DIFFEOMORPHIC TRANSPORT AND PRIOR-GUIDED GENERATION FOR ISOTROPIC VOLUME EM RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Volume electron microscopy (vEM) provides nanometer-scale imaging of cellular ultrastructure, enabling the study of neural circuits and subcellular morphology at unprecedented details. However, because the axial sampling interval is typically much larger than the in-plane resolution, the acquired volumes are highly anisotropic, which limits downstream 3D reconstruction, segmentation, and morphological analysis. Isotropic vEM reconstruction is therefore of central importance, as it recovers anatomically plausible intermediate sections between observed slices. Here we propose a new framework for isotropic reconstruction. Unlike existing axial super-resolution paradigms, we formulate the problem as a structured combination of geometry-preserving transport and detail-restoring generation. This perspective leads to a two-stage design, in which a continuous-time transport model ensures geometric fidelity of structures, while a compact-prior diffusion process recovers high-frequency details. In addition, our method supports arbitrary continuous interpolation between observed sections. Extensive experiments on four datasets from different cell types show that our approach better preserves ultrastructural texture and axial coherence, while achieving superior accuracy and inference efficiency compared with existing methods.

1 INTRODUCTION

Volume electron microscopy (vEM) Peddie et al. (2022) enables nanoscale analysis of cellular ultrastructure, including synaptic organization Dorkenwald et al. (2023) and subcellular morphology Dorkenwald et al. (2022); mic (2025). However, section thickness in most vEM systems remains on the order of tens of nanometers, far larger than the lateral voxel size Briggman & Bock (2012), leading to strongly anisotropic volumes. This anisotropy hinders downstream segmentation Sheridan et al. (2023), tracing Shapson-Coe et al. (2024), and morphological analysis Schubert et al. (2022), making isotropic reconstruction a key problem for biological research.

Current isotropic vEM reconstruction is mainly based on the side-view super-resolution Gao & Gao (2026); Chen et al. (2025); He et al. (2023), which reconstructs the XZ and YZ views and fuses them into a 3D volume (Figure 1). However, this paradigm has two key limitations. First, most methods rely on 2D networks to process different planes independently, which limits their ability to model intrinsic axial dependencies. Second, from the perspective of problem formulation, anisotropy in vEM fundamentally arises from missing continuous states along the axial dimension, *i.e.*, an unobserved intermediate structural evolution process between adjacent sections. However, this paradigm still formulates the task primarily as cross-view resolution enhancement rather than directly modeling intermediate morphological evolution. As a result, most methods only support super-resolution at fixed scaling factors and struggle to capture the continuous appearance changes of biological structures. More recently, diffusion-based methods Lu et al. (2024); Chen et al. (2025) have improved reconstruction quality, but their slow inference remains a major limitation, and most still follow the axial super-resolution formulation. Recent works such as vEMINR Yang et al. (2026) and EMGauss He et al. (2025) have explored alternative representations and problem formulations to move beyond conventional super-resolution and enable continuous-scale isotropic reconstruction.

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.

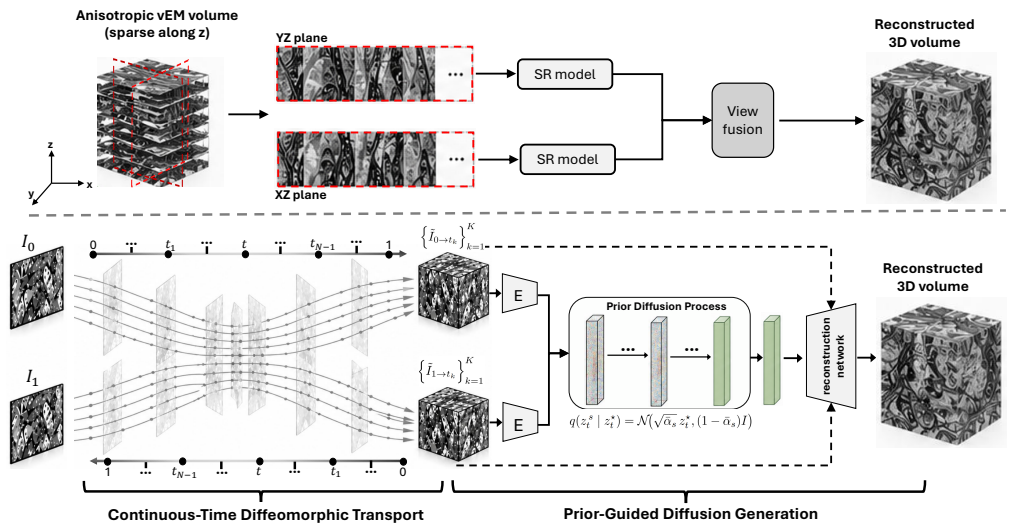


Figure 1: Comparison between the side-view super-resolution paradigm and our framework.

In this work, we argue that isotropic reconstruction should be viewed as a structured continuous-time recovery problem, rather than treated as direct image super-resolution or synthesis alone. Our key observation is that the missing intermediate states consist of two distinct components. One component describes the continuous axial evolution of structural contours and is responsible for preserving geometric shape and topological consistency. The other corresponds to high-frequency ultra-structure, which cannot be sufficiently recovered through stable transport alone.

From this perspective, we propose IsoCTG, a new isotropic reconstruction framework that combines continuous-time transport with prior-guided generation, as shown in Figure 1. Specifically, in Phase I we model axial evolution using a continuous-time diffeomorphic transport process at arbitrary queried positions. This formulation naturally supports continuous reconstruction at arbitrary scales rather than being restricted to a fixed interpolation factor. In Phase II, we further recover high-frequency details with a compact-prior diffusion model. Instead of performing diffusion directly in image space, we conduct diffusion in a compact prior space, which substantially reduces the number of denoising steps and thus improves generation efficiency. The recovered prior is then injected into the reconstruction network to predict the final slice. For reconstruction network, we propose a Query-Time-Modulated State Space Router (QTM-SSR), which couples latent feature propagation with continuous query coordinates instead of assuming fixed temporal steps. This design makes the model aware of queried axial positions and naturally supports arbitrary continuous interpolation. We further introduce Deformable Temporal Shift (DTS) modules in shallow decoding to improve local cross-slice consistency. Together, these components effectively balance geometric fidelity and detail restoration. Extensive experiments show that our method outperforms existing approaches while maintaining favorable inference efficiency with only a few denoising steps.

Our contributions are summarized as follows:

- We propose a geometry-first two-stage framework that combines continuous-time symmetric diffeomorphic transport with compact-prior diffusion, enabling reconstruction at arbitrary continuous positions between observed sections.
- We introduce a prior-guided continuous-time reconstruction network with a Query-Time-Modulated State Space Router and Deformable Temporal Shift to improve axial coherence, local structural alignment, and slice-level realism.
- Extensive experiments on four datasets spanning different cell types demonstrate that our method consistently outperforms existing approaches. In particular, it better preserves ultrastructural texture while achieving favorable inference efficiency by few denoising steps.

2 BACKGROUND

2.1 DIFFUSION MODELS

Diffusion models Ho et al. (2020); Song et al. (2020) learn to recover clean signals from progressively corrupted observations. Given a clean target z^*, the forward noising process and reverse denoising update can be written compactly as

$$q(z^s \mid z^*) = \mathcal{N}\big(z^s; \sqrt{\bar{\alpha}^s}\, z^*, (1-\bar{\alpha}^s)I\big), \qquad z^{s-1} = \frac{1}{\sqrt{\alpha^s}}\left(z^s - \frac{1-\alpha^s}{\sqrt{1-\bar{\alpha}^s}}\,\epsilon_\theta(z^s, s, c)\right), \quad (1)$$

where $\bar{\alpha}^s = \prod_{i=1}^{s} \alpha^i$ and ϵ_θ predicts the noise conditioned on the diffusion step, and condition c.

2.2 STATE SPACE MODELS

State space models (SSMs) Liu et al. (2024); Gu & Dao (2023) describe sequence dynamics through a latent state that evolves over time. In continuous time, a linear controlled SSM is written as

$$\frac{dh(\tau)}{d\tau} = A\,h(\tau) + B(\tau)\,u(\tau), \qquad y(\tau) = C(\tau)^\top h(\tau), \quad (2)$$

where $u(\tau)$ denotes the input signal, $h(\tau)$ the latent state, and $y(\tau)$ the output. The matrix A controls the intrinsic state dynamics, while $B(\tau)$ and $C(\tau)$ determine how input information is injected into the state and how the state is read out. More background is provided in Appendix C.

3 METHOD

3.1 FRAMEWORK OVERVIEW

Let $I_0, I_1 \in \mathbb{R}^{C\times H\times W}$ denote two endpoint slices from a serial electron microscopy (EM) stack. Given a query time set $\mathcal{T} = \{t_k\}_{k=1}^K$ with $0 < t_1 < \cdots < t_K < 1$, our goal is to predict intermediate slices $\hat{\mathcal{I}}_\mathcal{T} = \{\hat{I}_{t_k}\}_{k=1}^K$, where each $\hat{I}_{t_k} \in \mathbb{R}^{C\times H\times W}$, such that the reconstructed sequence forms a geometrically consistent isotropic 3D volume. Since $\mathcal{T}$ may contain an arbitrary number of real-valued positions in $(0,1)$, the model must operate in continuous time rather than on a fixed discrete interpolation grid.

We formulate this task as a transport-with-source problem Holm et al. (2009). Specifically, the unknown continuous slice field $\{I_t\}_{t\in[0,1]}$ is modeled by

$$\partial_t I_t(x) = -\,v(x)\cdot\nabla I_t(x) + s_t(x), \qquad x\in\Omega,\ t\in(0,1), \quad (3)$$

where $v:\Omega\to\mathbb{R}^2$ is a stationary in-plane transport field and s_t is a time-dependent source term. Here, the transport term captures smoothly evolving geometric propagation across depth, while s_t accounts for appearance changes that cannot be explained by transport alone, such as membrane sharpening and ultrastructural detail recovery.

Based on this view, our method follows a geometry-first, generation-second design. As shown in Figure 2, the overall pipeline is

$$(I_0, I_1, \mathcal{T}) \xrightarrow{\mathcal{R}} \left\{\big(\tilde{I}_{0\to t_k}, \tilde{I}_{1\to t_k}\big)\right\}_{k=1}^K \xrightarrow{\mathcal{G}} \{\hat{I}_{t_k}\}_{k=1}^K, \quad (4)$$

where $\mathcal{R}$ denotes the continuous-time transport module of Phase I and $\mathcal{G}$ denotes the prior-guided refinement module of Phase II.

