. The same μ_g and σ_g were then applied to the validation and test sets.

$$\hat{I}(v) = \frac{I(v) - \mu_g}{\sigma_g + \epsilon} \quad (\text{S2})$$

(4) Data augmentation

To improve robustness under limited labeled data, on-the-fly stochastic augmentation was applied to each sampled 3D patch during training. The augmentation set includes axis-wise random flipping (probability 0.5 per axis), random 90° rotations in the axial plane, multiplicative brightness shift (uniform scale factor in [0.9, 1.1]), and additive Gaussian noise with $\sigma = 0.02$ (probability 0.3). Geometric transformations were applied identically to the image patch and its corresponding label mask, and nearest-neighbor interpolation is used for mask resampling.

Appendix B: Detailed SAMamba3D architecture

B1. 3D adaptation of SAM

Extending SAM from 2D natural images to volumetric pore-scale data requires changes in patch embedding, parameter adaptation, shallow feature injection, and positional encoding.

(1) Fine-grained 3D patch embedding

The original SAM encoder partitions a 2D image into 16×16 patches. In 3D pore-scale imaging, coarse tokenization can suppress structures spanning only a few voxels, including narrow pore throats and thin wetting layers. A finer $4 \times 4 \times 4$ 3D patch embedding was used:

$$T = \text{Conv3D}(X; k = 4, s = 4) \in \mathbb{R}^{B \times \frac{D}{4} \times \frac{H}{4} \times \frac{W}{4} \times C} \quad (\text{S3})$$

where $X \in \mathbb{R}^{B \times 1 \times D \times H \times W}$ is the input volume, B is the batch size, D, H, W are spatial dimensions, and $C = 768$ is the feature dimension of the ViT-B backbone.

(2) Parameter-efficient adaptation

Low-rank adaptation was used to adapt the pretrained SAM representation without full retraining (Hu et al., 2021). In Eq. (S4), Δx is the low-rank residual update to feature x , W_A and W_B are trainable low-rank matrices. For each adapted block, the residual update is:

$$\Delta x = W_B \cdot \text{Dropout}(W_A \cdot x) \cdot \frac{\alpha}{r} \quad (\text{S4})$$

where $x \in \mathbb{R}^C$, $W_A \in \mathbb{R}^{C \times r}$, $W_B \in \mathbb{R}^{r \times C}$, $r=8$ is the LoRA rank and $\alpha=16$ is the LoRA scaling coefficient. In the implementation used here, LoRA is inserted into attention projections.

(3) Early volumetric fusion

Before patch embedding, the shallowest Mamba feature map s_0 was injected into the input pathway:

$$X' = X + |\lambda| \cdot \text{Proj}[\text{Upsample}(s_0)] \quad (\text{S5})$$

where λ is a learnable scalar initialized to 0.001. This provides a high-resolution volumetric cue before tokenization and also preserves a direct pathway from early 3D features to the decoder.

(4) Anisotropy-aware positional encoding

Micro-CT data may be anisotropic, with different voxel spacing along axial and lateral directions. Instead of isotropically resampling the input volume, anisotropy was encoded directly in the positional bias:

$$b(c, z) = \sin\left[\frac{2\pi(c+1)}{C} \cdot \frac{\rho z}{S}\right] \quad (\text{S6})$$

where c is the channel index, z is the depth coordinate, S is the base spatial scale, and $\rho = \Delta z / \Delta x$ is the anisotropy ratio.

B2. Hierarchical mamba branch and global conditioning

The Mamba branch performs hierarchical 3D feature extraction and provides both volumetric context and conditioning signals for feature interaction and decoding (Gu and Dao, 2023). It outputs multi-scale representations:

$$(s_0, s_1, s_2, s_3) \quad (\text{S7})$$

The feature maps s_0, s_1, s_2, s_3 have resolutions $D/2, D/4, D/8$, and $D/16$, and channel dimensions 48, 96, 192, and 384, respectively.

(1) Global descriptor

The coarsest feature map s_3 was converted into a compact global descriptor:

$$G = \text{MLP}[\text{GAP}(s_3)] \in \mathbb{R}^{B \times d} \quad (\text{S8})$$

where $d=384$, $\text{GAP}(\cdot)$ denotes global average pooling, and $\text{MLP}(\cdot)$ is a two-layer perceptron with GeLU activation. The descriptor G was used to condition feature fusion and decoder modulation.

