RESTACK-MAMBA: STAGE-WISE RESIDUAL REFINEMENT FOR VOLUME ELECTRON MICROSCOPY REGISTRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Volume electron microscopy (vEM) assembles three-dimensional tissue volumes from physically cut sections, but cutting, collection, and imaging introduce rigid motion and nonuniform elastic deformation. The central challenge is to correct these section-specific distortions while preserving long-range axial continuity without propagating errors through the stack. We present ReStack-Mamba, a self-supervised framework for this coupled local-and-global registration problem. A Local Elastic Aligner first estimates in-plane correspondence between neighboring sections. Two independently trained, z-preserving StackMamba stages then use XY, XZ, and YZ context to correct successive residual errors without changing the section order. Training the second stage on errors remaining after the first allows it to learn a distinct correction; robust axial post-processing suppresses isolated outliers. Across six simulated volumes spanning five tissues and one real-world dataset, ReStack-Mamba improves XZ/YZ SSIM by approximately 10% over existing methods and achieves the best aggregate PSNR, 3D NCC, and 3D MI. Per-section and deformation-severity analyses show that the advantage persists through long stacks and becomes more pronounced under stronger distortions. The real-volume result further shows improved axial continuity, supporting robust vEM registration across diverse tissue structures.

1 INTRODUCTION

Volume electron microscopy (vEM) reconstructs cells and tissues in three dimensions at nanometre resolution, enabling analyses of organelle organization and neural connectivity that isolated images cannot support (Peddie et al., 2022; Heinrich et al., 2021; Helmstaedter et al., 2013). Advances in image acquisition and large-volume reconstruction have expanded its use in quantitative cell biology and connectomics (Kievits et al., 2022; Popovych et al., 2024). In serial-section vEM, however, these analyses depend on accurate registration of the acquired two-dimensional sections: misalignment can break membranes and distort the spatial continuity of structures across the volume (Saalfeld et al., 2012).

Registration in serial-section vEM must correct two coupled errors: nonuniform deformation within each section and inconsistency across the assembled stack. Cutting and collection introduce translation, rotation, shear, and local distortion, while imaging can change intensity and obscure correspondences (Kievits et al., 2022; Peddie et al., 2022). Even when adjacent sections align well, small residual offsets become visible as axial jitter in the XZ and YZ views (Zhang et al., 2025b). Propagating pairwise transformations through a long stack can also accumulate drift (Xin et al., 2023). A useful solution must therefore combine flexible local correction with full-stack context without amplifying errors along the section sequence.

Existing methods address parts of this problem. TrakEM2 and elastic-sheet optimization recover section-wise correspondences through feature matching and global constraints (Cardona et al., 2012; Saalfeld et al., 2012), while ssEMnet learns local elastic alignment (Yoo et al., 2017). EFSR and SEAMLeSS reduce error accumulation through serial splitting or blockwise field refinement (Xin et al., 2023; Popovych et al., 2024); Gaussian-filter registration and vEMRec aggregate neighboring fields to improve axial consistency (Zhang et al., 2025b; 2026). NeuroTrajReg instead reg-

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

ularizes continuous deformation trajectories (Zhang et al., 2025a). These approaches improve local correspondence or regularize stack-wise motion, but do not explicitly learn successive residual corrections from a locally aligned volume while jointly retaining within-section detail and axial context.

We address this gap with ReStack-Mamba, a self-supervised framework separating local elastic correction from stack-level residual refinement. The Local Elastic Aligner (LEA) estimates in-plane displacement between neighboring sections. Two z-preserving StackMamba stages integrate XY, XZ, and YZ context to refine residual errors while keeping section order fixed. The second stage learns from the first stage's remaining errors, and deterministic axial refinement removes isolated outliers. We evaluate six simulated volumes from mouse liver, pancreas, kidney, heart, and skin tissues and a real Kasthuri11 stack. On the three main datasets, SSIM improves by 6.08%–9.87% over vEMRec, with up to 1.41 dB higher PSNR. ReStack-Mamba ranks first in XY SSIM for $\alpha = 2$–5; the other tissues and real stack further support its robustness across varied cellular structures. These results show that progressive residual learning improves axial coherence across different anatomy and deformation severity.

Our contributions are:

- We formulate vEM registration as coupled local-deformation correction and axial residual refinement, and develop a self-supervised pipeline that addresses both without sequentially propagating pairwise fields.
- We design a z-preserving StackMamba that integrates XY, XZ, and YZ features to predict bounded, content-adaptive in-plane residual fields while retaining the full section sequence.
- We train two independently parameterized stages on successive residual-error distributions and validate the design on six simulated volumes and a real stack through quantitative, section-wise, and deformation-severity analyses.

2 Related Work

2.1 Local elastic registration

Classical vEM pipelines estimate correspondences between sections and optimize rigid, affine, or elastic transformations. TrakEM2 combines feature-based initialization with elastic correction in a practical reconstruction system (Cardona et al., 2012), whereas Saalfeld et al. model sections as elastic sheets coupled by matched points and optimize a template-free volume under global elastic constraints (Saalfeld et al., 2012). Learning-based registration amortizes deformation estimation: VoxelMorph predicts dense displacement fields without ground-truth flow supervision (Balakrishnan et al., 2018), and ssEMnet combines a spatial transformer with learned features for vEM registration (Yoo et al., 2017). These methods provide effective local correspondence mechanisms, but section-wise alignment alone does not explicitly resolve residual inconsistency across a long registered stack.

2.2 Stack-level consistency in vEM

Recent methods explicitly model errors that emerge across a section sequence. EFSR combines unsupervised optical flow with accumulated-error compensation and serial splitting (Xin et al., 2023). SEAMLeSS uses artifact-robust encodings, online field refinement, vector voting, and blockwise field composition for petascale alignment (Popovych et al., 2024). Gaussian-filter-based registration aggregates neighboring deformation fields to separate high-frequency sectioning distortion from slowly varying morphology (Zhang et al., 2025b), and vEMRec incorporates this strategy into a high-throughput pipeline (Zhang et al., 2026). NeuroTrajReg instead represents pixel motion continuously and optimizes deformation trajectories with neural ordinary differential equations (Zhang et al., 2025a). These methods confirm the importance of axial regularity. Our approach learns content-dependent residual correction from the locally registered volume and applies it progressively through stage-specific models.

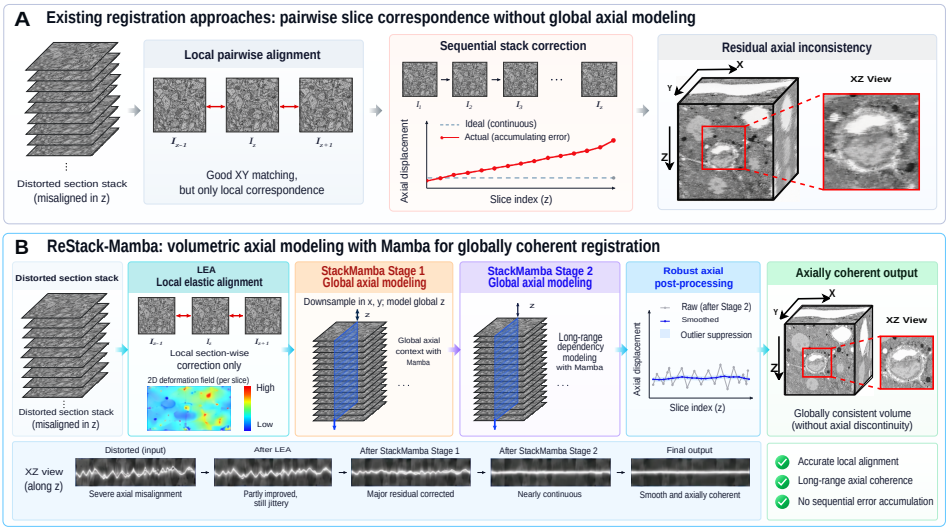

Figure 1: Comparison of conventional section-wise registration and ReStack-Mamba. ReStack-Mamba combines LEA, two independently parameterized StackMamba stages, and separate robust axial post-processing for progressive stack-level refinement.

3 METHOD

3.1 PROBLEM FORMULATION AND OVERVIEW

Given a serial-section vEM stack, the registration problem is to recover correspondence within each section while preserving the original section order and coherent structures across depth. Conventional section-wise registration (Figure 1a) aligns one section to a nearby reference. If a corrected section becomes the reference for the next, small pairwise errors can propagate through the stack; high similarity between immediate neighbors also does not guarantee continuous structures in the XZ and YZ views. ReStack-Mamba addresses these coupled local and axial errors in four steps (Figure 1b): LEA corrects local deformation, two independently trained StackMamba stages refine residual stack-level error, and robust axial refinement suppresses isolated corrections. Let $\mathcal{V} = \{I_z\}_{z=1}^{D}$ be the input stack, where $I_z \in \mathbb{R}^{H\times W}$. Each section is corrected by an in-plane field $\mathbf{u}_z \in \mathbb{R}^{2\times H\times W}$ through the differentiable backward warp

$$\mathcal{W}(I, \mathbf{u})(\mathbf{p}) = I(\mathbf{p} + \mathbf{u}(\mathbf{p}))\,, \tag{1}$$

where $\mathbf{p} = (x, y)$, with bilinear interpolation and border padding for off-grid samples.