In Phase I, we estimate a single stationary velocity field $v = \mathcal{R}(I_0, I_1)$ and use it to generate two transported drafts, $\tilde{I}_{0\to t}$ and $\tilde{I}_{1\to t}$, for any query time $t\in(0,1)$. These drafts provide geometry-consistent priors but cannot fully recover sharp boundaries or high-frequency ultrastructural textures. Phase II therefore performs compact-prior refinement. It first estimates an intermediate prior representation $\hat{z}_t = \mathcal{D}(\tilde{I}_{0\to t}, \tilde{I}_{1\to t})$ using a diffusion model $\mathcal{D}$, and then predicts a coarse reconstruction

$$P_t = \mathcal{G}_{\text{main}}\big(\tilde{I}_{0\to t}, \tilde{I}_{1\to t}, \hat{z}_t, t\big). \quad (5)$$

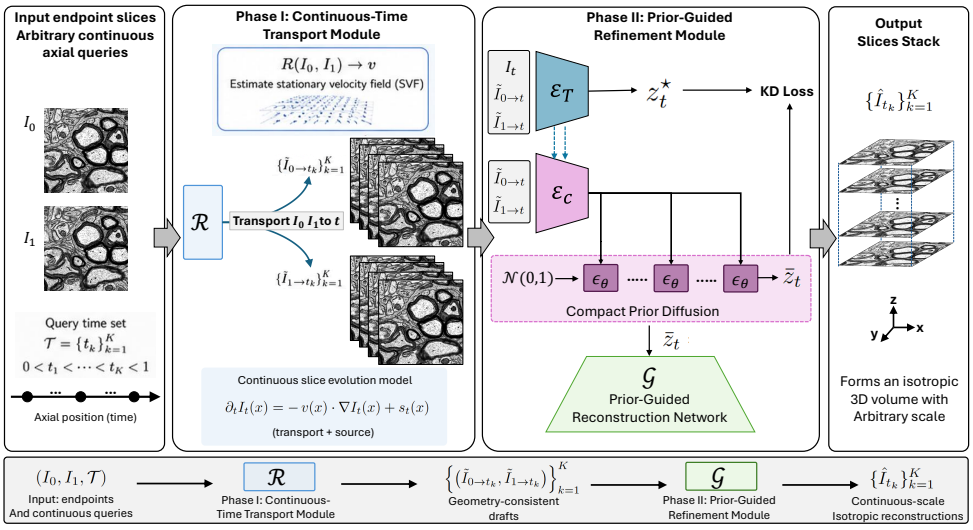


Figure 2: Overall pipeline of the proposed continuous-scale isotropic vEM reconstruction framework. Phase I performs continuous-time transport to generate geometry-consistent drafts from the two endpoint slices at arbitrary query positions, while Phase II uses compact-prior diffusion and texture refinement to restore high-frequency details and produce the final reconstructed slices.

A lightweight texture branch further predicts a residual correction $\Delta_t^{\text{tex}} = \mathcal{G}_{\text{tex}}(P_t, \tilde{I}_{0\to t}, \tilde{I}_{1\to t}, t)$, and the final output is given by

$$\hat{I}_t = P_t + \lambda\, \Delta_t^{\text{tex}}, \tag{6}$$

where $\lambda \in (0, 1)$ is a learnable residual scaling factor.

3.2 Phase I: Continuous-Time Symmetric Diffeomorphic Transport

Following Section 3.1, Phase I captures the transport-consistent component of anisotropic reconstruction by modeling the evolution between the two endpoint slices as a single continuous-time diffeomorphic trajectory. We instantiate this trajectory with a stationary velocity field (SVF)-based diffeomorphic parameterization Ashburner (2007); Krebs et al. (2019); Jena et al. (2024). Concretely, instead of predicting separate deformations for different query times, the transport module $\mathcal{R}$ predicts a single latent motion field $v = \mathcal{R}(I_0, I_1)$, where $v : \Omega \to \mathbb{R}^2$ and $\Omega \subset \mathbb{R}^2$ is the image domain. This velocity field defines how each point moves instantaneously over time, and its integration gives a family of diffeomorphic transformations ϕ_τ that describe the accumulated motion from the reference coordinates to time τ. The corresponding displacement field u_τ can then be understood as the offset induced by ϕ_τ relative to the identity mapping. In this way, v represents the underlying motion dynamics, ϕ_τ represents the resulting transformation trajectory, and u_τ is the explicit spatial displacement used for warping. For a query time $t \in (0, 1)$, reconstruction is performed by backward sampling from the two endpoints using

$$u_{t\to 0} = \text{Exp}(-tv),\ \ u_{t\to 1} = \text{Exp}((1-t)v),\ \ \tilde{I}_{0\to t} = \mathcal{W}(I_0, u_{t\to 0}),\ \ \tilde{I}_{1\to t} = \mathcal{W}(I_1, u_{t\to 1}), \tag{7}$$

where $\mathcal{W}(I, u)(x) = I(x + u(x))$ denotes differentiable backward warping. Since all query times are generated from the same field, the model naturally supports arbitrary-rate isotropic reconstruction. Full architectural details are provided in Appendix D.1.

3.3 Phase II: Prior-Guided Continuous-Time Generation

Given the two drafts $\tilde{I}_{0\to t}$ and $\tilde{I}_{1\to t}$, Phase II activates the residual detail component of anisotropic reconstruction. Once the transport-consistent structure has been obtained by Phase I, the remaining uncertainty is mainly concentrated in high-frequency details. We therefore avoid image-space diffusion Ho et al. (2020); Song et al. (2020) and instead perform diffusion in a compact prior space Xia et al. (2024); Chen et al. (2023), whose output guides the reconstruction network. Concretely, the

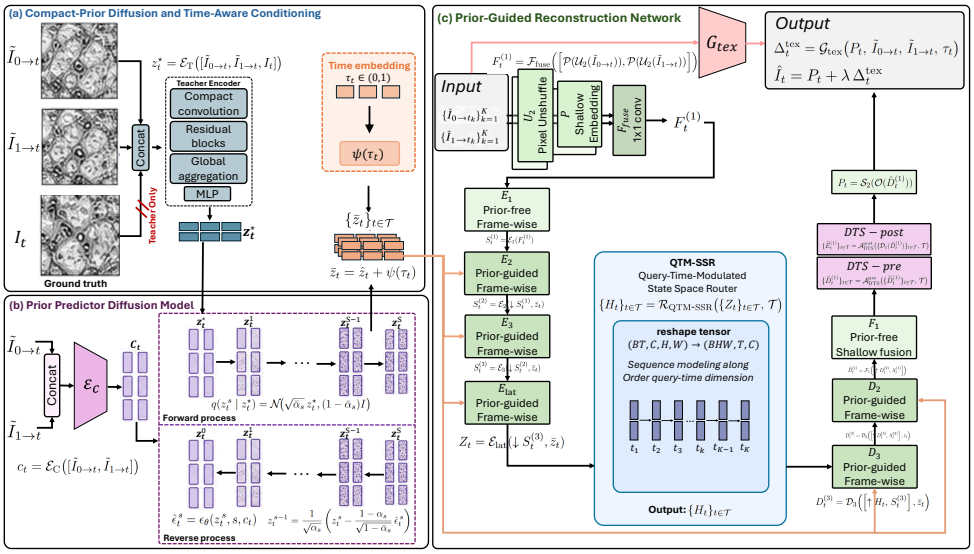


Figure 3: Overview of the Prior-Guided Continuous-Time Generation module. It can be divided into three parts: Compact-Prior Diffusion and Time-Aware Conditioning, Prior Predictor Diffusion, and Prior-Guided Reconstruction Network.

generative module operates in two steps (Figure 3). First, using the ground-truth intermediate slice I_t, a teacher prior extractor learns a compact prior $z_t^\star$ from $\tilde{I}_{0\to t}$ and $\tilde{I}_{1\to t}$. Second, a lightweight diffusion model predicts an approximation $\hat{z}_t$ of this prior from the two drafts. A continuous time embedding is then added to form the time-aware prior $\bar{z}_t = \hat{z}_t + \psi(t)$, which is fed into the prior-guided reconstruction network to produce the final slice $\hat{I}_t$.

3.3.1 Compact-Prior Diffusion and Time-Aware Conditioning

We first construct a teacher prior space using the ground-truth intermediate section I_t, as shown in Figure 3(a). A teacher encoder network $\mathcal{E}_{\mathrm{T}}$ maps the two transported drafts and the target slice to a compact prior

$$z_t^\star = \mathcal{E}_{\mathrm{T}}\big([\tilde{I}_{0\to t}, \tilde{I}_{1\to t}, I_t]\big), \tag{8}$$

where $[\cdot]$ denotes channel-wise concatenation. This teacher prior provides the supervision target for the second-stage prior estimator.

At inference time, the target slice is unavailable, so we learn a compact-prior diffusion model that predicts the same prior $\hat{z}_t$ from the drafts, as shown in Figure 3(b). A condition encoder extracts a compact descriptor

$$c_t = \mathcal{E}_{\mathrm{C}}\big([\tilde{I}_{0\to t}, \tilde{I}_{1\to t}]\big), \tag{9}$$

and diffusion is performed in prior space rather than image space. Starting from the teacher prior $z_t^\star$, the forward process is defined as

$$q(z_t^s \mid z_t^*) = \mathcal{N}\big(z_t^s; \sqrt{\bar{\alpha}^s}\, z_t^*, (1-\bar{\alpha}^s)I\big), \tag{10}$$

where t denotes the slice index and $s \in \{1, \dots, S\}$ denotes the diffusion step. Since the drafts already provide substantial structural cues, diffusion in this compact prior space requires far fewer denoising steps than image-space diffusion. In addition, to make the prior aware of the axial position, we inject a continuous coordinate via $\bar{z}_t = \hat{z}_t + \psi(\tau_t)$, where $\tau_t \in (0, 1)$ denotes the normalized axial coordinate and $\psi(\cdot)$ is a time embedding.

3.3.2 Prior-Guided Reconstruction Network

Given an ordered query time set $\mathcal{T} = \{t_k\}_{k=1}^K$, the reconstruction network takes the two draft sequences $\{I_{0\to t}, I_{1\to t}\}_{t\in\mathcal{T}}$ and $\{\bar{z}_t\}_{t\in\mathcal{T}}$ as input, and predicts the final slices in a prior-guided

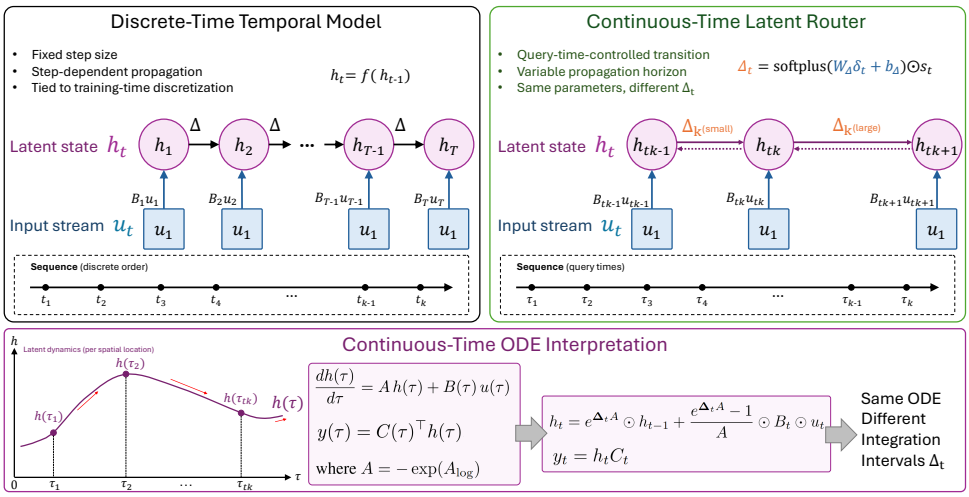


Figure 4: Our Query-Time-Modulated State Space Router models latent evolution as a continuous-time dynamical system, where query-dependent step sizes Δ_t define variable integration intervals of a shared latent ODE.

coarse-to-fine manner, as shown in Figure 3(c). In implementation, the ordered query stack is reshaped from (B, T, C, H, W) to (BT, C, H, W) for frame-wise spatial processing, while temporal interaction across the ordered query dimension is introduced explicitly in dedicated temporal modules.