(2) Spatial importance map

To identify spatially informative regions, an importance map was derived from the intermediate feature map s_2 :

$$M = \sigma[\phi(s_2)] \in [0,1]^{B \times 1 \times D' \times H' \times W'} \quad (\text{S9})$$

where $\phi(\cdot)$ is a lightweight prediction head and $\sigma[\cdot]$ is the sigmoid function. In practice, this map assigns high scores to spatially informative regions, which are often concentrated near uncertain interfaces and boundary-rich areas.

(3) Global conditioning signals

From the global descriptor G , two sets of conditioning parameters were derived. First, a vector of injection strengths $\gamma \in (0,1)^K$ controls the contribution of global features at selected encoder depths. Second, FiLM parameters (γ_j, β_j) are produced for each decoder stage j :

$$\gamma = \sigma[\text{MLP}_\gamma(G)], \quad (\gamma_j, \beta_j) = \text{MLP}_j(G) \quad (\text{S10})$$

These parameters are volume-dependent and allowed the model to modulate feature processing according to the global context of the current input.

B3. Layer-wise SAM–mamba interaction

The two encoder branches interact at selected depths. Interaction is staged: early layers use one-way injection from the Mamba branch to the SAM branch, whereas deeper layers allow bidirectional exchange once the SAM features become semantically more stable.

(1) Shallow one-way injection

At early interaction depths, Mamba features were injected into the SAM branch through a dual-pathway adapter. The local pathway captured fine 3D structure using depthwise-separable 3D convolutions, and the global pathway aggregated coarse statistics through spatial pooling. The two pathways were combined using a learned gate:

$$F = w_l \cdot F_{\text{local}} + w_g \cdot F_{\text{global}} = \text{Softmax}\{W \cdot [\text{GAP}(F_{\text{local}}), F_{\text{global}}]\} \quad (\text{S11})$$

where $F_{\text{local}} \in \mathbb{R}^{B \times C_m \times D_m \times H_m \times W_m}$, $F_{\text{global}} \in \mathbb{R}^{B \times C_m}$, and W is a learned projection matrix. At these depths, reverse flow from the SAM branch to the Mamba branch was disabled.

(2) Deep bidirectional interaction

At deeper interaction depths, SAM features contain more stable boundary information and were projected back into the Mamba feature space. The reverse pathway was gated by a learnable scalar initialized to zero, so that feedback was introduced gradually during training rather than at initialization.

(3) Selective computation

The spatial importance map M derived from the global controller enables differentiated processing within the SAM encoder. Regions with high importance scores, typically phase interfaces, were processed using full attention-based encoding, while low-information regions were handled through a lightweight pathway. This routing strategy aligns computational effort with the spatial distribution of physically meaningful structures. Since informative regions occupied only a small fraction of the volume, this mechanism reduced redundant computation while preserving segmentation accuracy.

(4) Cross-scale fusion

At each interaction stage, SAM features, Mamba features, and cross-layer context were fused by:

$$F_{\text{out}} = \alpha \cdot \text{Attn}(F_{\text{sam}}) + \beta \cdot \text{SE}(\text{Proj}(F_{\text{mamba}})) + \gamma \cdot \text{CA}(\text{Proj}(F_{\text{ctx}})) \quad (\text{S12})$$

where $\alpha + \beta + \gamma = 1$ through softmax normalization. This prevented any single feature source from dominating the fused representation.

B4. FiLM-modulated multi-scale decoder

The decoder reconstructed voxel-level predictions while combining global consistency with local boundary fidelity. Starting from the coarsest scale, it progressively upsampled and

refined the feature representation using the Mamba skip features, the cross-encoder fusion features, the global conditioning signals, and the high-resolution stem feature.

(1) FiLM modulation

At decoder stage j , the Mamba skip feature s_j was modulated by the global descriptor (Perez et al., 2018):

$$s'_j = \gamma_j(G) \odot s_j + \beta_j(G), \quad (\text{S13})$$

where $\gamma_j(G)$ and $\beta_j(G)$ are channel-wise scaling and shifting terms predicted from G . The parameters were initialized so that the modulation is initially close to the identity map.