LEA estimates $\mathbf{u}_z$ from neighboring sections and warps I_z, yielding the locally aligned stack $\mathcal{V}^{(0)} = \mathcal{V}^{\mathrm{LEA}}$. At stage $s \in \{1, 2\}$, StackMamba G_{ϕ_s} receives $\mathcal{V}^{(s-1)}$, encodes a D_p-section window as $\mathbf{F}$, fuses the XY/XZ/YZ sequences $\mathbf{S}_{xy}, \mathbf{S}_{xz}, \mathbf{S}_{yz}$ into $\mathbf{Y}$, and decodes a bounded residual field $\Delta\mathbf{u}^{(s)}$:

$$\Delta\mathbf{u}^{(s)} = G_{\phi_s}\left(\mathcal{V}^{(s-1)}\right), \qquad \mathcal{V}^{(s)} = \mathcal{W}\left(\mathcal{V}^{(s-1)}, \Delta\mathbf{u}^{(s)}\right), \quad s \in \{1, 2\}. \tag{2}$$

Stage 1 maps $\mathcal{V}^{(0)}$ to $\mathcal{V}^{(1)}$; Stage 2 is trained on fixed Stage 1 outputs and maps $\mathcal{V}^{(1)}$ to $\mathcal{V}^{(2)}$. Overlapping-window displacement predictions are blended into a full-stack field before each warp, so $\mathcal{V}^{(2)}$ denotes the merged Stage 2 volume. Axial refinement combines it with $\mathcal{V}^{(1)}$ to produce the final registered stack $\widehat{\mathcal{V}}$.

3.2 LOCAL ELASTIC ALIGNER

Figure 2a starts with a reference section I_{z-1} and a moving section I_z. Context and feature encoders feed multi-scale representations to a convolutional gated recurrent unit, which predicts flow increments at $1/16$, $1/8$, and $1/4$ resolution. A fusion head upsamples the composed flow to image

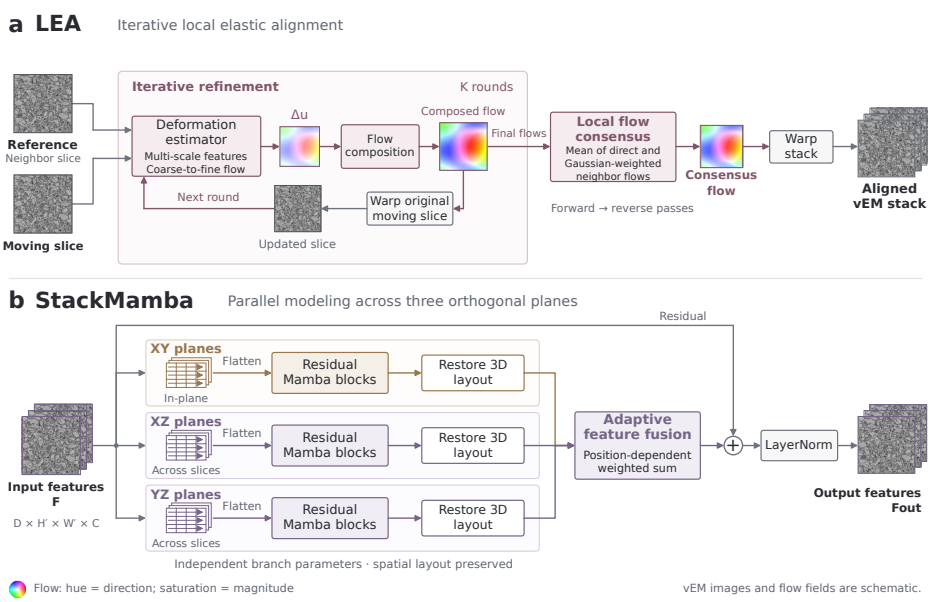

Figure 2: Data flow in the two learned modules. (a) LEA iteratively estimates and composes local flows, warps the original moving section at each round, and aggregates neighbor-induced flows before stack warping. (b) The StackMamba feature block scans XY, XZ, and YZ sequences in parallel, restores their spatial layouts, and fuses them with the encoder residual; a stage-specific decoder subsequently predicts the displacement field.

resolution. At round k, LEA composes the new increment with the preceding field and rewarps the original moving section:

$$\begin{aligned} \mathbf{u}_z^{(k)} &= \mathcal{W}\Big(\mathbf{u}_z^{(k-1)}, \Delta\mathbf{u}_z^{(k)}\Big) + \Delta\mathbf{u}_z^{(k)}, \\ J_z^{(k)} &= \mathcal{W}\Big(I_z, \mathbf{u}_z^{(k)}\Big). \end{aligned} \tag{3}$$

The updated section $J_z^{(k)}$ enters the next refinement round. After the final round, Gaussian-weighted consensus combines the direct and neighbor-induced fields into $\overline{\mathbf{u}}_z^{\mathrm{LEA}}$; forward and reverse passes reduce dependence on the starting end of the stack (Appendix S5.2). Warping every original section with its consensus field yields the first registered stack:

$$\mathcal{V}^{(0)} = \big\{\mathcal{W}\big(I_z, \overline{\mathbf{u}}_z^{\mathrm{LEA}}\big)\big\}_{z=1}^{D}. \tag{4}$$

For self-supervised training, I_{z+1} provides a second similarity target. The LEA objective is

$$\mathcal{L}_{\mathrm{LEA}} = \mathcal{L}_{\mathrm{tri}} + \lambda_{\mathrm{LEA}}\mathcal{L}_{\mathrm{LEA-reg}}, \tag{5}$$

where $\mathcal{L}_{\mathrm{tri}}$ maximizes Pearson correlation with both adjacent sections and $\mathcal{L}_{\mathrm{LEA-reg}}$ penalizes first-order variation of the iterative flow updates. Appendix Figure S5 and Appendix S5.2 give the estimator architecture and full inference procedure.

3.3 Z-PRESERVING STACKMAMBA

LEA corrects local geometry, but section-to-section jitter can remain. StackMamba therefore analyzes a D_p-section window jointly: the XY branch models in-plane morphology, whereas XZ and YZ branches expose axial discontinuities. Mamba's input-dependent state transitions model these sequences with linear complexity in scan length (Gu & Dao, 2023). Plane-specific scans, rather than a generic image scan, keep within-section appearance and axial continuity distinct (Liu et al., 2024).

Figure 2b shows the feature block within one StackMamba stage. The encoder uses strides of $(1, 2, 2)$, downsampling only in-plane dimensions while retaining the full axial extent. With a

channel-last layout, its feature tensor is $\mathbf{F}^{(s)} \in \mathbb{R}^{B\times D_p\times H'\times W'\times C}$. Reshaping this tensor by plane produces three sequences:

$$\mathbf{S}_{xy}^{(s)} \in \mathbb{R}^{(BD_p)\times(H'W')\times C}, \quad \mathbf{S}_{xz}^{(s)} \in \mathbb{R}^{(BH')\times(D_pW')\times C}, \quad \mathbf{S}_{yz}^{(s)} \in \mathbb{R}^{(BW')\times(D_pH')\times C}. \tag{6}$$

The xy sequence captures within-section structure; the xz and yz sequences traverse depth. Each branch has independent residual Mamba parameters. Its output is restored to the common 3D layout before position-dependent gating and the encoder skip connection:

$$\begin{aligned}
\mathbf{H}_a^{(s)} &= \mathrm{Restore}_a\Big[\mathbf{S}_a^{(s)} + \mathrm{Mamba}_a\Big(\mathrm{LN}(\mathbf{S}_a^{(s)})\Big)\Big], \\
\boldsymbol{\alpha}^{(s)} &= \mathrm{softmax}\Big(g[\mathbf{H}_{xy}^{(s)};\mathbf{H}_{xz}^{(s)};\mathbf{H}_{yz}^{(s)}]\Big), \\
\mathbf{Y}^{(s)} &= \mathrm{LN}\left(\mathbf{F}^{(s)} + \sum_{a\in\{xy,xz,yz\}} \alpha_a^{(s)}\mathbf{H}_a^{(s)}\right).
\end{aligned} \tag{7}$$

The stage-specific decoder maps the fused output features to a bounded two-channel in-plane displacement field,

$$\Delta\mathbf{u}^{(s)} = u_{\max}\tanh\Big(d_{\phi_s}(\mathbf{Y}^{(s)})\Big). \tag{8}$$

After blending overlapping-window displacement predictions into a full-stack field, the two stage updates are

$$\begin{aligned}
\mathcal{V}^{(1)} &= \mathcal{W}\Big(\mathcal{V}^{(0)}, \Delta\mathbf{u}^{(1)}\Big), \\
\mathcal{V}^{(2)} &= \mathcal{W}\Big(\mathcal{V}^{(1)}, \Delta\mathbf{u}^{(2)}\Big).
\end{aligned} \tag{9}$$

The first stage corrects the LEA output; the second takes the frozen first-stage result as input and estimates only its remaining distortion. Appendix Figure S6 provides the complete architecture.

3.4 Stage-wise residual learning and objective

The two stages share an architecture but not parameters. Stage 1 is trained on $\mathcal{V}^{(0)}$ patches; after it is frozen, its full-volume output $\mathcal{V}^{(1)}$ supplies Stage 2 training inputs. The second stage thus targets the remaining error. Each stage is optimized independently using

$$\begin{aligned}
\mathcal{L}_{\mathrm{stack}}^{(s)} = \lambda_{\mathrm{sim}}\mathcal{L}_{\mathrm{sim}}^{(s)} + \lambda_{\mathrm{id}}\mathcal{L}_{\mathrm{id}}^{(s)} + \lambda_{\mathrm{mag}}\mathcal{L}_{\mathrm{mag}}^{(s)} + \lambda_{\mathrm{grad}}\mathcal{L}_{\mathrm{grad}}^{(s)} \\
+ \lambda_{\mathrm{anchor}}\mathcal{L}_{\mathrm{anchor}}^{(s)} + \lambda_{z1}\mathcal{L}_{z1}^{(s)} + \lambda_{z2}\mathcal{L}_{z2}^{(s)}.
\end{aligned} \tag{10}$$

The similarity term maximizes multi-scale local normalized cross-correlation between adjacent refined sections. The identity, magnitude, and spatial-gradient terms limit unnecessary or irregular corrections; the anchor term constrains boundary and mean displacement; and the first- and second-order axial terms suppress section-to-section jitter. They act on successive residual fields along the section axis:

$$\begin{aligned}
\mathcal{L}_{z1}^{(s)} &= \frac{1}{D_p-1}\sum_{z=1}^{D_p-1}\|\Delta\mathbf{u}_{z+1}^{(s)} - \Delta\mathbf{u}_z^{(s)}\|_1, \\
\mathcal{L}_{z2}^{(s)} &= \frac{1}{D_p-2}\sum_{z=1}^{D_p-2}\|\Delta\mathbf{u}_{z+2}^{(s)} - 2\Delta\mathbf{u}_{z+1}^{(s)} + \Delta\mathbf{u}_z^{(s)}\|_1.
\end{aligned} \tag{11}$$

Definitions of the remaining loss terms and their weights are reported in Appendices S5.3 and S1.