We first map the two drafts to a shared shallow representation

$$F_t^{(1)} = \mathcal{F}_{\text{fuse}}\Big(\Big[\mathcal{P}(\mathcal{U}_2(\tilde{I}_{0\to t})), \mathcal{P}(\mathcal{U}_2(\tilde{I}_{1\to t}))\Big]\Big), \tag{11}$$

where $\mathcal{U}_2(\cdot)$ denotes PixelUnshuffle, $\mathcal{P}(\cdot)$ the shallow embedding, and $\mathcal{F}_{\text{fuse}}(\cdot)$ a fusion projection. The resulting features are processed by a shared encoder that builds a three-level skip hierarchy and a latent feature:

$$S_t^{(1)} = \mathcal{E}_1(F_t^{(1)}) \quad S_t^{(2)} = \mathcal{E}_2(\downarrow S_t^{(1)}, \bar{z}_t) \quad S_t^{(3)} = \mathcal{E}_3(\downarrow S_t^{(2)}, \bar{z}_t) \quad Z_t = \mathcal{E}_{\text{lat}}(\downarrow S_t^{(3)}, \bar{z}_t). \tag{12}$$

To enforce coherent evolution across ordered axial positions, we introduce a Query-Time-Modulated State Space Router (QTM-SSR):

$$\{H_t\}_{t\in\mathcal{T}} = \mathcal{R}_{\text{QTM-SSR}}\big(\{Z_t\}_{t\in\mathcal{T}}, \mathcal{T}\big). \tag{13}$$

The routed latent features are then progressively decoded with skip fusion:

$$D_t^{(3)} = \mathcal{D}_3\Big(\Big[\uparrow H_t,\, S_t^{(3)}\Big], \bar{z}_t\Big), \quad D_t^{(2)} = \mathcal{D}_2\Big(\Big[\uparrow D_t^{(3)},\, S_t^{(2)}\Big], \bar{z}_t\Big), \quad \widetilde{D}_t^{(1)} = \mathcal{F}_1\Big(\Big[\uparrow D_t^{(2)},\, S_t^{(1)}\Big]\Big), \tag{14}$$

where $\mathcal{D}_3$ and $\mathcal{D}_2$ denote prior-guided decoder blocks, and $\mathcal{F}_1$ denotes the shallow fusion projection.

To compensate for residual local misalignment across nearby query positions, we introduce lightweight deformable temporal shift (DTS) modules before and after the shallow decoder stage:

$$\{\bar{D}_t^{(1)}\}_{t\in\mathcal{T}} = \mathcal{A}_{\text{DTS}}^{\text{pre}}\big(\{\widetilde{D}_t^{(1)}\}_{t\in\mathcal{T}}, \mathcal{T}\big), \qquad \{\widetilde{E}_t^{(1)}\}_{t\in\mathcal{T}} = \mathcal{A}_{\text{DTS}}^{\text{post}}\big(\{\mathcal{D}_1(\bar{D}_t^{(1)})\}_{t\in\mathcal{T}}, \mathcal{T}\big), \tag{15}$$

followed by a lightweight refinement stack to obtain the final feature representation $\hat{D}_t^{(1)}$. Each DTS module reshapes the shallow feature back to (B, T, C, H, W), aligns a subset of channels from the previous query position to the current one, and fuses them through a learned gate.

Finally, the refined feature is projected to a slice prediction $P_t = \mathcal{S}_2(\mathcal{O}(\hat{D}_t^{(1)}))$, where $\mathcal{O}(\cdot)$ is the output projection and $\mathcal{S}_2(\cdot)$ denotes PixelShuffle with scale factor 2. A lightweight full-resolution texture branch then predicts a residual correction $\Delta_t^{\text{tex}} = \mathcal{G}_{\text{tex}}\big(P_t, \tilde{I}_{0\to t}, \tilde{I}_{1\to t}, \tau_t\big)$, where τ_t denotes the normalized query coordinate used by the texture branch. The final reconstruction is obtained by Eq. equation 6. Full architectural details are provided in Appendix E.1.

3.3.3 QUERY-TIME-MODULATED STATE SPACE ROUTER

A key requirement of isotropic reconstruction is that latent feature evolution should be aware of the queried axial positions, rather than tied to a fixed discrete frame rate. A purely discrete temporal module would only model propagation over a predetermined sampling grid, making it difficult to generalize to arbitrary continuous query times. To address this, we introduce a Query-Time-Modulated State Space Router (QTM-SSR) based on a bidirectional state-space model (SSM) Gu & Dao (2023); Liu et al. (2024). As illustrated in Figure 4, our design replaces fixed-step temporal propagation with a query-time-controlled transition mechanism.

Let $\{Z_t\}_{t\in\mathcal{T}}$ denote the ordered latent feature sequence. The latent tensor is rearranged from (BT, C, H, W) to (BHW, T, C), so that the state-space scan is performed along the ordered query-time dimension at each spatial location. For each spatial location, the router transforms the latent sequence into an input stream u_t, a gating signal g_t, and dynamic parameters (δ_t, B_t, C_t). Here, δ_t is an input-dependent scalar step descriptor, while B_t and C_t are content-adaptive input and readout parameters of the state-space model.

Our key design is to make the effective SSM Gu & Dao (2023) step size explicitly query-time aware:

$$\mathbf{\Delta}_t = \mathrm{softplus}(W_\Delta \delta_t + b_\Delta) \odot s_t, \tag{16}$$

where W_Δ and b_Δ are learnable projection parameters that map δ_t to a channel-wise positive discretization step, $\mathrm{softplus}(\cdot)$ ensures numerical positivity, s_t is the time scale associated with query time t, and $\odot$ denotes element-wise multiplication. As visualized in Figure 4, varying $\mathbf{\Delta}_t$ enables different propagation ranges under the same learned dynamics, allowing the router to operate at arbitrary continuous query times without retraining.

Given a stable transition matrix A from Eq. 2, we use exact zero-order-hold discretization for selective scan:

$$h_t^{\rightarrow} = e^{\mathbf{\Delta}_t A} \odot h_{t-1}^{\rightarrow} + \frac{e^{\mathbf{\Delta}_t A} - 1}{A} \odot B_t \odot u_t, \qquad y_t^{\rightarrow} = \langle h_t^{\rightarrow}, C_t \rangle, \tag{17}$$

where $h_t^{\rightarrow}$ is the forward hidden state, $y_t^{\rightarrow}$ is the corresponding readout, and $\langle \cdot, \cdot \rangle$ denotes the inner product along the state dimension. The backward direction is defined analogously to obtain $y_t^{\leftarrow}$.

The two directional outputs are then fused as

$$r_t = W_o\big(\big(y_t^{\rightarrow} + y_t^{\leftarrow}\big) \odot \phi(g_t)\big), \tag{18}$$

where $\phi(\cdot)$ denotes the gating nonlinearity and W_o is the output projection. Full architectural details are provided in Appendix E.1.2.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets and baselines. We evaluate on four EM datasets Heinrich et al. (2021): mus-liver, mus-cortex-3, fly-protocerebral-bridge, and fly-ellipsoid-body, covering diverse mammalian and fly-brain ultrastructures. For each dataset, we use $2048 \times 2048 \times 2048$ voxel sub-volumes for training and evaluation. We compare with representative anisotropic EM reconstruction methods, including VE-Mamba Gao & Gao (2026), vEMINR Yang et al. (2026), and EMDiffuse Lu et al. (2024). All baselines are retrained on our datasets following their original settings.

4.2 EXPERIMENTAL RESULTS

The quantitative results in Table 1 show that IsoCTG delivers consistently strong performance across all four datasets under both $4\times$ and $8\times$ settings. It achieves the best PSNR/SSIM on *mus-liver* at both $4\times$ (23.10/0.79) and $8\times$ (20.96/0.69), outperforming the strongest baseline vEMINR. Similar gains are observed on *fly-protocerebral-bridge*, while IsoCTG also obtains the best result on *fly-ellipsoid-body* at $8\times$. Although vEMINR is slightly better on *mus-cortex-3* and *fly-ellipsoid-body* at $4\times$, the margins are small. Overall, IsoCTG ranks at or near the top across datasets, with more pronounced advantages under the more challenging $8\times$ setting.

Table 1: Quantitative comparison on four datasets under $4\times$ and $8\times$ axial reconstruction settings. Best and second-best results are marked in **bold** and underline, respectively.

Dataset	Scale	VEMamba Gao & Gao (2026)		EMDiffuse Lu et al. (2024)		vEMINR Yang et al. (2026)		**IsoCTG (Ours)**	
		PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
mus-liver	4×	22.12	**0.79**	21.67	0.71	22.39	0.78	**23.10**	**0.79**
	8×	20.09	0.64	19.11	0.55	20.23	0.67	**20.96**	**0.69**
mus-cortex-3	4×	**21.85**	0.83	20.02	0.77	21.37	**0.84**	21.43	**0.84**
	8×	**19.17**	0.72	16.92	0.59	18.44	**0.73**	18.47	0.72
fly-protocerebral-bridge	4×	18.71	0.76	19.45	0.73	19.66	0.78	**20.08**	**0.80**
	8×	**17.35**	0.65	17.20	0.57	16.91	0.65	17.11	**0.67**
fly-ellipsoid-body	4×	19.94	0.85	18.97	0.80	**20.38**	**0.86**	20.34	0.85
	8×	14.92	0.70	15.61	0.62	17.48	**0.75**	**17.54**	**0.75**