(2) High-resolution stem pathway

To preserve thin interfaces and small structures, the high-resolution stem feature F_{stem} during reconstruction was reintroduced:

$$x' = x + \lambda \cdot \text{Refine}([x, F_{\text{stem}}]), \quad (\text{S14})$$

where λ is a learnable gate initialized to zero.

The final prediction was produced by a $3 \times 3 \times 3$ convolution followed by a $1 \times 1 \times 1$ projection to the K -class output space.

Appendix C: Training and inference configuration

For reproducibility, the training and inference configuration is recorded here. SAMamba3D used 128^3 patches at stride 32 with batch size 2, the AdamW optimizer with weight decay 1×10^{-5} , automatic mixed precision, and a 200-epoch cap with early stopping at patience 20 on the source-validation Dice. Training proceeded in two stages. Stage A trained the Mamba branch, the adapters, the neck, and the decoder with the SAM encoder frozen. Stage

B fine-tuned the SAM LoRA parameters, the LayerNorm parameters, and the bidirectional gate at one-tenth of the Mamba learning rate. The backbone was an SAM ViT-B encoder adapted to 3D, the LoRA rank was 8 with scale 16, and the trainable parameter count was approximately 47.6 M. Inference used Gaussian-weighted sliding-window prediction at 0.7 overlap followed by a per-voxel argmax over the matrix, wetting-fluid and non-wetting-fluid classes, on one NVIDIA A100 80 GB GPU at about 16 GFLOPs per forward pass.

Appendix D: Architecture ablation

The component ablation removes the Mamba branch, the LoRA adapters, the cross-branch bridge, and the FiLM modulation in turn. Each variant was retrained under the same source-only protocol. The source-like water-wet case changes little across variants, while the oil-wet, mixed-wet, and scCO₂ cases show larger drops. This supports the interpretation that the added components mainly contribute under domain shift rather than by overfitting the source-like case.

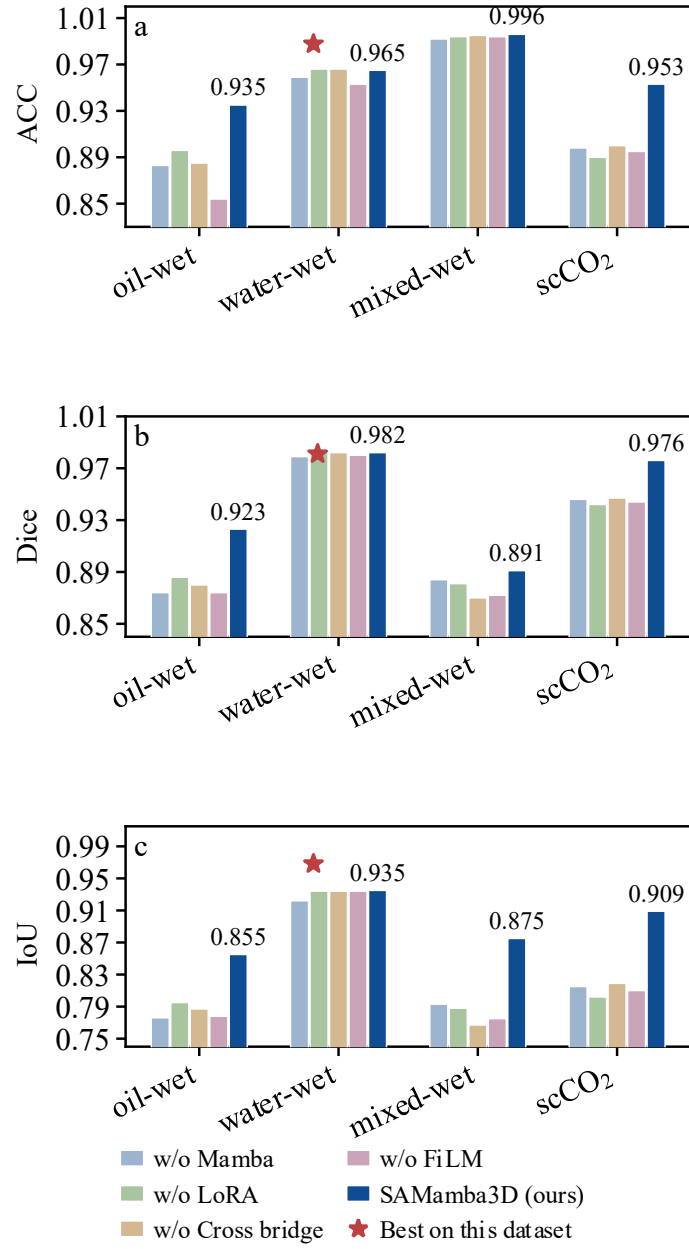


Fig. S1. Mechanism and ablation summary used to interpret the component-level contributions.