3.5 Robust axial refinement

Overlapping Stage 2 predictions are first blended with separable Gaussian patch weights. Let $\mathcal{V}^{(2)}$ denote the blended volume, $\mathcal{V}^{(1)}$ the Stage 1 input, $G_{\sigma_z}*_z$ axial Gaussian smoothing, and $\mathcal{T}$ a radius-2 trimmed-mean operator. The final volume is

$$\widehat{\mathcal{V}} = (\mathcal{T}\circ\mathcal{T})\Big[\mathcal{V}^{(1)} + G_{\sigma_z}*_z\Big(\mathcal{V}^{(2)} - \mathcal{V}^{(1)}\Big)\Big], \qquad \sigma_z = 0.35. \tag{12}$$

Each trimmed-mean pass discards the smallest and largest values in a five-section axial neighborhood. These deterministic inference-time operations suppress isolated axial outliers while retaining the Stage 1 image content; Appendix S5.4 gives the exact operators and boundary handling.

Table 1: Quantitative comparison of various methods on three simulated vEM datasets. (The best values are shown in bold, and the second-best values are underlined.)

Dataset	Metric	TrakEM2	SEAMLeSS	EFSR	vEMRec	Ours
jrc_mus_liver	PSNR (↑)	20.598	21.192	17.961	<u>23.515</u>	**24.923**
	SSIM (↑)	0.3846	0.3961	0.3522	<u>0.5806</u>	**0.6379**
	NCC (↑)	0.7564	0.7703	0.5091	<u>0.8666</u>	**0.9041**
	MI (↑)	0.6895	0.7576	0.3479	<u>1.0720</u>	**1.1121**
jrc_mus_pancreas	PSNR (↑)	20.392	21.123	15.801	<u>24.295</u>	**24.728**
	SSIM (↑)	0.4148	0.4363	0.3436	<u>0.6314</u>	**0.6729**
	NCC (↑)	0.8155	0.8389	0.3082	<u>0.9258</u>	**0.9351**
	MI (↑)	0.7176	0.7778	0.1562	<u>1.1239</u>	**1.2019**
jrc_mus_liver3	PSNR (↑)	22.100	21.863	13.759	<u>26.228</u>	**26.386**
	SSIM (↑)	0.5144	0.5145	0.3625	<u>0.7156</u>	**0.7591**
	NCC (↑)	0.9194	0.9133	0.2563	<u>0.9704</u>	**0.9724**
	MI (↑)	0.9775	0.9635	0.2205	<u>1.3630</u>	**1.4456**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate ReStack-Mamba on six simulated vEM volumes, including two liver volumes and volumes from kidney, heart, pancreas, and skin (Xu et al., 2021; Ackerman et al., 2024), together with one real-world Kasthuri11 stack (Kasthuri et al., 2015). The simulated volumes provide undeformed references for quantitative evaluation, while the real stack is evaluated qualitatively under acquisition artifacts. All methods use identical input stacks and evaluation protocols. Dataset dimensions and deformation settings are reported in Appendix S1 and Table S1.

We report PSNR and SSIM on the XY, XZ, and YZ views, and compute normalized cross-correlation (NCC) and mutual information (MI) over each complete volume. XY metrics measure section-wise fidelity, whereas XZ and YZ expose axial discontinuity. Higher values indicate better registration for all metrics. Exact definitions and aggregation rules are provided in Appendix S1.

LEA and the two StackMamba stages are trained sequentially, with Stage 2 optimized on fixed Stage 1 outputs. During inference, overlapping predictions are blended before robust axial refinement. Preprocessing, patch sizes, optimization schedules, loss weights, stage-wise training, and inference settings are detailed in Appendix S1.

4.2 QUANTITATIVE EVALUATION

We compare ReStack-Mamba with vEMRec (Zhang et al., 2026), SEAMLeSS (Popovych et al., 2024), TrakEM2 (Cardona et al., 2012), and EFSR (Xin et al., 2023) under the same deformation, preprocessing, and evaluation protocol. Table 1 reports the three simulated volumes with the largest axial SSIM gains over vEMRec. PSNR and SSIM are aggregated across the three orthogonal views using the number of evaluated slices as weights, whereas NCC and MI are computed over the complete 3D volume.

ReStack-Mamba achieved the best performance on all four metrics for each of the three simulated datasets, consistently outperforming vEMRec, the strongest baseline. Compared with vEMRec, ReStack-Mamba yielded relative SSIM gains of 6.08%–9.87% across the three anatomies and a maximum PSNR improvement of 1.41 dB on jrc_mus_liver. These gains were accompanied by improvements in both NCC and MI, indicating closer agreement with the reference volumes across complementary similarity measures. The consistent advantage across all evaluated metrics and anatomies demonstrates that the improvement is not confined to a single metric or dataset, supporting the effectiveness of ReStack-Mamba for volumetric reconstruction. Complete results for all five methods on the remaining simulated volumes are provided in Appendix S2, Table S2; PSNR and SSIM results for individual views across all six datasets are provided in Tables S3 and S4, respectively.

Table 2: Ablation study of core components on the simulated jrc_mus_liver dataset. (The best values are shown in bold, and the second-best values are underlined.)

Variant	PSNR ↑			SSIM ↑			NCC ↑	MI ↑
	XY	XZ	YZ	XY	XZ	YZ		
LEA only	<u>23.352</u>	<u>24.231</u>	<u>24.536</u>	0.6232	0.5788	0.5751	0.8795	<u>1.0861</u>
+ Stage 1	23.331	24.205	24.502	<u>0.6252</u>	0.6095	<u>0.6118</u>	<u>0.8798</u>	1.0822
+ Stage 2	23.333	24.206	24.503	<u>0.6252</u>	<u>0.6115</u>	0.6112	<u>0.8798</u>	1.0833
+ Robust axial post-processing	**24.267**	**25.004**	**25.300**	**0.6369**	**0.6380**	**0.6386**	**0.9041**	**1.1121**

Error accumulation across the section sequence. Sequential registration can propagate local errors through a long section stack. We therefore compute SSIM independently for each XY section and plot the scores against section index. Figure 5 shows jrc_mus_liver, jrc_mus_pancreas, and jrc_mus_liver3, The remaining results are shown in Appendix Figure S1. The fixed first boundary section is excluded for every method. Points show individual-section scores, and curves show a centered 21-section moving average. ReStack-Mamba achieves higher mean per-section SSIM than vEMRec. This sustained advantage across stacks with different lengths and anatomy supports the intended role of full-stack residual refinement: it corrects remaining section-wise errors without introducing cumulative drift.

Robustness to deformation severity. We evaluate robustness on six simulated datasets by scaling fixed deformation fields with $\alpha \in \{1, 2, 3, 4, 5\}$ (Figure 6 and Appendix Figure S2). This increases displacement magnitude while preserving anatomy and deformation patterns, with each method's configuration fixed across levels. The results show that vEMRec achieves the highest mean XY-view SSIM at $\alpha = 1$, whereas ReStack-Mamba ranks first at every higher level, exceeding vEMRec by 0.0401–0.0432 at $\alpha = 5$. Across the full range, ReStack-Mamba's mean SSIM decreases by 13.8%–18.7%, compared with 20.3%–27.1% for vEMRec, demonstrating greater robustness to increasing displacement under this protocol. This robustness may be attributed to stack-level residual correction, which uses cross-section context to constrain residual misalignment and limit its impact on reconstruction quality under large deformations.

4.3 Qualitative evaluation

Figures 3 and 4 compare registration quality on simulated and real vEM data, respectively. On simulated data, competing methods leave slice-to-slice discontinuities along membranes and tissue boundaries in axial views, whereas ReStack-Mamba better matches the undeformed reference while preserving local morphology, consistent with Table 1. Additional comparisons are provided in Appendix S3, Figure S3. On the real Kasthuri11 dataset, all methods process the same prepared volume and are compared at matched anatomical locations using the same contrast window. ReStack-Mamba reduces abrupt section-wise breaks in both highlighted regions while preserving visible tissue and organelle contours. These observations suggest that its benefits extend to real data with acquisition artifacts.

4.4 Ablation study on simulated mouse liver

Dataset and protocol. We use jrc_mus_liver for the component ablation and jrc_mus_liver3 for the stage-design study. All variants start from the same LEA output and use identical settings; we report view-specific PSNR and SSIM together with volume-level NCC and MI.

Contribution of individual components. Table 2 evaluates the components cumulatively. The third row is the complete learned model, whereas the last row adds robust axial post-processing at inference and therefore represents the full pipeline.