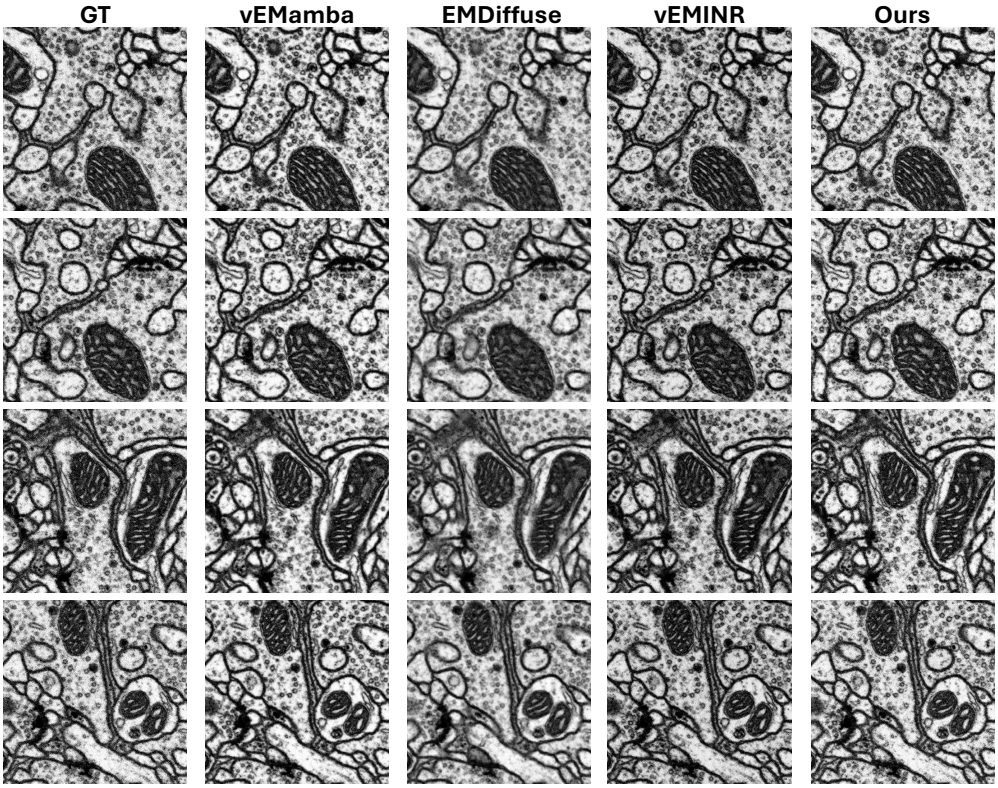
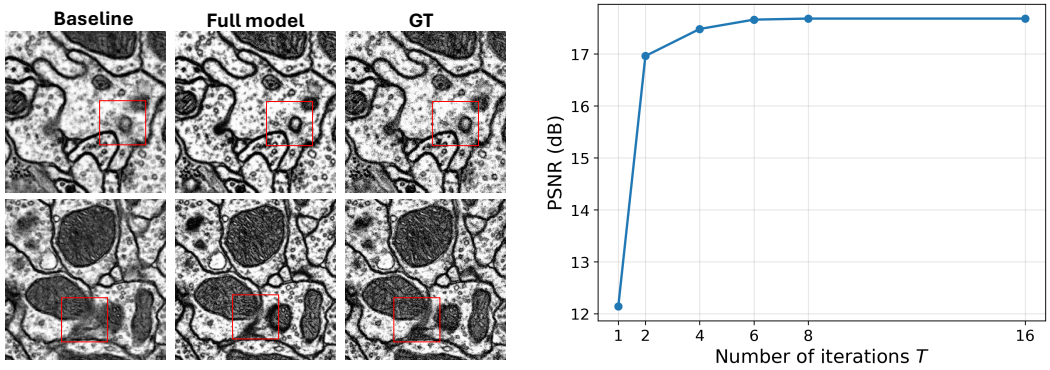


Figure 5: Isotropic xy-slice reconstruction results on anisotropic datasets. More results are provided in Appendix F.2.

The visual comparisons in Fig. 5 further support these results. At $4\times$, competing methods recover the main tissue structures but often blur thin membranes or lose fine details. IsoCTG produces sharper boundaries and cleaner textures closer to the ground truth. At $8\times$, the gap becomes more evident: EMDiffuse tends to over-smooth structures, while vEMINR may produce incomplete local morphology. In contrast, IsoCTG better preserves membrane continuity and structural details, showing stronger robustness for large-ratio axial reconstruction.

4.3 Ablation Study

Component ablation. Table 2(a) evaluates QTM-SSR and DTS. Both modules independently improve the baseline, while their combination achieves the best performance. QTM-SSR provides the larger individual gain, indicating the importance of query-dependent cross-slice modeling, whereas



(a) Qualitative comparison.

(b) Number of diffusion iterations.

Figure 6: Analysis of the compact diffusion prior. **(a)** The full model produces sharper structures than the baseline without prior guidance. **(b)** Reconstruction performance versus the number of diffusion iterations T.

Table 2: Ablation studies of the proposed components and training strategy. Best results are shown in **bold**.

(a) QTM-SSR and DTS

Variant	QTM-SSR	DTS	PSNR ↑	SSIM ↑
QTM-SSR only	✓	✗	17.62	0.78
DTS only	✗	✓	17.53	0.77
Full model	✓	✓	**17.66**	**0.79**

(b) Diffusion prior and training strategy

Variant	Training	PSNR ↑	SSIM ↑
No prior	–	17.34	0.76
Separate training	Separate	17.39	0.77
Joint training	Joint	**17.66**	**0.79**

DTS further improves local alignment around non-rigid structures and boundaries. The two modules are therefore complementary.

Diffusion prior and joint optimization. As shown in Table 2(b), introducing the compact diffusion prior improves PSNR from 17.34 to 17.66 dB and SSIM from 0.76 to 0.79 when jointly optimized with the reconstruction network. Separate training provides only a marginal improvement, suggesting that a frozen prior predictor cannot fully adapt to the reconstruction objective. Joint optimization allows the prior and reconstruction branches to co-adapt and yields the best performance. The compact diffusion branch adds only 6.38M parameters.

Diffusion iterations. We evaluate $T \in \{1, 2, 4, 6, 8, 16\}$ diffusion iterations, with the variance schedule linearly interpolated from $\beta_1 = 0.1$ to $\beta_T = 0.99$. As shown in Fig. 6b, a single iteration is insufficient to produce informative prior features, while performance improves rapidly with additional iterations and largely saturates at $T = 6$. Figure 6a further shows that prior guidance improves the reconstruction of fine structures and textures.

5 Conclusion

In this paper, we presented isoCTG, a continuous framework for isotropic reconstruction in serial electron microscopy that combines geometric transport and compact-prior generation. Instead of treating anisotropic recovery as direct slice synthesis, isoCTG decomposes the problem into a transport-consistent component for continuous geometric propagation and a residual component for high-frequency detail restoration. This leads to a two-stage design: a continuous-time transport module that produces geometry-consistent drafts, followed by a compact-prior diffusion refinement module that recovers missing ultrastructural details at arbitrary query positions. Extensive experiments on four different datasets demonstrate that our proposed method outperforms state-of-the-art methods.

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

Functional connectomics spanning multiple areas of mouse visual cortex. *Nature*, 640(8058):435–447, 2025.

John Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007.

Kevin L Briggman and Davi D Bock. Volume electron microscopy for neuronal circuit reconstruction. *Current opinion in neurobiology*, 22(1):154–161, 2012.

Bohao Chen, Yanchao Zhang, Yanan Lv, Hua Han, and Xi Chen. Self-supervised axial super-resolution for volume microscopy via diffusion-guided structure distillation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 467–477. Springer, 2025.

Zheng Chen, Yulun Zhang, Ding Liu, Jinjin Gu, Linghe Kong, Xin Yuan, et al. Hierarchical integration diffusion model for realistic image deblurring. *Advances in neural information processing systems*, 36:29114–29125, 2023.

Shiyu Deng, Xueyang Fu, Zhiwei Xiong, Chang Chen, Dong Liu, Xuejin Chen, Qing Ling, and Feng Wu. Isotropic reconstruction of 3d em images with unsupervised degradation learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 163–173. Springer, 2020.

Sven Dorkenwald, Claire E McKellar, Thomas Macrina, Nico Kemnitz, Kisuk Lee, Ran Lu, Jingpeng Wu, Sergiy Popovych, Eric Mitchell, Barak Nehoran, et al. Flywire: online community for whole-brain connectomics. *Nature methods*, 19(1):119–128, 2022.

Sven Dorkenwald, Peter H Li, Michał Januszewski, Daniel R Berger, Jeremy Maitin-Shepard, Agnes L Bodor, Forrest Collman, Casey M Schneider-Mizell, Nuno Maçarico da Costa, Jeff W Lichtman, et al. Multi-layered maps of neuropil with segmentation-guided contrastive learning. *Nature Methods*, 20(12):2011–2020, 2023.

Longmi Gao and Pan Gao. Vemamba: Efficient isotropic reconstruction of volume electron microscopy with axial-lateral consistent mamba. *arXiv preprint arXiv:2603.00887*, 2026.

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

Jia He, Yan Zhang, Wenhao Sun, Ge Yang, and Fei Sun. Isovem: isotropic reconstruction for volume electron microscopy based on transformer. *bioRxiv*, pp. 2023–11, 2023.

Yumeng He, Zanwei Zhou, Yekun Zheng, Chen Liang, Yunbo Wang, and Xiaokang Yang. Emgauss: Continuous slice-to-3d reconstruction via dynamic gaussian modeling in volume electron microscopy. *arXiv preprint arXiv:2512.06684*, 2025.

Larissa Heinrich, Davis Bennett, David Ackerman, Woohyun Park, John Bogovic, Nils Eckstein, Alyson Petruncio, Jody Clements, Song Pang, C Shan Xu, et al. Whole-cell organelle segmentation in volume electron microscopy. *Nature*, 599(7883):141–146, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Darryl Holm, Alain Trouvé, and Laurent Younes. The euler-poincaré theory of metamorphosis. *Quarterly of Applied Mathematics*, 67(4):661–685, 2009.

Rohit Jena, Deeksha Sethi, Pratik Chaudhari, and James C Gee. Deep learning in medical image registration: Magic or mirage? *Advances in Neural Information Processing Systems*, 37:108331–108353, 2024.

Julian Krebs, Hervé Delingette, Boris Mailhé, Nicholas Ayache, and Tommaso Mansi. Learning a probabilistic model for diffeomorphic registration. *IEEE transactions on medical imaging*, 38(9): 2165–2176, 2019.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2024.

Chixiang Lu, Kai Chen, Heng Qiu, Xiaojun Chen, Gu Chen, Xiaojuan Qi, and Haibo Jiang. Diffusion-based deep learning method for augmenting ultrastructural imaging and volume electron microscopy. *Nature communications*, 15(1):4677, 2024.

Gayathri Mahalingam, Russel Torres, Daniel Kapner, Eric T Trautman, Tim Fliss, Shamishtaa Seshamani, Eric Perlman, Rob Young, Samuel Kinn, JoAnn Buchanan, et al. A scalable and modular automated pipeline for stitching of large electron microscopy datasets. *Elife*, 11:e76534, 2022.

Andreas Mueller, Deborah Schmidt, Jan Philipp Albrecht, Lucas Rieckert, Maximilian Otto, Leticia Elizabeth Galicia Garcia, Gunar Fabig, Michele Solimena, and Martin Weigert. Modular segmentation, spatial analysis and visualization of volume electron microscopy datasets. *Nature Protocols*, 19(5):1436–1466, 2024.

Christopher J Peddie, Christel Genoud, Anna Kreshuk, Kimberly Meechan, Kristina D Micheva, Kedar Narayan, Constantin Pape, Robert G Parton, Nicole L Schieber, Yannick Schwab, et al. Volume electron microscopy. *Nature Reviews Methods Primers*, 2(1):51, 2022.