Appendix E: Input patch size sensitivity

Inference used Gaussian-weighted sliding-window prediction with 0.7 overlap and per-voxel argmax over matrix, wetting-fluid, and non-wetting-fluid classes. The main architecture comparison used 96^3 input patches so that all models were evaluated under the same patch size.

Patch-size sensitivity was treated as a deployment analysis, not as target-domain tuning. In a sweep from 48 to 256 voxels on a 512^3 volume, the Dice score increased from 0.9699 to 0.9717. Most of the gain was reached by a 96-voxel edge. Peak memory increased from 1.70 to 4.71 GB. For full-volume deployment, a 128-voxel edge retained 99.97% of the best Dice at 2.03 GB. A 96-voxel edge is a practical option when memory is constrained.

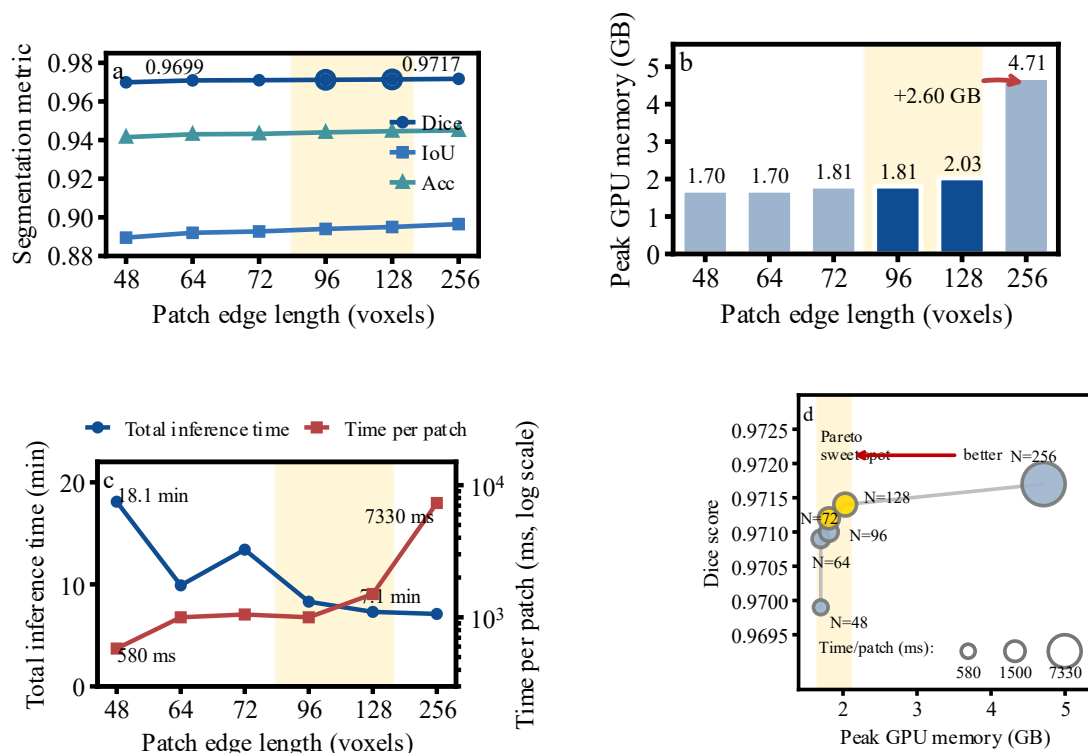


Fig. S2. Effect of inference patch size on Dice score and computational memory.

References

- Gu, A., Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. arXiv Preprint arXiv: 2312.00752, 2023.
- Hu, E. J., Shen, Y., Wallis, P., et al. LoRA: Low-rank adaptation of large language models. arXiv Preprint arXiv: 2106.09685, 2021.

Perez, E., Strub, F., de Vries, H., et al. FiLM: Visual reasoning with a general conditioning layer. Paper Presented at AAAI Conference on Artificial Intelligence (AAAI), New Orleans, Louisiana, 2-7 February, 2018.