Stage 1 raises XZ/YZ SSIM from 0.5788/0.5751 to 0.6095/0.6118, indicating that orthogonal StackMamba branches correct axial errors left by LEA. Stage 2 changes these scores only marginally before post-processing. Robust axial post-processing then raises them to 0.6380/0.6386 (10.23%/11.05% above LEA) and improves XY/XZ/YZ PSNR by 0.92/0.77/0.76 dB. NCC and MI

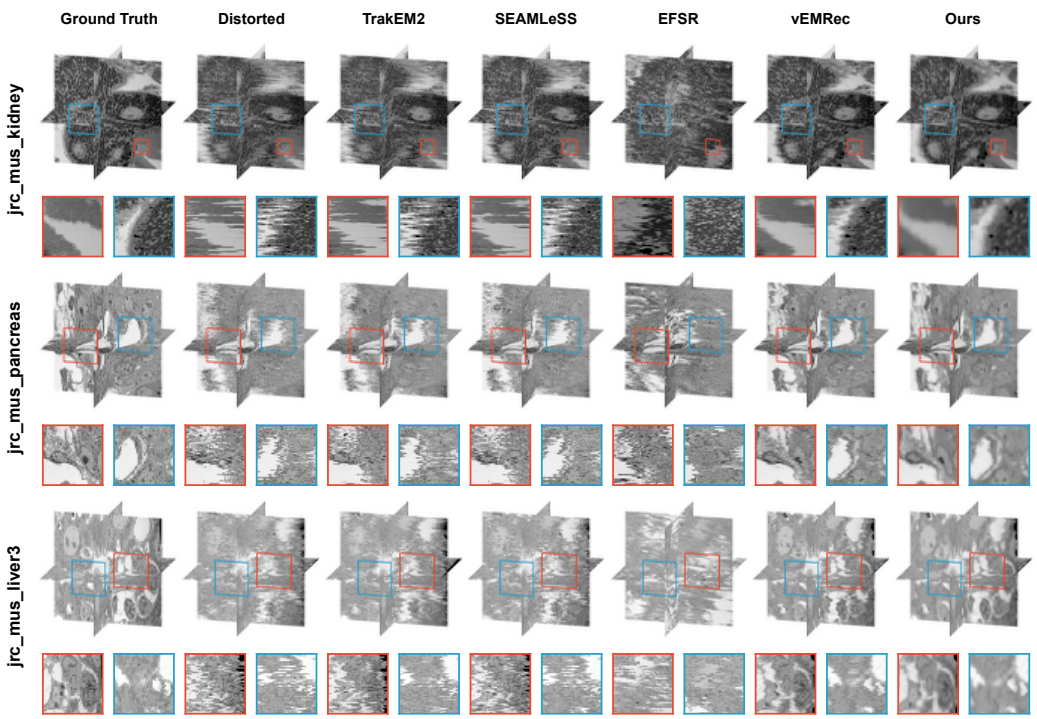

Figure 3: Visual comparison of registration methods on the jrc_mus_kidney, jrc_mus_pancreas, and jrc_mus_liver3 datasets.

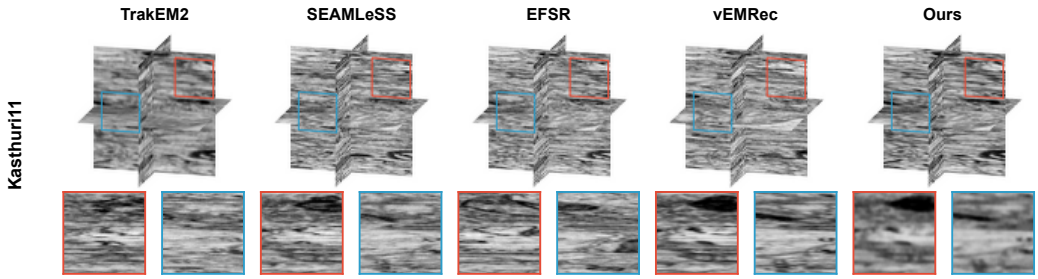

Figure 4: Qualitative comparison on the real Kasthuri11 vEM volume. Red and blue boxes mark the shared regions enlarged below the orthogonal views; all methods use the same anatomical location and contrast window.

increase from 0.8795/1.0861 to 0.9041/1.1121. These results show complementary roles: plane-aware learning reduces structured misalignment, while robust axial refinement removes isolated errors.

Effect of stage-specific training. To test whether Stage 2 learns a distinct residual correction, we compare one-stage StackMamba, Stage 1 applied twice with shared weights, and independently trained Stage 1 and Stage 2 on jrc_mus_liver3 (Table 3). The shared-weight variant controls for the number of refinement passes. Relative to one stage, sharing raises XZ/YZ SSIM from 0.7310/0.7405 to 0.7395/0.7499; independent training further raises them to 0.7419/0.7511 and gives the highest MI (1.4223 versus 1.4154). Although one stage has the highest NCC (0.9709), the independent stages yield the strongest axial structural similarity and information agreement, indicating useful correction of residual errors rather than repetition of Stage 1. Matched outputs appear in Appendix S4 and Figure S4.

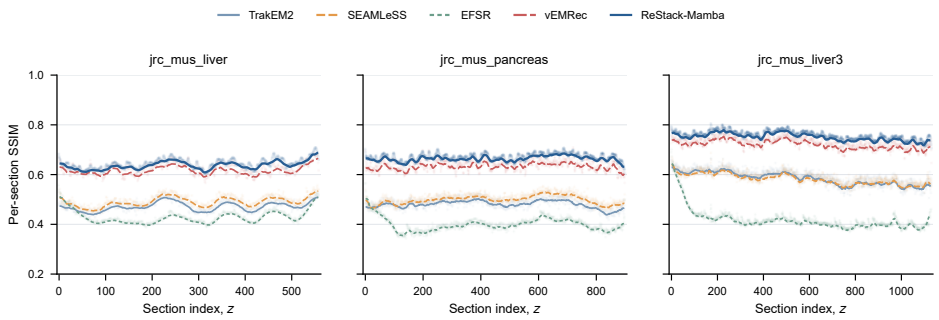

Figure 5: Per-section XY SSIM on the three datasets used for the main comparison. Points denote individual sections, and curves show centered 21-section moving averages. The fixed first boundary section is excluded for every method.

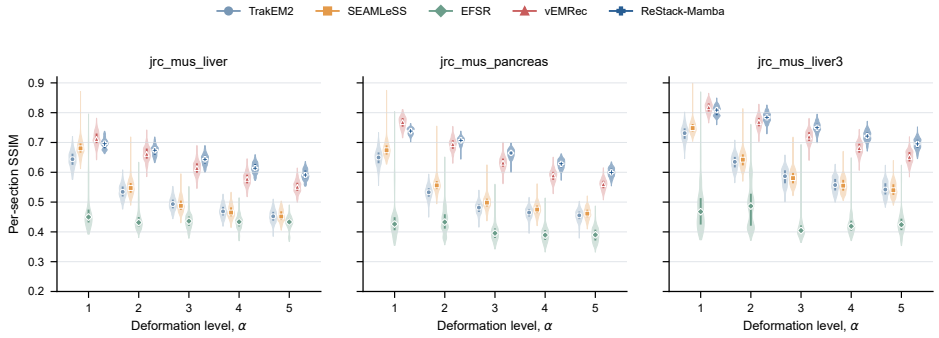

Figure 6: Robustness to deformation severity on jrc_mus_liver, jrc_mus_pancreas, and jrc_mus_liver3. Violins show per-section XY SSIM at each deformation level; symbols and bars indicate the median and interquartile range.

Table 3: Ablation study of stage-wise refinement on the simulated jrc_mus_liver3 dataset. (The best values are shown in bold, and the second-best values are underlined.)

Variant	XZ SSIM ↑	YZ SSIM ↑	3D NCC ↑	3D MI ↑
One-stage StackMamba	0.7310	0.7405	**0.9709**	1.4039
Stage 1 applied twice (shared)	<u>0.7395</u>	<u>0.7499</u>	<u>0.9708</u>	<u>1.4154</u>
Stage 1 + Stage 2 (independent)	**0.7419**	**0.7511**	0.9683	**1.4223**

5 Conclusion

We presented ReStack-Mamba, a self-supervised framework for Volume Electron Microscopy Registration. Its central contribution is to decouple local elastic correction from full-stack residual refinement, allowing section-specific deformation and long-range axial inconsistency to be addressed at their appropriate spatial scales. The z-preserving StackMamba design retains the section sequence and integrates XY, XZ, and YZ context, while independently trained stages adapt to successive residual-error distributions rather than repeatedly applying one mapping. Across six simulated datasets, ReStack-Mamba achieved the best aggregate PSNR, SSIM, 3D NCC, and 3D MI, with approximately 10% higher axial SSIM than existing registration methods. Per-section analysis showed that these gains persist through long stacks without cumulative drift, and deformation-severity experiments demonstrated an increasing advantage under stronger distortions. On the real Kasthuri11 volume, ReStack-Mamba also produced more continuous axial structures while preserving major anatomical contours. Together, these results establish stage-wise residual learning as an effective and promising strategy for accurate, robust, and axially coherent vEM registration.

AI USE STATEMENT

In this work, we used generative AI tools to assist with feedback on experimental design and the interpretation and presentation of experimental results. We additionally used generative AI tools for drafting and editing parts of the manuscript, improving writing clarity and organization, literature search, identifying relevant references, and formatting citations. All AI-assisted technical statements were manually reviewed and verified by the authors. All reported quantitative results were obtained from author-run experiments and were not generated or modified by generative AI. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

David Ackerman, Emma Avetissian, Christopher K. E. Bleck, et al. Data release: High-resolution imaging and segmentation of P7 mouse tissue microarchitecture using FIB-SEM and machine learning. *bioRxiv*, 2024. doi: 10.1101/2024.09.05.611438.

Guha Balakrishnan, Amy Zhao, Mert R. Sabuncu, John Guttag, and Adrian V. Dalca. An unsupervised learning model for deformable medical image registration. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9252–9260, 2018.

Albert Cardona, Stephan Saalfeld, Johannes Schindelin, Ignacio Arganda-Carreras, Stephan Preibisch, Mark Longair, Pavel Tomancak, Volker Hartenstein, and Rodney J. Douglas. TrakEM2 software for neural circuit reconstruction. *PLOS ONE*, 7(6):e38011, 2012. doi: 10.1371/journal.pone.0038011.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Larissa Heinrich, Davis Bennett, David Ackerman, Woohyun Park, John Bogovic, Nils Eckstein, et al. Whole-cell organelle segmentation in volume electron microscopy. *Nature*, 599(7883): 141–146, 2021. doi: 10.1038/s41586-021-03977-3. URL https://doi.org/10.1038/s41586-021-03977-3.