Philipp J Schubert, Sven Dorkenwald, Michał Januszewski, Jonathan Klimesch, Fabian Svara, Andrei Mancu, Hashir Ahmad, Michale S Fee, Viren Jain, and Joergen Kornfeld. Syconn2: dense synaptic connectivity inference for volume electron microscopy. *Nature Methods*, 19(11):1367–1370, 2022.

Alexander Shapson-Coe, Michał Januszewski, Daniel R Berger, Art Pope, Yuelong Wu, Tim Blakely, Richard L Schalek, Peter H Li, Shuohong Wang, Jeremy Maitin-Shepard, et al. A petavoxel fragment of human cerebral cortex reconstructed at nanoscale resolution. *Science*, 384 (6696):eadk4858, 2024.

Arlo Sheridan, Tri M Nguyen, Diptodip Deb, Wei-Chung Allen Lee, Stephan Saalfeld, Srinivas C Turaga, Uri Manor, and Jan Funke. Local shape descriptors for neuron segmentation. *Nature methods*, 20(2):295–303, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Zejin Wang, Guodong Sun, Guoqing Li, Lijun Shen, Lina Zhang, and Hua Han. Stdin: Spatio-temporal distilled interpolation for electron microscope images. *Neurocomputing*, 505:188–202, 2022.

Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, Radu Timotfe, and Luc Van Gool. Diffi2i: efficient diffusion model for image-to-image translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1578–1593, 2024.

Jibin Yang, Jie Huo, Muyu Liu, Chenjie Feng, Yan Zhang, Gang Pan, Wenjia Meng, and Renmin Han. veminr: Ultra-fast isotropic reconstruction for volume electron microscopy with implicit neural representation. *Advanced Science*, pp. e11922, 2026.

A Appendix

B Related works

Isotropic vEM reconstruction aims to recover missing axial information from anisotropic volumes for accurate 3D structural analysis. Early methods mainly relied on interpolation Mahalingam et al. (2022); Mueller et al. (2024) or optical-flow-based alignment Wang et al. (2022), which are simple and practical but often produce blur and artifacts under large anisotropy. Recent work has largely shifted to side-view super-resolution Gao & Gao (2026); Chen et al. (2025); He et al. (2023); Deng et al. (2020), which learns axial recovery on the XZ and YZ planes and then fuses the reconstructed views into a 3D volume. Within this paradigm, GAN-based methods such as IsoGAN Deng et al. (2020) improve perceptual sharpness but are often sensitive to training instability and hallucination artifacts. Transformer-based approaches such as IsoVEM He et al. (2023) model cross-slice spatial continuity with video-style attention and achieve stronger structural consistency, but they often incur substantial computational and memory costs. Diffusion-based methods, including vEMDiffuse Lu et al. (2024) and D2R Chen et al. (2025), further enhance reconstruction quality yet they still inherit the limitations of side-view generation and multi-stage supervision. More recently, VEMamba Gao & Gao (2026) introduces a 3D dependency reordering strategy based on Mamba sequence modeling, aiming to better capture axial–lateral correlations together with degradation-aware reconstruction. EMGauss He et al. (2025) reformulates slice-to-3D recovery as a dynamic scene rendering problem with Gaussian splatting, treating axial progression as the evolution of 2D Gaussian point clouds. vEMINR Yang et al. (2026) instead models the volume as a continuous implicit neural representation, enabling continuous-scale reconstruction while learning degradation patterns from anisotropic observations. Despite these advances, most existing methods still formulate isotropic vEM reconstruction primarily as plane-wise super-resolution, which limits their ability to explicitly model intrinsic cross-slice dependencies and the continuous morphological evolution between adjacent sections.

C Supplementary Background

C.1 Diffusion Models

Diffusion models Ho et al. (2020); Song et al. (2020) learn a reverse denoising process that recovers clean signals from progressively corrupted observations. Let z^* denote a clean target variable. In the forward process, Gaussian noise is gradually added over diffusion steps $s = 1, \dots, S$. The marginal distribution at step s is

$$q(z^s \mid z^*) = \mathcal{N}\big(z^s; \sqrt{\bar{\alpha}^s}\, z^*,\, (1-\bar{\alpha}^s)I\big), \tag{19}$$

where $\alpha^s \in (0,1)$ is the noise schedule and $\bar{\alpha}^s = \prod_{i=1}^{s} \alpha^i$. Equivalently, a noisy sample can be written as

$$z^s = \sqrt{\bar{\alpha}^s}\, z^* + \sqrt{1-\bar{\alpha}^s}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{20}$$

As s increases, the signal content is gradually destroyed and the distribution approaches isotropic Gaussian noise.

The learning problem is to invert this corruption process. A neural network ϵ_θ is trained to predict the injected noise from the noisy sample z^s, the diffusion step s, and optional conditional inputs c:

$$\hat{\epsilon}^s = \epsilon_\theta(z^s, s, c). \tag{21}$$

A common training objective is the noise-prediction loss

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{z^*,\,\epsilon,\,s}\left[\|\epsilon - \epsilon_\theta(z^s, s, c)\|_2^2\right]. \tag{22}$$

By learning to estimate the perturbation ϵ, the model can infer how to progressively remove noise and recover a structured latent variable.

At inference time, the reverse process iteratively denoises the sample from large s to small s. In the deterministic form commonly used in restoration-oriented diffusion models, one reverse update can be written as

$$z^{s-1} = \frac{1}{\sqrt{\alpha^s}}\left(z^s - \frac{1-\alpha^s}{\sqrt{1-\bar{\alpha}^s}}\,\hat{\epsilon}^s\right). \tag{23}$$

Repeating this update over multiple steps yields a clean estimate consistent with the condition c.

Compared with image-space diffusion, latent-space or compact-space diffusion is often much more efficient, because the denoising process operates on a lower-dimensional representation instead of directly on high-resolution images. In our framework, diffusion is used in a compact prior space to estimate an intermediate prior representation for slice reconstruction. Since the transport module already provides strong geometric guidance, the diffusion model mainly focuses on recovering missing fine structures and high-frequency details, which allows accurate reconstruction with relatively few denoising steps.

C.2 State Space Models

State space models (SSMs) Liu et al. (2024); Gu & Dao (2023) are sequence modeling frameworks that represent temporal dynamics through a latent state evolving over time. Their central idea is to maintain a hidden state that summarizes past information and propagates it forward according to a dynamical system, while the output at each step is obtained through a readout operation on that state.

In continuous time, a linear controlled SSM is written as

$$\frac{dh(\tau)}{d\tau} = A\,h(\tau) + B(\tau)\,u(\tau), \qquad y(\tau) = C(\tau)^\top h(\tau), \tag{24}$$

where $u(\tau)$ denotes the input signal, $h(\tau)$ the latent state, and $y(\tau)$ the output. The matrix A controls the intrinsic state dynamics, while $B(\tau)$ and $C(\tau)$ determine how input information is injected into the state and how the state is read out.

For sequence modeling, the continuous-time system is discretized over intervals of length Δ_t. Under zero-order-hold discretization, the input is assumed constant within each interval, and the exact discrete update becomes

$$h_t = e^{\Delta_t A} h_{t-1} + \int_0^{\Delta_t} e^{(\Delta_t - \tau)A} B_t u_t \, d\tau, \qquad y_t = C_t^\top h_t. \tag{25}$$

When A is invertible, the integral term has the closed form

$$h_t = e^{\Delta_t A} h_{t-1} + A^{-1}\big(e^{\Delta_t A} - I\big)\, B_t u_t, \qquad y_t = C_t^\top h_t. \tag{26}$$

This update shows that the hidden-state transition depends explicitly on the step size Δ_t, rather than assuming a fixed discrete interval.

This dependence on Δ_t is particularly important when observations are queried at irregular or continuous positions. Unlike standard discrete sequence models that are naturally tied to a fixed index grid, SSMs can adapt their transition behavior according to the actual interval between consecutive states. As a result, they provide a principled mechanism for modeling variable temporal or spatial evolution with shared parameters.

In modern vision models, SSMs are often implemented in a selective or input-conditioned manner, so that the propagation behavior can adapt to local content and long-range dependencies while retaining linear-time state propagation. In our framework, the SSM serves as the latent temporal propagation mechanism inside the Query-Time-Modulated State Space Router. There, the transition is modulated by query-time information, enabling the model to propagate features across ordered target slices while remaining explicitly aware of continuous axial positions.

D Additional Details on Phase I

D.1 Network Architecture

Phase I takes two endpoint sections $I_0, I_1 \in \mathbb{R}^{C\times H\times W}$ as input and predicts a stationary velocity field (SVF) $v \in \mathbb{R}^{2\times H\times W}$. The SVF defines a continuous diffeomorphic trajectory between the two endpoints and can be integrated at arbitrary query times to generate geometry-consistent intermediate drafts. The architecture consists of a shared multiscale encoder, a symmetric midpoint decoder, a diffeomorphic integrator, and a continuous-time warping module.

Shared multiscale encoder. The two endpoint sections are processed by a shared four-level pyramid encoder $\mathcal{E}$, producing feature pyramids $\{f_0^{(\ell)}\}_{\ell=0}^{3}$ and $\{f_1^{(\ell)}\}_{\ell=0}^{3}$ at resolutions H, $H/2$, $H/4$, and $H/8$. For efficiency, the two inputs are concatenated along the batch dimension and encoded jointly before being separated into two branches. A lightweight residual extractor is used at full resolution, while lower-resolution stages employ stride-2 downsampling followed by large-kernel attention blocks. This design preserves fine boundary information while progressively enlarging the receptive field.

Symmetric midpoint decoder. The decoder estimates the SVF in a coarse-to-fine manner over levels $\ell = 3, 2, 1, 0$. At each level, the current velocity estimate $v^{(\ell)}$ is resized to the corresponding spatial resolution while preserving its pixel-unit convention. Instead of directly aligning one endpoint to the other, we symmetrically transport both endpoint features toward their temporal midpoint:

$$\begin{aligned} u_{\mathrm{mid}\to 0}^{(\ell)} &= \mathrm{Exp}\Big(-\tfrac{1}{2}v^{(\ell)}\Big), \quad \bar{f}_0^{(\ell)} = \mathcal{W}\Big(f_0^{(\ell)}, u_{\mathrm{mid}\to 0}^{(\ell)}\Big), \\ u_{\mathrm{mid}\to 1}^{(\ell)} &= \mathrm{Exp}\Big(\tfrac{1}{2}v^{(\ell)}\Big), \quad \bar{f}_1^{(\ell)} = \mathcal{W}\Big(f_1^{(\ell)}, u_{\mathrm{mid}\to 1}^{(\ell)}\Big). \end{aligned} \tag{27}$$

This midpoint construction avoids privileging either endpoint and provides a common reference for estimating the inter-slice motion.

From the aligned features, we construct the symmetric statistics $m^{(\ell)} = (\bar{f}_0^{(\ell)} + \bar{f}_1^{(\ell)})/2$, $a^{(\ell)} = |\bar{f}_1^{(\ell)} - \bar{f}_0^{(\ell)}|$, and $p^{(\ell)} = \bar{f}_0^{(\ell)} \odot \bar{f}_1^{(\ell)}$, together with the signed directional difference $d^{(\ell)} = \bar{f}_1^{(\ell)} - \bar{f}_0^{(\ell)}$. A symmetric branch models endpoint-invariant correspondence, while a gated directional branch retains the orientation of the deformation:

$$\begin{aligned} s^{(\ell)} &= m^{(\ell)} + \mathcal{B}_{\mathrm{sym}}\Big([m^{(\ell)}, a^{(\ell)}, p^{(\ell)}]\Big), \\ r^{(\ell)} &= d^{(\ell)} + \mathcal{G}^{(\ell)}\Big([m^{(\ell)}, a^{(\ell)}, p^{(\ell)}]\Big) \odot \mathcal{B}_{\mathrm{dir}}\Big([\bar{f}_0^{(\ell)}, \bar{f}_1^{(\ell)}, d^{(\ell)}]\Big), \\ c^{(\ell)} &= [s^{(\ell)}, r^{(\ell)}]. \end{aligned} \tag{28}$$

The fused representation is refined progressively from coarse to fine. Starting from the second decoding level, the current correlation feature is augmented with the upsampled hidden context from the previous coarser level. A refinement head predicts a residual velocity update and the context for the next stage, yielding

$$\begin{aligned} \tilde{c}^{(\ell)} &= \begin{cases} c^{(\ell)}, & \ell = 3, \\ [c^{(\ell)}, \uparrow h^{(\ell+1)}], & \ell < 3, \end{cases} \\ (\Delta v^{(\ell)}, h^{(\ell)}) &= \mathcal{R}^{(\ell)}(\tilde{c}^{(\ell)}), \\ v^{(\ell)} &= \begin{cases} \Delta v^{(\ell)}, & \ell = 3, \\ \mathrm{Resize}(v^{(\ell+1)}) + \Delta v^{(\ell)}, & \ell = 2, 1, 0. \end{cases} \end{aligned} \tag{29}$$

At the finest resolution, we replace the full fusion-refinement block with a lightweight residual head that uses the midpoint-aligned features, the propagated context, and the current velocity estimate to predict the final update. This reduces full-resolution computation while retaining accurate boundary localization.

Diffeomorphic integrator. The final decoder output $v = v^{(0)}$ is represented in pixel units and converted into diffeomorphic displacement fields using scaling and squaring Ashburner (2007). For an arbitrary scaled velocity field w, the integration is performed as

$$u^{(0)} = \frac{w}{2^N}, \qquad u^{(n+1)} = u^{(n)} + \mathcal{W}(u^{(n)}, u^{(n)}), \quad n = 0, \ldots, N-1, \qquad \mathrm{Exp}(w) = u^{(N)}. \tag{30}$$

All velocity and displacement fields follow the same pixel-unit convention, and geometric sampling uses `align_corners=True` throughout.

Continuous-time field generation and warping. Because all query positions share the same SVF, arbitrary intermediate locations are generated simply by scaling v before integration. For each $t \in$

$\mathcal{T}$, we compute the forward physical deformations and the corresponding backward sampling fields as

$$\begin{aligned} u_{0\to t} &= \mathrm{Exp}(tv), & u_{1\to t} &= \mathrm{Exp}((t-1)v), \\ u_{t\to 0} &= \mathrm{Exp}(-tv), & u_{t\to 1} &= \mathrm{Exp}((1-t)v), \\ \tilde{I}_{0\to t} &= \mathcal{W}(I_0, u_{t\to 0}), & \tilde{I}_{1\to t} &= \mathcal{W}(I_1, u_{t\to 1}). \end{aligned} \tag{31}$$

In implementation, the scaled velocity fields for all query times are packed along the batch dimension and integrated jointly. The resulting transported drafts $\{\tilde{I}_{0\to t}, \tilde{I}_{1\to t}\}_{t\in\mathcal{T}}$ are passed to Phase II, while the same Phase I model can be queried at arbitrary continuous positions without modifying the network parameters.

D.2 Training Objectives

Phase I is trained to estimate a stationary velocity field (SVF) whose continuous-time transports are consistent with the ground-truth intermediate slices while remaining geometrically regular. Given the predicted SVF v, the two endpoint sections are transported to each training query position $t \in \mathcal{T}$, producing $\tilde{I}_{0\to t}$ and $\tilde{I}_{1\to t}$.

We optimize Phase I with an image alignment objective together with smoothness and folding regularization:

$$\mathcal{L}_{\text{Phase I}} = \mathcal{L}_{\text{align}} + \lambda_{\text{smooth}} \left(\|\nabla_x v\|_2^2 + \|\nabla_y v\|_2^2 \right) + \lambda_{\text{jac}} \mathbb{E}[\max(0, -\det J_\phi)] \,, \tag{32}$$

where $\phi = \mathrm{Exp}(v)$ denotes the integrated deformation. The alignment loss $\mathcal{L}_{\text{align}}$ averages the supervision from both endpoint-driven transports over all training query positions and combines local normalized cross-correlation (LNCC) with image-gradient consistency. The smoothness term regularizes spatial variations of the SVF, while the Jacobian penalty discourages local folding of the integrated deformation.

D.3 Training Details

We train Phase I for 600 epochs using AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-4}, and an initial learning rate of 5×10^{-4}. A 10-epoch warm-up is used, followed by learning-rate decay to a minimum of 5×10^{-5}. The scaling-and-squaring integrator uses 7 steps. The loss weights are set to $\lambda_{\text{grad}} = 2.0$, $\lambda_{\text{smooth}} = 0.005$, and $\lambda_{\text{jac}} = 0.0005$.

E Additional Details on Phase II

E.1 Network Architecture

Phase II takes the two transported drafts from Phase I, $\tilde{I}_{0\to t}$ and $\tilde{I}_{1\to t}$, together with the continuous query time t, and reconstructs the target slice I_t. The architecture consists of three components: a compact prior extraction network, a prior-guided reconstruction network, and a diffusion-based prior predictor.

E.1.1 Prior Extraction Network

The prior extractor learns a compact intermediate prior representation (IPR) that summarizes slice-specific restoration cues. During teacher training, the two transported drafts and the ground-truth target slice are first rearranged by PixelUnshuffle with scale factor 4, concatenated, and encoded by a compact convolutional network $\mathcal{E}_{\text{pri}}$. To make the prior aware of the queried axial position, a sinusoidal time encoding followed by an MLP is added to the compact representation:

$$\begin{aligned} z_t^\star &= \mathcal{E}_{\text{pri}}\Big([\mathcal{U}_4(\tilde{I}_{0\to t}), \mathcal{U}_4(\tilde{I}_{1\to t}), \mathcal{U}_4(I_t)] \Big) \,, \\ \psi(t) &= \mathrm{MLP}(\mathrm{PE}(t)), \qquad \bar{z}_t = z_t + \psi(t), \end{aligned} \tag{33}$$

where $z_t = z_t^\star$ during teacher training and $z_t = \hat{z}_t$ when the diffusion-predicted prior is used at inference time.

E.1.2 PRIOR-GUIDED RECONSTRUCTION NETWORK

Given the two transported drafts and the time-aware prior $\bar{z}_t$, the reconstruction network first predicts a coarse slice and then restores high-frequency details with a lightweight texture branch. The spatial backbone follows a four-level U-shaped architecture. The two drafts are first downsampled and fused at the input stage; the shallow encoder is prior-free, whereas deeper encoder, latent, and decoder blocks are modulated by $\bar{z}_t$. This design injects the compact restoration prior mainly into low-resolution semantic features while preserving high-resolution spatial details.

At the bottleneck, a Query-Time-Modulated State Space Router (QTM-SSR) models coherent evolution across the ordered query positions. The routed features are then progressively decoded with skip connections. Two lightweight Deformable Temporal Shift (DTS) modules are inserted around the shallow decoder to compensate for local non-rigid misalignment. The resulting feature is projected to the image domain by an output convolution and PixelShuffle, yielding the coarse prediction P_t.

To further recover membrane boundaries and ultrastructural textures, we use a lightweight full-resolution refinement branch. Shallow features extracted from the two drafts are interpolated according to the normalized query coordinate τ_t, and the resulting feature is combined with a detached coarse prediction to estimate a bounded texture residual:

$$\begin{aligned} f_t^{\mathrm{mix}} &= \tanh\Big((1-\tau_t)\mathcal{H}(\tilde{I}_{0\to t}) + \tau_t \mathcal{H}(\tilde{I}_{1\to t})\Big), \\ \Delta_t^{\mathrm{tex}} &= 0.1\,\tanh\big(\mathcal{G}_{\mathrm{tex}}[\mathrm{sg}(P_t), f_t^{\mathrm{mix}}, \tau_t \mathbf{1}]\big), \\ \hat{I}_t &= P_t + \sigma(\omega)\Delta_t^{\mathrm{tex}}. \end{aligned} \tag{34}$$

Here $\mathrm{sg}(\cdot)$ denotes stop-gradient, ω is a learnable scalar, and for the k-th query in an ordered group of length T we use $\tau_t = k/(T+1)$.

Query-Time-Modulated State Space Router. The QTM-SSR operates at the lowest-resolution latent space, where the feature tensor is rearranged from (BT, C, H, W) to (BHW, T, C) such that each spatial location forms a sequence along the ordered query dimension. After normalization, projection, and depthwise temporal convolution, the router produces a content stream u_t, a gating stream g_t, and the dynamic state-space parameters (δ_t, B_t, C_t).

Unlike a fixed-step state-space model, the effective transition step is explicitly modulated by the query-time scale s_t. The forward state update and bidirectional fusion are summarized as

$$\begin{aligned} \boldsymbol{\Delta}_t &= \mathrm{softplus}(W_\Delta \delta_t + b_\Delta) \odot s_t, \\ h_t^{\rightarrow} &= e^{\boldsymbol{\Delta}_t A} \odot h_{t-1}^{\rightarrow} + \frac{e^{\boldsymbol{\Delta}_t A} - 1}{A} \odot B_t \odot u_t, \\ r_t &= W_o\big(\big(h_t^{\rightarrow} C_t + h_t^{\leftarrow} C_t\big) \odot \phi(g_t)\big). \end{aligned} \tag{35}$$

The backward state $h_t^{\leftarrow}$ is obtained analogously by scanning the query sequence in reverse order. The fused representation is reshaped back to the spatial tensor domain and combined with the latent input through a residual connection. In this way, QTM-SSR provides global query-aware propagation at the deepest feature level.

Deformable Temporal Shift Module. While QTM-SSR captures long-range evolution in latent space, high-resolution features still require local alignment between nearby query positions. The DTS module therefore applies deformable alignment only to a subset of channels, while the remaining channels directly bypass the module.