Moritz Helmstaedter, Kevin L. Briggman, Srinivas C. Turaga, Viren Jain, H. Sebastian Seung, and Winfried Denk. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, 500(7461):168–174, 2013. doi: 10.1038/nature12346. URL https://doi.org/10.1038/nature12346.

Narayanan Kasthuri, Kenneth J. Hayworth, Daniel R. Berger, et al. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015. doi: 10.1016/j.cell.2015.06.054.

Arent J. Kievits, Ryan Lane, Elizabeth C. Carroll, and Jacob P. Hoogenboom. How innovations in methodology offer new prospects for volume electron microscopy. *Journal of Microscopy*, 287(3): 114–137, 2022. doi: 10.1111/jmi.13134. URL https://doi.org/10.1111/jmi.13134.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. VMamba: Visual state space model. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/baa2da9ae4bfed26520bb61d259a3653-Abstract.html.

Christopher J. Peddie, Christel Genoud, Anna Kreshuk, et al. Volume electron microscopy. *Nature Reviews Methods Primers*, 2(1):51, 2022. doi: 10.1038/s43586-022-00131-9. URL https://doi.org/10.1038/s43586-022-00131-9.

Sergiy Popovych, Thomas Macrina, Nico Kemnitz, Manuel Castro, Barak Nehoran, Zhen Jia, J. Alexander Bae, Eric Mitchell, Shang Mu, Eric T. Trautman, Stephan Saalfeld, Kai Li, and H. Sebastian Seung. Petascale pipeline for precise alignment of images from serial section electron microscopy. *Nature Communications*, 15:289, 2024. doi: 10.1038/s41467-023-44354-0.

Stephan Saalfeld, Richard Fetter, Albert Cardona, and Pavel Tomancak. Elastic volume reconstruction from series of ultra-thin microscopy sections. *Nature Methods*, 9(7):717–720, 2012. doi: 10.1038/nmeth.2072.

Tong Xin, Yanan Lv, Haoran Chen, Linlin Li, Lijun Shen, Guangcun Shan, Xi Chen, and Hua Han. A novel registration method for long-serial section images of EM with a serial split technique based on unsupervised optical flow network. *Bioinformatics*, 39(8):btad436, 2023. doi: 10.1093/bioinformatics/btad436.

C. Shan Xu, Song Pang, Gleb Shtengel, et al. An open-access volume electron microscopy atlas of whole cells and tissues. *Nature*, 599(7883):147–151, 2021. doi: 10.1038/s41586-021-03992-4.

Inwan Yoo, David G. C. Hildebrand, Willie F. Tobin, Wei-Chung Allen Lee, and Won-Ki Jeong. ssEMnet: Serial-section electron microscopy image registration using a spatial transformer network with learned features. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 249–257. Springer, 2017.

Zhenbang Zhang, Jingtong Feng, Hongjia Li, Haythem El-Messiry, Zhiqiang Xu, and Renmin Han. Unsupervised trajectory optimization for 3d registration in serial section electron microscopy using neural ODEs. In *Advances in Neural Information Processing Systems*, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b9ecf4d84999a61783c360c3782e801e-Abstract-Conference.html.

Zhenbang Zhang, Hongjia Li, Zhiqiang Xu, Wenjia Meng, and Renmin Han. A gaussian filter-based 3d registration method for series section electron microscopy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(1):1156–1164, 2025b. doi: 10.1609/aaai.v39i1.32103.

Zhenbang Zhang, Hongjia Li, Zhongjun Yang, Zhiqiang Xu, Duanchen Sun, Xin Gao, Fa Zhang, and Renmin Han. vEMRec: High-resolution volume electron microscopy reconstruction based on structure-preserving and high-fidelity 3d alignment. *Advanced Science*, 13(23):e19098, 2026. doi: 10.1002/advs.202519098.

Table S1: Datasets used in the experiments.

Dataset	Sections	Section size	α	Evaluation
jrc_mus_liver	558	800×800	4	Quantitative and qualitative
jrc_mus_kidney	1,387	800×800	3	Quantitative and qualitative
jrc_mus_heart	1,061	800×800	3	Quantitative and qualitative
jrc_mus_pancreas	898	800×800	3	Quantitative and qualitative
jrc_mus_skin	1,231	800×800	3	Quantitative and qualitative
jrc_mus_liver3	1,127	800×800	3	Quantitative and qualitative
Kasthuri11	1,024	1,024×1,024	–	Qualitative only

S1 Additional experimental details

S1.1 Datasets and evaluation protocol

Table S1 summarizes the six simulated volumes used for quantitative evaluation. Each volume contains 800×800-pixel sections; the number of sections varies with the specimen. The simulator independently samples a smooth in-plane elastic field for every section except the first, using a fixed random seed of 42 and Gaussian scale $\sigma = 0.08$. The primary jrc_mus_liver experiment uses deformation magnitude $\alpha = 4$, whereas the five OpenOrganelle volumes use the common $\alpha = 3$ simulation. This procedure preserves an undeformed reference for direct evaluation, and all methods receive the same distorted stack for a given specimen. The real Kasthuri11 experiment uses the same 1,024-section, 1,024×1,024-pixel central crop described in the real-data subsection of the main paper and is evaluated qualitatively because no undeformed reference is available.

S1.2 Metrics and aggregation

PSNR and SSIM are computed independently on the XY, XZ, and YZ views. The XY view measures section-wise image fidelity, whereas XZ and YZ directly expose discontinuities along the sectioning axis. The aggregate PSNR and SSIM values in the main paper are weighted by the number of evaluated images in each view,

$$\overline{m} = \frac{N_{xy} m_{xy} + N_{xz} m_{xz} + N_{yz} m_{yz}}{N_{xy} + N_{xz} + N_{yz}}, \tag{S1}$$

where m denotes PSNR or SSIM and N_a is the number of images in view a. NCC and MI are computed once over the complete registered and reference volumes; MI is reported in nats. Higher values indicate better registration for all reported metrics.

S1.3 Training and inference settings

LEA is trained on 256×256 triplet patches for 80 epochs with Adam, a batch size of 4, an initial learning rate of 10^{-3}, three recurrent flow-refinement iterations, and $\lambda_{\mathrm{LEA}} = 18$. The learning rate is reduced by a factor of ten after 90% of training. During stack inference, locally estimated displacement fields are combined by Gaussian-weighted neighborhood consensus, and forward and reverse passes reduce dependence on traversal direction.

Each StackMamba stage is trained independently for 24 epochs on 16×256×256 patches. The encoder width is $C = 24$, each orthogonal branch contains two residual Mamba blocks, and the predicted in-plane residual displacement is bounded by $u_{\max} = 0.5$ pixels. We use AdamW with a learning rate of 5×10^{-6} and weight decay 10^{-5}. The loss weights $(\lambda_{\mathrm{sim}}, \lambda_{\mathrm{id}}, \lambda_{\mathrm{mag}}, \lambda_{\mathrm{grad}}, \lambda_{\mathrm{anchor}}, \lambda_{z1}, \lambda_{z2})$ are $(1.0, 1.0, 0.50, 0.80, 0.30, 0.40, 0.80)$. At inference, overlapping predictions are blended with separable Gaussian patch weights. The Stage 2 image residual is then smoothed along z with $\sigma_z = 0.35$, followed by two radius-2 trimmed-mean passes.

S1.4 Stage-wise optimization protocol

The full training and inference sequence is as follows:

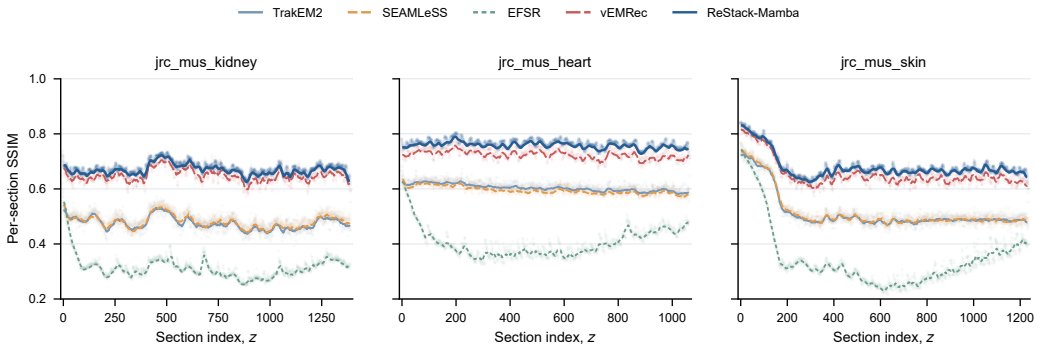

Figure S1: Per-section XY SSIM on the three datasets reported in the supplementary comparison. Points denote individual sections, and curves show centered 21-section moving averages. The fixed first boundary section is excluded for every method.

1. Train LEA on neighboring-section triplets using the self-supervised triplet-similarity and flow-regularity objective.
2. Apply the trained LEA to the complete distorted stack to obtain the locally aligned volume $\mathcal{V}^{(0)}$.
3. Train StackMamba Stage 1 on patches sampled from $\mathcal{V}^{(0)}$, then freeze its parameters and infer the full Stage 1 volume $\mathcal{V}^{(1)}$.
4. Initialize a second StackMamba with the same architecture but independent parameters, and train it on patches sampled from the fixed Stage 1 output $\mathcal{V}^{(1)}$.
5. Infer the Stage 2 residual field, blend overlapping patch predictions, warp $\mathcal{V}^{(1)}$, and apply the separate deterministic axial post-processing to obtain the final volume.

This ordering ensures that Stage 2 is optimized for the residual-error distribution left by Stage 1 rather than being a repeated application of the same learned mapping.

S1.5 Comparison controls

For each simulated dataset, all methods are evaluated against the same undeformed reference and receive exactly the same distorted input. PSNR, SSIM, NCC, and MI are computed with a shared evaluation pipeline. The supplementary qualitative plate uses the same image index and crop for every method within a dataset and view. Its contrast window is estimated once from the corresponding ground-truth crop and then applied unchanged to every method, preventing method-specific display adjustment.