For the motion-sensitive feature X_t^a, an offset is predicted from the current feature and its previous-query counterpart, scaled by the query-time factor s_t, and used to warp the previous feature. A learned gate then adaptively fuses the aligned and current representations:

$$\begin{aligned} \Delta p_t &= s_t \odot \mathcal{F}_{\mathrm{flow}}([X_t^a, X_{t^-}^a]), \\ \widehat{X}_{t^-}^a &= \mathcal{W}(X_{t^-}^a, \Delta p_t), \\ M_t &= \sigma\Big(\mathcal{F}_{\mathrm{gate}}([X_t^a, \widehat{X}_{t^-}^a])\Big), \\ Y_t^a &= (1 - M_t) \odot X_t^a + M_t \odot \widehat{X}_{t^-}^a. \end{aligned} \tag{36}$$

Here t^- denotes the previous query position. Two independent DTS modules are used before and after the shallow decoder, providing complementary local alignment to the global propagation performed by QTM-SSR.

E.1.3 Diffusion Prior Predictor Network

The teacher prior $z_t^\star$ depends on the ground-truth target slice and is therefore unavailable at inference time. We consequently learn a conditional diffusion model that predicts this compact prior using only the two transported drafts. A lightweight condition encoder summarizes the draft pair into a descriptor $c_t = \mathcal{E}_{\mathrm{cond}}([\tilde{I}_{0\to t}, \tilde{I}_{1\to t}])$. Because diffusion is performed in the low-dimensional prior space rather than the image space, the denoiser can be implemented with a lightweight residual MLP instead of a U-Net.

During training, Gaussian noise is added to the teacher prior, and the denoiser predicts the injected noise conditioned on the draft representation:

$$\begin{aligned} z_t^s &= \sqrt{\bar{\alpha}^s}\, z_t^\star + \sqrt{1-\bar{\alpha}^s}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \\ \hat{\epsilon}_t^s &= \epsilon_\theta(z_t^s, s, c_t) = \mathcal{M}_{\mathrm{denoise}}([c_t, \tilde{s}, z_t^s]). \end{aligned} \tag{37}$$

Here $\tilde{s}$ denotes the normalized diffusion step. At inference time, the reverse process starts from Gaussian noise and produces the predicted prior $\hat{z}_t$, which is subsequently combined with the continuous time embedding and used to condition the reconstruction network.

E.2 Training Objective

Phase II is trained with a standard reconstruction objective between the predicted slice $\hat{I}_t$ and the corresponding ground-truth slice I_t. We adopt the Charbonnier loss for robust pixel-wise supervision:

$$\mathcal{L}_{\mathrm{rec}} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\left(\hat{I}_t^{(i)} - I_t^{(i)}\right)^2 + \epsilon^2}, \tag{38}$$

where N denotes the total number of pixels and ϵ is a small constant for numerical stability. This reconstruction objective is used to optimize the Phase II reconstruction network.

E.3 Training Details

Teacher stage. We first jointly train the prior extractor and reconstruction network for $520{,}000$ iterations using Adam with an initial learning rate of 1.2×10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The learning rate is scheduled using cosine annealing with a $1{,}000$-iteration warm-up and a minimum learning rate of 3.0×10^{-5}. We additionally employ exponential moving averaging (EMA) with a decay of 0.999 and mixed-precision training. The reconstruction network is optimized using the reconstruction objective described in Section E.2.

Student stage. The student model is also trained for $520{,}000$ iterations. During the first $9{,}000$ iterations, only the diffusion prior predictor is warmed up to approximate the teacher prior. Afterwards, the predicted prior is fed into the reconstruction network and the full student model is jointly optimized for the remaining iterations. The optimizer and learning-rate schedule follow those used in the teacher stage.

Table 3: Continuous-query reconstruction performance of IsoCTG. The model is trained only at $t = \{2/8, 4/8, 6/8\}$ (Seen), while $t = \{1/8, 3/8, 5/8, 7/8\}$ are unseen during training. All positions are evaluated using the same checkpoint without retraining. Each entry reports PSNR/SSIM.

Dataset	$t = \frac{1}{8}$ Unseen	$t = \frac{2}{8}$ Seen	$t = \frac{3}{8}$ Unseen	$t = \frac{4}{8}$ Seen	$t = \frac{5}{8}$ Unseen	$t = \frac{6}{8}$ Seen	$t = \frac{7}{8}$ Unseen
mus-liver	22.36 / 0.76	**22.58 / 0.78**	22.17 / 0.75	**22.46 / 0.77**	22.21 / 0.75	**22.51 / 0.78**	22.30 / 0.76
mus-cortex-3	19.92 / 0.79	**20.24 / 0.81**	19.86 / 0.78	**20.15 / 0.80**	19.79 / 0.78	**20.19 / 0.81**	19.95 / 0.79
fly-protocerebral-bridge	18.23 / 0.72	**18.51 / 0.74**	18.11 / 0.71	**18.42 / 0.73**	18.16 / 0.71	**18.47 / 0.74**	18.27 / 0.72

F More Results

F.1 Continuous-Query Generalization

A key property of IsoCTG is its ability to reconstruct intermediate sections at arbitrary axial query positions rather than being restricted to a fixed discrete interpolation grid. To directly evaluate this property, we conduct an additional continuous-query generalization experiment in which the model is trained only on a sparse subset of axial positions and evaluated at both seen and unseen query locations. Specifically, we normalize the interval between two observed endpoint sections to $t \in [0, 1]$ and use $t = \{2/8, 4/8, 6/8\}$ as the training query positions. At test time, the same checkpoint is evaluated at all seven intermediate locations $t = \{1/8, \ldots, 7/8\}$. Therefore, $t = \{1/8, 3/8, 5/8, 7/8\}$ correspond to query positions that are never directly observed during training. No fine-tuning or scale-specific retraining is performed for these unseen locations. This protocol allows us to examine whether the learned reconstruction function generalizes between the discrete positions used for supervision.

As shown in Table 3, query positions observed during training generally achieve the highest reconstruction accuracy, while the interleaved unseen positions remain close in both PSNR and SSIM. Across all three datasets, the degradation at unseen locations is modest and no abrupt performance drop is observed between neighboring seen and unseen queries. For example, on *mus-liver*, the unseen positions at $t = 3/8$ and $t = 5/8$ achieve 22.17/0.75 and 22.21/0.75, respectively, compared with 22.46/0.77 at the neighboring seen position $t = 4/8$. Similar trends are observed on *mus-cortex-3* and *fly-protocerebral-bridge*.

Interestingly, reconstruction quality is not determined solely by whether a query position is seen during training. Unseen positions closer to the observed endpoints can still achieve strong performance, whereas locations near the center of the interval are sometimes more challenging due to the larger amount of geometric and appearance variation that must be inferred from both endpoints. Overall, the smooth performance variation across seen and unseen positions indicates that IsoCTG does not simply memorize a fixed interpolation grid and can generalize to sparsely supervised intermediate locations using a single checkpoint without retraining, supporting its continuous-query capability.

F.2 More visualization results

More visualizations are shown in the Figures 7-10.

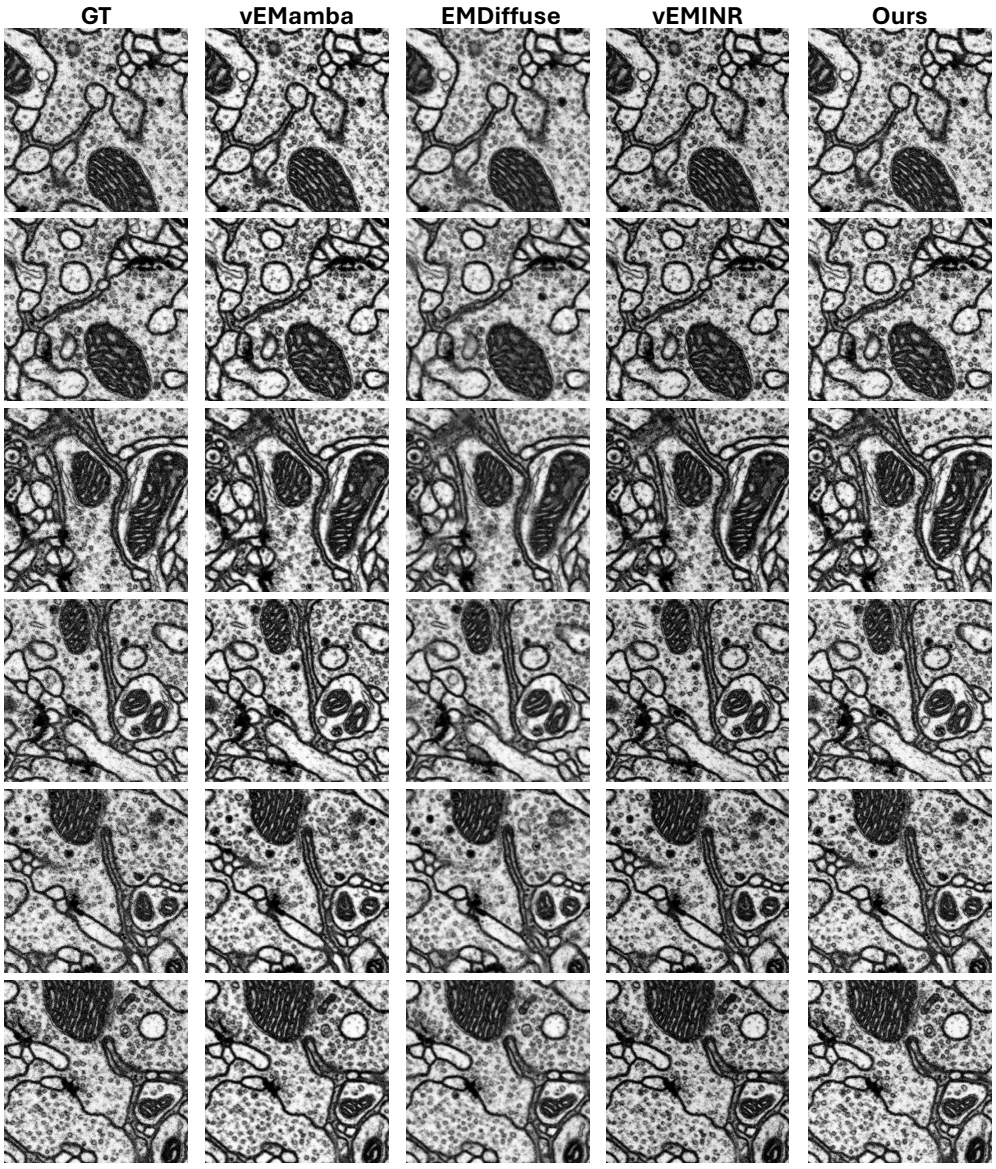


Figure 7: Isotropic xy-slice reconstruction results on anisotropic datasets.

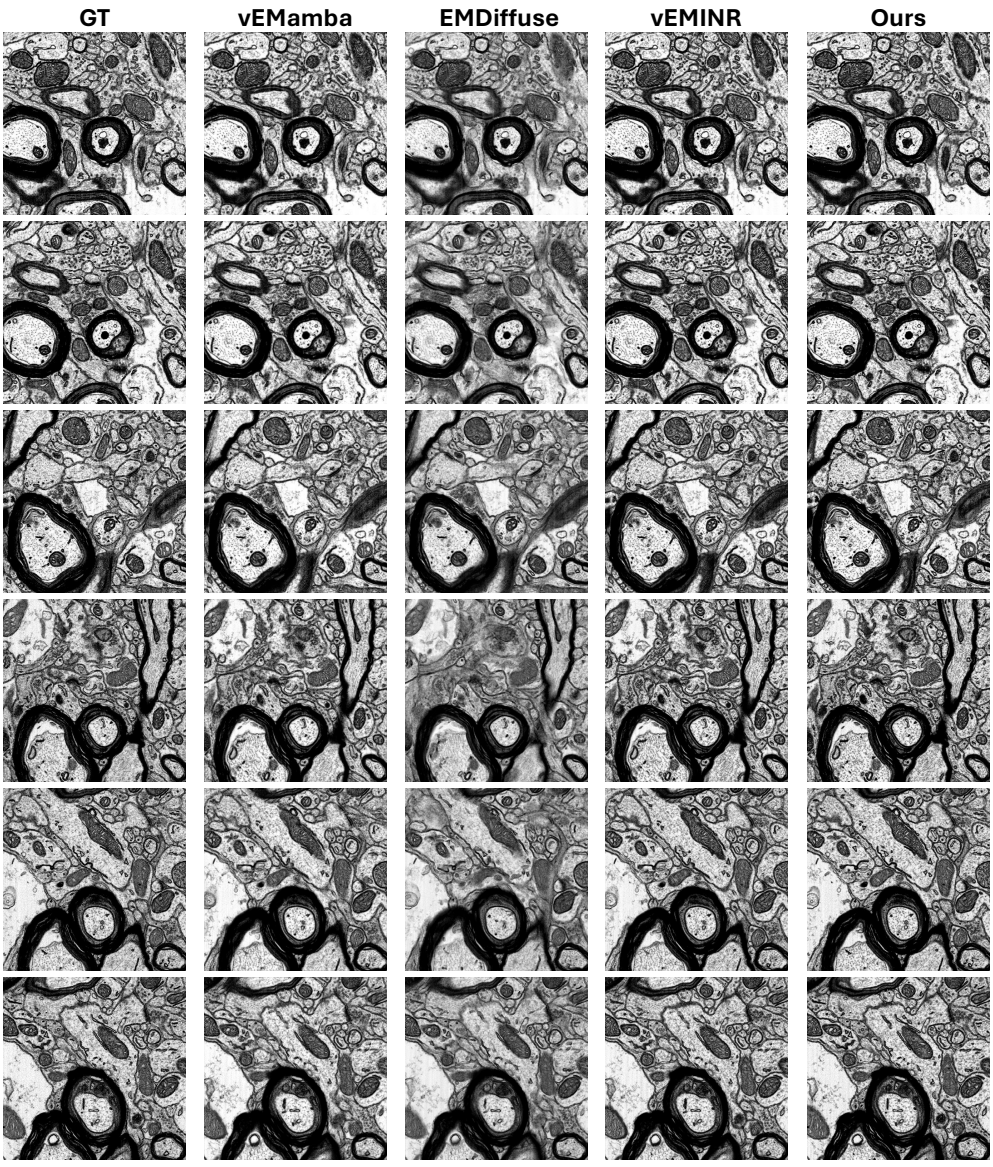


Figure 8: Isotropic xy-slice reconstruction results on anisotropic datasets.

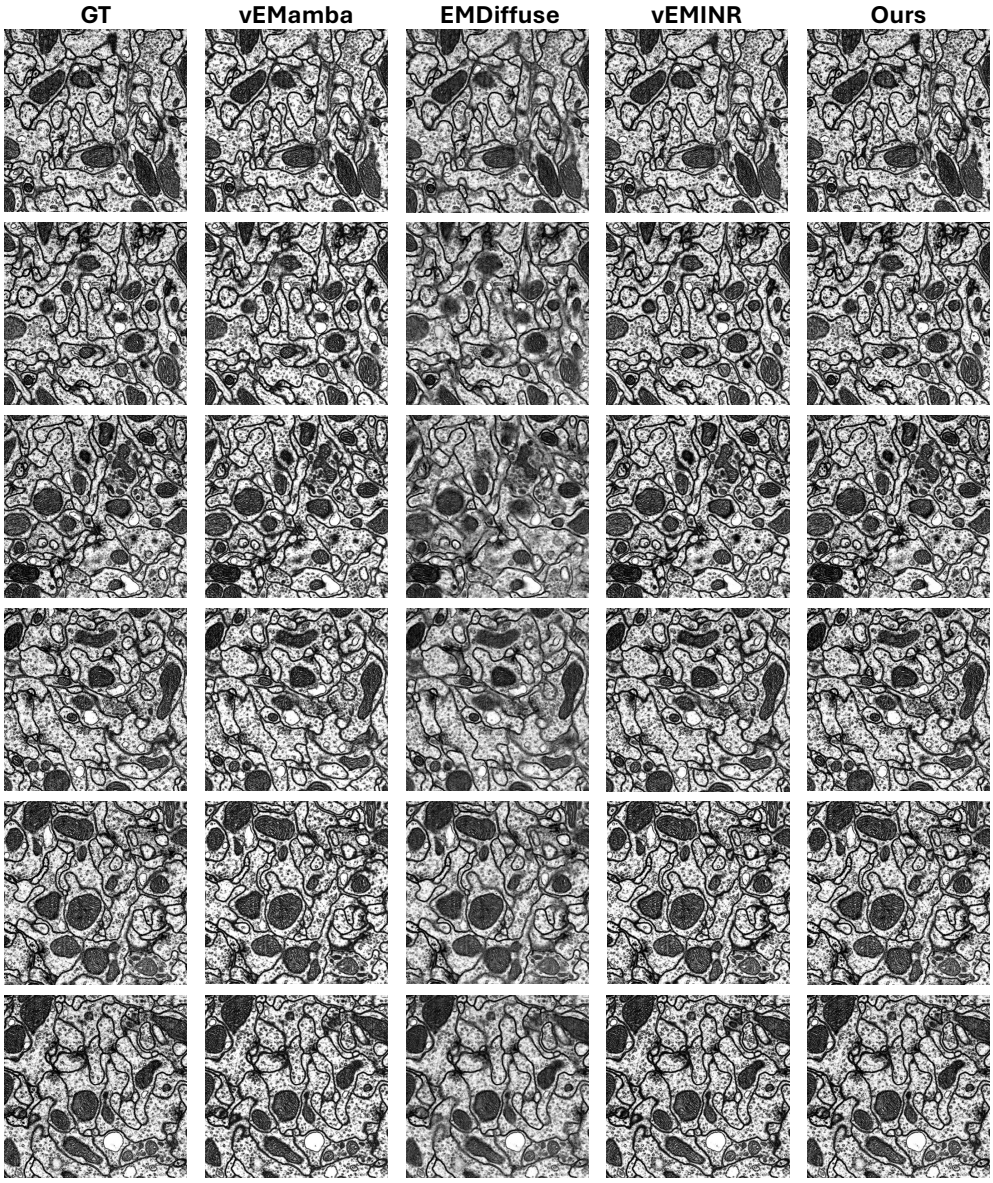


Figure 9: Isotropic xy-slice reconstruction results on anisotropic datasets.

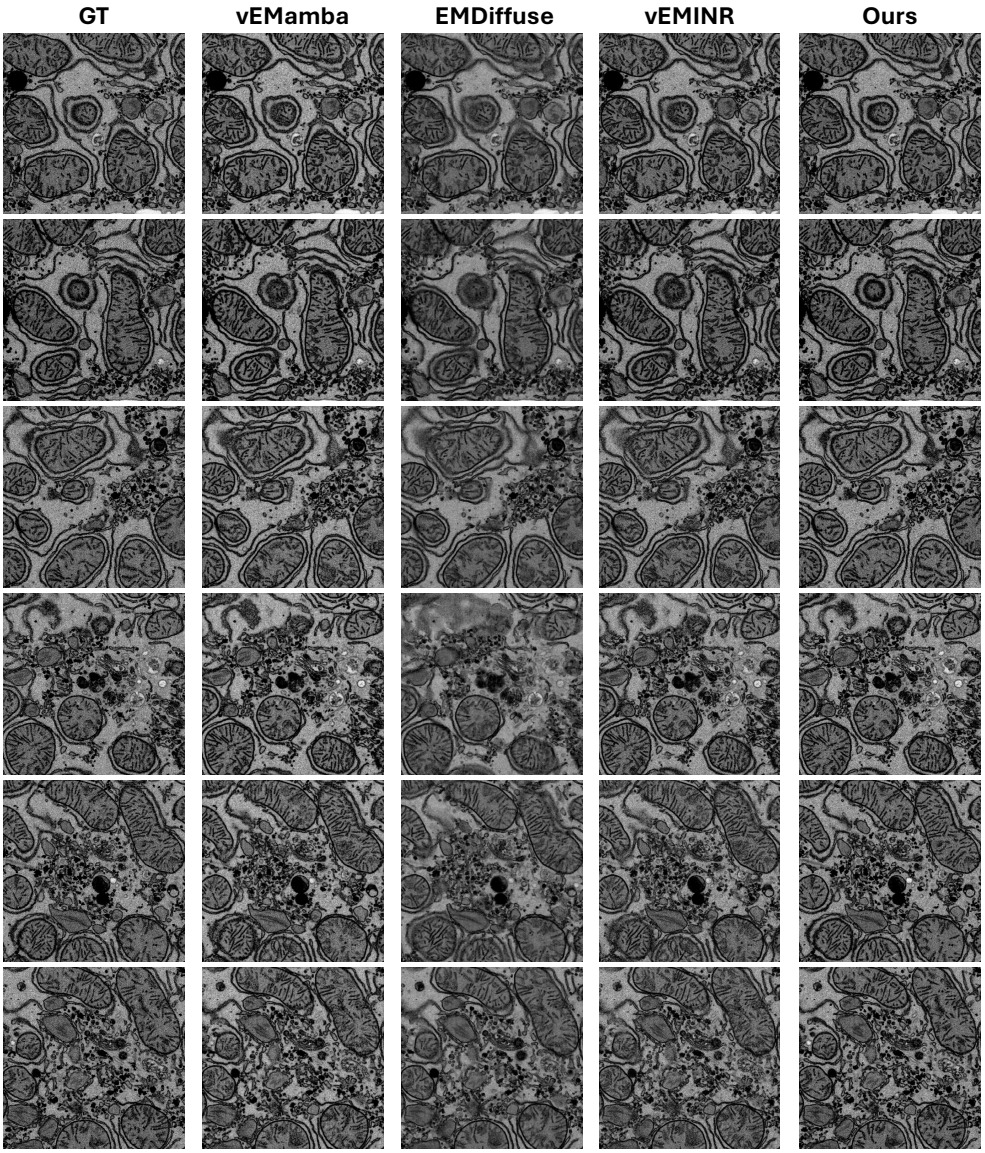


Figure 10: Isotropic xy-slice reconstruction results on anisotropic datasets.