S2 Complete quantitative results

The main paper reports the three volumes with the largest axial SSIM gain over vEMRec. Table S2 provides the complementary five-method comparison on jrc_mus_kidney, jrc_mus_heart, and jrc_mus_skin. ReStack-Mamba improves the weighted SSIM over vEMRec by 2.87%, 5.64%, and 3.78% on these datasets, respectively. The corresponding weighted PSNR gains are 0.08, 0.51, and 0.29 dB. Improvements in 3D NCC and MI are also consistent across the three volumes.

Figure S1 reports per-section XY SSIM on jrc_mus_kidney, jrc_mus_heart, and jrc_mus_skin. ReStack-Mamba attains higher mean per-section SSIM than vEMRec on these datasets (0.6686 vs. 0.6479, 0.7574 vs. 0.7230, and 0.6785 vs. 0.6536, respectively) and is higher on 1308/1386, 1056/1060, and 1180/1230 evaluated sections.

Figure S2 reports robustness across deformation levels on jrc_mus_kidney, jrc_mus_heart, and jrc_mus_skin. ReStack-Mamba exceeds vEMRec from $\alpha = 3$ onward on jrc_mus_kidney and jrc_mus_skin and from $\alpha = 2$ onward on jrc_mus_heart. At the strongest deformation ($\alpha = 5$), its

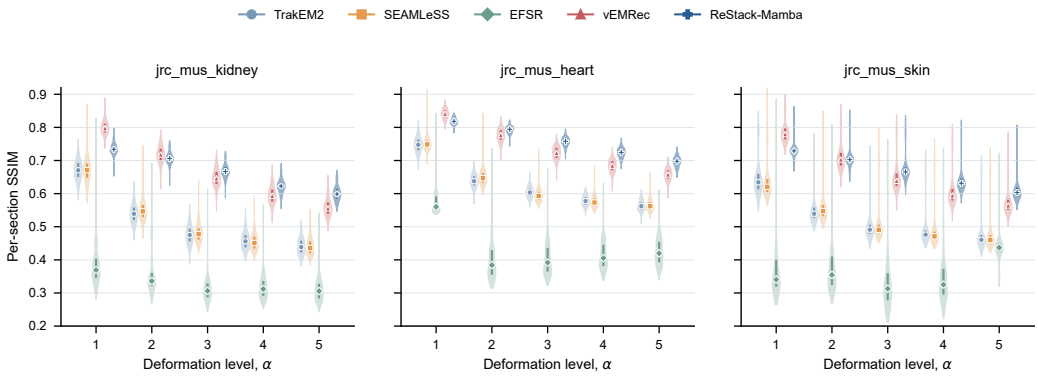

Figure S2: Deformation robustness on jrc_mus_kidney, jrc_mus_heart, and jrc_mus_skin. Violin distributions show per-section XY SSIM for deformation levels $\alpha = 1$–5; markers denote the median and vertical segments denote the interquartile range. The fixed first boundary section is excluded for every method.

Table S2: Quantitative comparison of various methods on the three additional simulated datasets not included in the main comparison. (The best values are shown in bold, and the second-best values are underlined.)

Dataset	Metric	TrakEM2	SEAMLeSS	EFSR	vEMRec	ReStack-Mamba
jrc_mus_kidney	PSNR (↑)	20.228	20.343	13.467	<u>24.363</u>	**24.445**
	SSIM (↑)	0.4147	0.4258	0.2582	<u>0.6502</u>	**0.6689**
	NCC (↑)	0.8654	0.8659	0.1888	<u>0.9497</u>	**0.9512**
	MI (↑)	0.7309	0.7454	0.0558	<u>1.1192</u>	**1.1450**
jrc_mus_heart	PSNR (↑)	22.438	21.866	13.092	<u>26.450</u>	**26.964**
	SSIM (↑)	0.5651	0.5578	0.3746	<u>0.7279</u>	**0.7689**
	NCC (↑)	0.9550	0.9483	0.5757	<u>0.9824</u>	**0.9847**
	MI (↑)	0.9051	0.8760	0.2821	<u>1.1677</u>	**1.2362**
jrc_mus_skin	PSNR (↑)	20.795	20.840	13.418	<u>24.792</u>	**25.084**
	SSIM (↑)	0.4729	0.4809	0.3112	<u>0.6585</u>	**0.6834**
	NCC (↑)	0.9235	0.9229	0.5058	<u>0.9697</u>	**0.9719**
	MI (↑)	0.7268	0.7343	0.2052	<u>1.0245</u>	**1.0775**

mean per-section SSIM is 0.6006, 0.6970, and 0.6208, compared with 0.5571, 0.6579, and 0.5836 for vEMRec, respectively. The widening advantage at higher deformation levels demonstrates that the stack-level refinement remains effective as section-wise distortion becomes more severe.

Tables S3 and S4 report view-specific PSNR and SSIM on all six simulated volumes. Relative to vEMRec, the XZ/YZ SSIM gains are 11.91%/12.61% on jrc_mus_liver, 2.53%/2.67% on jrc_mus_kidney, 6.13%/6.33% on jrc_mus_heart, 7.56%/7.46% on jrc_mus_pancreas, 3.76%/3.80% on jrc_mus_skin, and 7.33%/7.35% on jrc_mus_liver3. ReStack-Mamba therefore produces the highest axial-view SSIM in every case and improves XZ/YZ PSNR on all six volumes. On jrc_mus_kidney and jrc_mus_liver3, vEMRec obtains slightly higher XY PSNR, while ReStack-Mamba retains higher XY SSIM. This distinction supports the intended role of stack-level refinement: improving axial continuity without requiring every in-plane metric to increase.

Table S3: Comparison of PSNR across different views on all simulated datasets. (The best values are shown in bold, and the second-best values are underlined.)

Dataset	View	TrakEM2	SEAMLeSS	EFSR	vEMRec	ReStack-Mamba
jrc_mus_liver	XY	19.981	20.640	17.549	<u>22.672</u>	**24.267**
	XZ	20.613	21.314	17.631	<u>23.878</u>	**25.004**
	YZ	21.015	21.456	18.578	<u>23.740</u>	**25.300**
jrc_mus_kidney	XY	20.008	20.125	13.187	**24.201**	<u>24.165</u>
	XZ	20.508	20.660	13.613	<u>24.606</u>	**24.883**
	YZ	20.331	20.404	13.807	<u>24.400</u>	**24.494**
jrc_mus_heart	XY	22.366	21.846	12.911	<u>26.388</u>	**26.828**
	XZ	22.476	21.900	12.882	<u>26.550</u>	**27.064**
	YZ	22.497	21.859	13.543	<u>26.432</u>	**27.045**
jrc_mus_pancreas	XY	20.000	20.763	15.345	<u>23.856</u>	**24.164**
	XZ	20.563	21.320	15.729	<u>24.890</u>	**25.517**
	YZ	20.661	21.330	16.384	<u>24.192</u>	**24.574**
jrc_mus_skin	XY	20.718	20.770	13.264	<u>24.687</u>	**24.902**
	XZ	20.933	21.019	13.488	<u>25.011</u>	**25.367**
	YZ	20.776	20.768	13.585	<u>24.734</u>	**25.080**
jrc_mus_liver3	XY	21.815	21.578	13.231	**26.016**	<u>25.992</u>
	XZ	22.435	22.212	13.425	<u>26.612</u>	**27.073**
	YZ	22.167	21.916	14.835	<u>26.143</u>	**26.253**

Table S4: Comparison of SSIM across different views on all simulated datasets. (The best values are shown in bold, and the second-best values are underlined.)

Dataset	View	TrakEM2	SEAMLeSS	EFSR	vEMRec	ReStack-Mamba
jrc_mus_liver	XY	0.4705	0.4906	0.4284	<u>0.6150</u>	**0.6369**
	XZ	0.3549	0.3607	0.3271	<u>0.5701</u>	**0.6380**
	YZ	0.3544	0.3656	0.3242	<u>0.5671</u>	**0.6386**
jrc_mus_kidney	XY	0.4755	0.4800	0.3129	<u>0.6480</u>	**0.6686**
	XZ	0.3562	0.3721	0.2034	<u>0.6451</u>	**0.6614**
	YZ	0.3678	0.3853	0.2183	<u>0.6592</u>	**0.6768**
jrc_mus_heart	XY	0.6039	0.5949	0.4052	<u>0.7231</u>	**0.7574**
	XZ	0.5335	0.5257	0.3446	<u>0.7211</u>	**0.7653**
	YZ	0.5454	0.5406	0.3639	<u>0.7408</u>	**0.7877**
jrc_mus_pancreas	XY	0.4800	0.4988	0.3988	<u>0.6325</u>	**0.6637**
	XZ	0.3743	0.3974	0.3088	<u>0.6272</u>	**0.6746**
	YZ	0.3822	0.4052	0.3165	<u>0.6344</u>	**0.6817**
jrc_mus_skin	XY	0.5140	0.5164	0.3453	<u>0.6538</u>	**0.6786**
	XZ	0.4400	0.4521	0.2833	<u>0.6614</u>	**0.6863**
	YZ	0.4426	0.4551	0.2864	<u>0.6627</u>	**0.6879**
jrc_mus_liver3	XY	0.5842	0.5816	0.4163	<u>0.7196</u>	**0.7505**
	XZ	0.4614	0.4625	0.3186	<u>0.7082</u>	**0.7601**
	YZ	0.4693	0.4720	0.3306	<u>0.7174</u>	**0.7701**

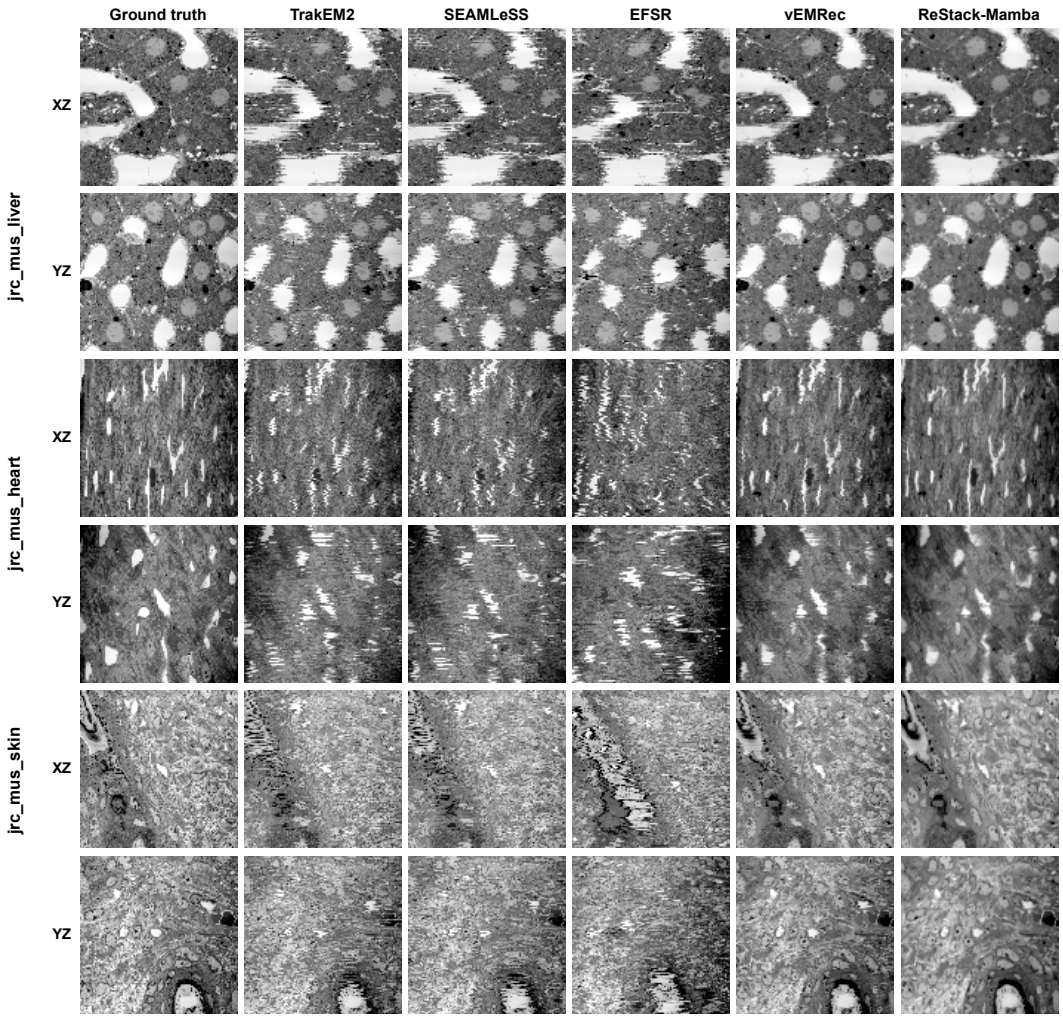

Figure S3: Axial-view comparison on the three simulated datasets omitted from the main qualitative figure. All methods use the same location, crop, and contrast window within each dataset and view.

S3 ADDITIONAL QUALITATIVE COMPARISON

Figure S3 complements the main qualitative comparison with the three simulated volumes not jointly visualized there. We show matched XZ and YZ locations for every method and use a single ground-truth-derived contrast window within each dataset and view. TrakEM2, SEAMLeSS, and EFSR retain pronounced axial jitter in these locations. vEMRec substantially improves continuity, while ReStack-Mamba more closely follows the reference trajectories of tissue boundaries and internal structures. The differences are most evident on jrc_mus_liver and jrc_mus_heart and are consistent with the view-specific SSIM values in Table S4.

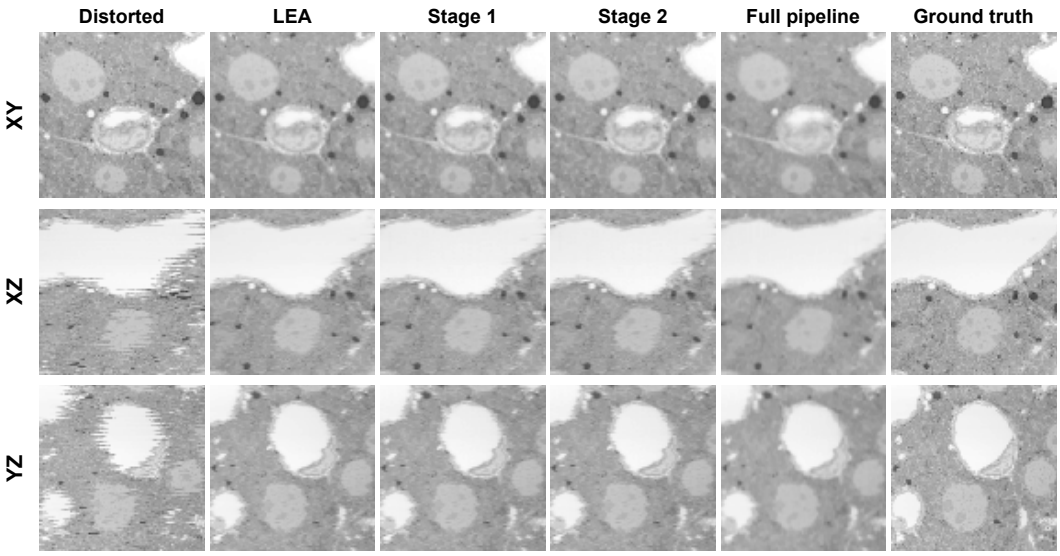

Figure S4: Intermediate outputs of the ReStack-Mamba pipeline on jrc_mus_liver. All columns show the same XY, XZ, and YZ location.

S4 Stage-wise ablation and intermediate registration results

Table 3 isolates stage-specific learning on jrc_mus_liver3. All variants start from the same LEA output and use the same architecture and inference settings, with robust axial post-processing disabled. The one-stage model is applied once, the shared two-pass variant applies the Stage 1 weights twice, and the independent two-stage variant trains Stage 2 on the fixed outputs of Stage 1. Applying Stage 1 twice increases XZ/YZ SSIM from 0.7310/0.7405 to 0.7395/0.7499. The independent Stage 2 further improves these values to 0.7419/0.7511 and raises MI from 1.4154 to 1.4223, while NCC decreases slightly from 0.9708 to 0.9683. These results show that stage-specific learning provides a modest but consistent refinement of axial alignment over weight sharing. Figure S4 visualizes the output after each component on jrc_mus_liver at a fixed location. LEA removes most of the simulated local deformation. Stage 1 substantially improves continuity in XZ and YZ, consistent with the component ablation reported in the main paper. Stage 2 makes a smaller residual correction, and the separate deterministic axial post-processing suppresses the remaining isolated section-wise variation. The final output is smoother along the sectioning axis; therefore, fine axial texture should be interpreted together with the quantitative fidelity metrics rather than from visual continuity alone.

S5 Detailed method formulation

S5.1 Optional edge-based rigid initialization

LEA targets local elastic distortion rather than very large rigid offsets. For stacks containing substantial rotations or translations, we optionally initialize registration using normalized edge maps. For section I_z, Gaussian smoothing followed by Sobel filtering gives

$$E_z = \mathrm{clip}\left(\frac{\sqrt{(S_x * G_\sigma * I_z)^2 + (S_y * G_\sigma * I_z)^2}}{q_{0.9}}, 0, 1 \right), \tag{2}$$

where $q_{0.9}$ is the 90th percentile of the nonzero gradient magnitude. We select the middle section as the anchor and align the remaining sections outward. For each candidate rotation α, translation is estimated by phase correlation,

$$\boldsymbol{\delta}^*(\alpha) = \arg\max_{\boldsymbol{\delta} \in \Omega} \mathcal{F}^{-1}\left[\frac{\mathcal{F}(E_{\mathrm{ref}})\overline{\mathcal{F}(R_\alpha E_{\mathrm{mov}})}}{\left|\mathcal{F}(E_{\mathrm{ref}})\overline{\mathcal{F}(R_\alpha E_{\mathrm{mov}})}\right| + \epsilon} \right](\boldsymbol{\delta}), \tag{3}$$

and scored by cosine similarity over the valid overlap,

$$s(\alpha) = \frac{\langle E_{\mathrm{ref}}, T_{\boldsymbol{\delta}^*(\alpha)} R_\alpha E_{\mathrm{mov}} \rangle_{\Omega_v}}{\|E_{\mathrm{ref}}\|_{\Omega_v} \|T_{\boldsymbol{\delta}^*(\alpha)} R_\alpha E_{\mathrm{mov}}\|_{\Omega_v}}. \tag{4}$$

A coarse angular search is followed by a fine search around the best angle. The search is bounded by $\pm 30^\circ$, uses 2° and 0.25° steps, limits translation to 25% of the longer image side, and accepts a transform only when $s(\alpha) \geq 0.15$. Each accepted transform is estimated against the already aligned neighboring section. This initialization is disabled in the standard experiments.

S5.2 LEA TRAINING AND STACK INFERENCE

For a triplet (I_{z-1}, I_z, I_{z+1}), the deformation estimator consumes the preceding and center sections,

$$\mathbf{u}_z^{\mathrm{LEA}} = f_\theta(I_{z-1}, I_z), \qquad \widetilde{I}_z = \mathcal{W}(I_z, \mathbf{u}_z^{\mathrm{LEA}}), \tag{5}$$

whereas the following section contributes an additional similarity constraint. The iterative flow-composition rule is given in Equation 3.

For two image patches A and B, Pearson correlation is

$$\rho(A, B) = \frac{\mathbb{E}[(A - \mu_A)(B - \mu_B)]}{\sqrt{(\mathrm{Var}(A) + \epsilon)(\mathrm{Var}(B) + \epsilon)}}. \tag{6}$$

The triplet similarity and flow regularization terms are

$$\mathcal{L}_{\mathrm{tri}} = [1 - \rho(I_{z-1}, \widetilde{I}_z)] + [1 - \rho(I_{z+1}, \widetilde{I}_z)], \tag{7}$$

$$\mathcal{L}_{\mathrm{LEA-reg}} = 10 \sum_k \frac{1}{2} \left(\|\nabla_x \Delta\mathbf{u}^{(k)}\|_2^2 + \|\nabla_y \Delta\mathbf{u}^{(k)}\|_2^2 \right), \tag{8}$$

giving $\mathcal{L}_{\mathrm{LEA}} = \mathcal{L}_{\mathrm{tri}} + \lambda_{\mathrm{LEA}} \mathcal{L}_{\mathrm{LEA-reg}}$.

During inference, LEA evaluates a neighborhood $\mathcal{N}(z)$ around each center section and combines neighbor-induced flows with the direct flow:

$$\overline{\mathbf{u}}_z^{\mathrm{LEA}} = \frac{1}{2} \left[f_\theta(\widetilde{I}_{z-1}, I_z) + \sum_{j \in \mathcal{N}(z)} \bar{w}_j f_\theta(\widetilde{I}_j, I_z) \right]. \tag{9}$$

Provisionally warped neighbors $\widetilde{I}_j$ are updated as the window moves through the stack. Two passes with alternating forward and reverse order reduce dependence on the starting end of the sequence.

S5.3 STACK-LEVEL SELF-SUPERVISED OBJECTIVE

For stage s, let $\mathcal{P}^{(s-1)}$ be the input patch and $\widehat{\mathcal{P}}^{(s)} = \mathcal{W}(\mathcal{P}^{(s-1)}, \Delta\mathbf{u}^{(s)})$ the refined output. The similarity term averages multi-scale local normalized cross-correlation over adjacent refined sections:

$$\mathcal{L}_{\mathrm{sim}}^{(s)} = -\frac{1}{D_p - 1} \sum_{z=1}^{D_p - 1} \mathrm{MSNCC}(\widehat{I}_{z+1}^{(s)}, \widehat{I}_z^{(s)}). \tag{10}$$

Image consistency and residual magnitude are constrained by

$$\mathcal{L}_{\mathrm{id}}^{(s)} = \|\widehat{\mathcal{P}}^{(s)} - \mathcal{P}^{(s-1)}\|_2^2, \qquad \mathcal{L}_{\mathrm{mag}}^{(s)} = \|\Delta\mathbf{u}^{(s)}\|_2^2. \tag{11}$$

Spatial regularity is imposed through

$$\mathcal{L}_{\mathrm{grad}}^{(s)} = \frac{1}{3} \left(\|\nabla_x \Delta\mathbf{u}^{(s)}\|_2^2 + \|\nabla_y \Delta\mathbf{u}^{(s)}\|_2^2 + \|\nabla_z \Delta\mathbf{u}^{(s)}\|_2^2 \right), \tag{12}$$

and boundary and zero-mean constraints use

$$\mathcal{L}_{\mathrm{anchor}}^{(s)} = \|\Delta\mathbf{u}_1^{(s)}\|_2^2 + \|\Delta\mathbf{u}_{D_p}^{(s)}\|_2^2 + \left\| \frac{1}{D_p H_p W_p} \sum_{z,\mathbf{p}} \Delta\mathbf{u}_z^{(s)}(\mathbf{p}) \right\|_2^2. \tag{13}$$

The first- and second-order axial penalties are defined in Equation 11.

The complete objective is the weighted sum reported in Equation 10; its weights are listed in Appendix S1.

S5.4 Robust axial refinement

Let $\widetilde{\mathcal{V}}^{(2)}$ be the Stage 2 volume after Gaussian blending of overlapping patch predictions. We first smooth the Stage 2 image residual relative to $\mathcal{V}^{(1)}$:

$$\mathcal{V}_G^{(2)} = \mathcal{V}^{(1)} + G_{\sigma_z} *_z \left(\widetilde{\mathcal{V}}^{(2)} - \mathcal{V}^{(1)}\right), \qquad \sigma_z = 0.35. \tag{14}$$

For a volume $\mathcal{X}$, let $x_{(1)} \le \cdots \le x_{(5)}$ be the sorted values in the radius-2 axial neighborhood at each spatial location. The trimmed-mean operator is

$$\mathcal{T}(\mathcal{X})_z(\mathbf{p}) = \frac{1}{3}\sum_{j=2}^{4} x_{(j)}, \tag{15}$$

and two successive passes produce

$$\widehat{\mathcal{V}} = \mathcal{T}\Big(\mathcal{T}(\mathcal{V}_G^{(2)})\Big). \tag{16}$$

Reflection padding is used at the stack boundaries. Both operations are deterministic inference-time refinements rather than learned modules.

S6 Scope and limitations

ReStack-Mamba is designed to correct in-plane misalignment in serial-section stacks. Missing sections, folds, tears, severe contamination, and other defects that remove image content are outside the scope of the current deformation model. The deterministic axial refinement improves continuity but can smooth fine texture, as is visible in the real-data comparison. Moreover, the Kasthuri11 experiment has no undeformed reference, so it supports conclusions about visible axial coherence but not absolute registration accuracy. Finally, vEMRec performs best at the mildest simulated deformation level ($\alpha = 1$); the advantage of ReStack-Mamba becomes consistent as deformation severity increases from $\alpha = 2$ to $\alpha = 5$. These observations delimit the current claims and motivate future work on artifact-aware confidence estimation and adaptive refinement strength.

S7 Detailed network architectures

Figure S5 expands the Local Elastic Aligner described in Section 3.2. The pairwise estimator reuses fixed-image features and context across refinement rounds, encodes the progressively warped moving section at multiple scales, and composes each incremental field with the accumulated displacement. The neighboring section contributes only to the triplet similarity loss during training; at inference, Gaussian weights aggregate neighbor-induced displacement fields rather than filtering image intensities.

Figure S6 details the tri-plane feature-modeling core shared by the two StackMamba stages. The encoder preserves the full sectioning dimension while downsampling only the in-plane dimensions. The XY branch retains within-section context, whereas the XZ and YZ branches explicitly traverse the section sequence. Stage 1 and Stage 2 use the same architecture with independent parameters; decoding and bounded residual-field prediction follow the fused representation as described in Section 3.3.

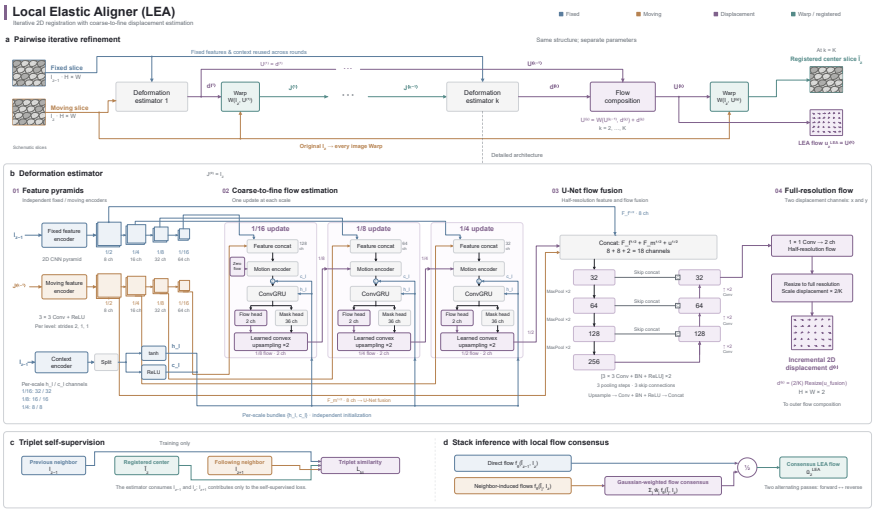

Figure S5: Detailed architecture of the Local Elastic Aligner. LEA iteratively composes coarse-to-fine two-dimensional displacement updates, uses triplet-based self-supervision during training, and combines neighbor-induced fields by Gaussian-weighted flow consensus during stack inference.

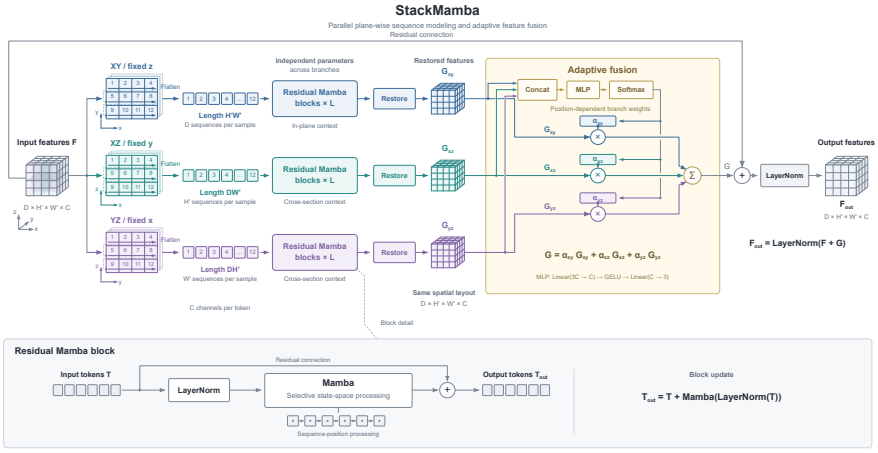

Figure S6: Tri-plane feature modeling in StackMamba. The encoded feature volume is serialized along the XY, XZ, and YZ planes, processed by independent residual Mamba branches, restored to a common spatial layout, and combined using position-dependent softmax weights. The fused features are added to the encoder residual before layer normalization.