
Unsupervised Trajectory Optimization for 3D Registration in Serial Section Electron Microscopy using Neural ODEs

Zhenbang Zhang*
MBZUAI; Shandong University
zhangzhenbang2021@gmail.com

Jingtong Feng*
Shandong University
jingtongf404@gmail.com

Hongjia Li
Beijing Institute of Technology
lihongjia96@gmail.com

Haythem El-Messiry
Canadian University Dubai
haythem.elmessiry@cud.ac.ae

Zhiqiang Xu[†]
MBZUAI
zhiqiang.xu@mbzuai.ac.ae

Renmin Han[†]
Shandong University
hanrenmin@sdu.edu.cn

Abstract

Series Section Electron Microscopy (ssEM) has emerged as a pivotal technology for deciphering nanoscale biological architectures. Three-dimensional (3D) registration is a critical step in ssEM, tasked with rectifying axial misalignments and nonlinear distortions introduced during serial sectioning. The core scientific challenge lies in achieving distortion mitigation without erasing the natural morphological deformations of biological tissues, thereby enabling faithful reconstruction of 3D ultrastructural organization. In this study, we present a paradigm-shifting optimization framework that rethinks 3D registration through the lens of manifold trajectory optimization. We propose the first continuous trajectory dynamics formulation for 3D registration and introduce a novel optimization strategy. Specifically, we introduce a dual optimization objective that inherently balances global trajectory smoothness with local structural preservation, while developing a solver that combines Gauss-Seidel iteration with Neural ODEs to systematically integrate biophysical priors with data-driven deformation compensation. Extensive experiments on multiple datasets spanning diverse tissue types demonstrate our method’s superior performance in structural restoration accuracy and cross-tissue robustness.

1 Introduction

Series Section Electron Microscopy (ssEM) has emerged as a powerful technology for nanoscale Three-dimensional (3D) visualization of biological systems. Its impact spans diverse domains including neuroscience connectomics [84, 79], developmental biology [19, 43], and clinical diagnostics [29]. Recently, MICrONS project [1, 13, 15, 4] utilizes ssEM as the fundamental method for neural circuit reconstruction. The ssEM workflow typically involves a series of computational steps, including two-dimensional (2D) stitching [21], 3D registration [53, 82], and 3D segmentation [40, 47].

*Co-First Authors.

[†]Corresponding Authors.

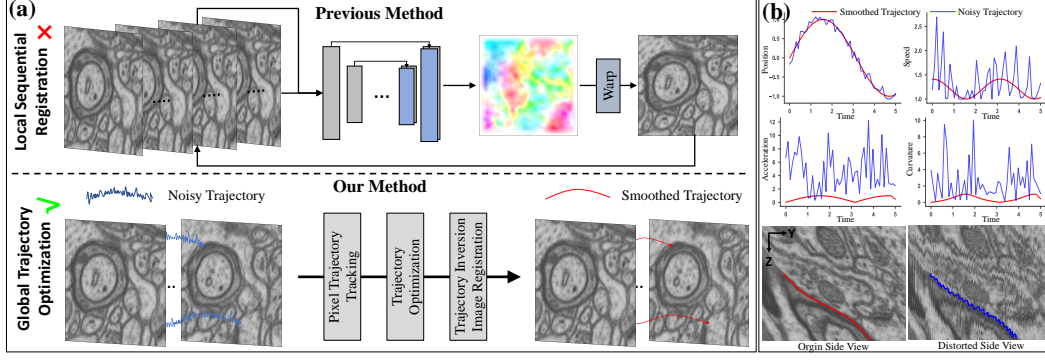


Figure 1: **Trajectory analysis and inspiration:** Left: Previous methods is limited to adjacent registration, while our method achieves a paradigm shift by global trajectory optimization. Right: The upper-right demonstrates **position/velocity/acceleration/curvature** comparisons between smoothed and noisy 1D trajectories. The lower-right displays the side view of the original and distorted data. This inspires us to explore the inner link between trajectory curvature mutation and nonlinear distortion from the perspective of biophysical motion.

Among these, 3D registration is particularly critical, as it corrects axial misalignments and nonlinear distortions, ultimately determining the accuracy and reliability of the final 3D structure.

The fundamental challenge stems from the topological entanglement of biological signal and technical noise – specifically, the indistinguishability between natural cellular morphological deformation and nonlinear distortion induced by sample preparation [76, 51]. Furthermore, state-of-the-art equipment now generates TB-scale image data daily [77]. The massive data volume and error accumulation in long sequences [36] greatly complicate fast and accurate 3D registration.

Several effective methods have been developed [56, 80, 86, 76, 42, 82]. However, these methods often fail to account for natural deformations between slices, and prioritizing pixel-level similarity may erase biologically meaningful changes, compromising the reconstructed structure. Recently, Zhang et al. [82] decouple natural deformation from nonlinear distortion by modeling them as low and high-frequency components. This insight motivated us to investigate the intrinsic characteristics of nonlinear distortion through the lens of biological motion coherence. We discovered that nonlinear distortion is essentially a sudden change in local curvature. As shown in Figure 1(b), anatomical structures inherently exhibit smooth manifold properties, maintaining evolutionary consistency in morphological changes. Mechanical sectioning, however, disrupts this continuity, introducing abrupt curvature mutations that contradict tissue biomechanics.

Building upon this, we propose a novel optimization framework, NeuroTrajReg, which redefines 3D registration as a motion trajectory optimization problem. We establish a continuous trajectory dynamics formulation for ssEM, integrating spatiotemporal constraints to effectively eliminate nonlinear distortions while preserving the natural evolution of tissue morphology. We introduce a dual optimization objective that inherently balances global trajectory smoothness and local structural preservation, while developing a solver that combines Gauss-Seidel iteration with Neural ODEs to systematically integrate biophysical priors with data-driven deformation compensation. Extensive experiments spanning diverse tissue types demonstrate NeuroTrajReg’s superior performance in structural restoration accuracy and robustness across different tissue types. The main contributions of this paper are summarized as follows:

- We analyze the nonlinear distortions from the perspective of biological motion coherence. Based on this analysis, we redefine 3D registration as a continuous manifold trajectory optimization problem and propose the first continuous trajectory dynamics formulation.
- We propose a novel framework that integrates biophysical priors with data-driven deformation compensation, achieving a theory-guided yet data-corrected approach.
- Extensive experiments demonstrate that NeuroTrajReg achieves superior performance in accurately and faithfully reconstructing biological 3D structures while maintaining cross-tissue robustness.

2 Related works

3D registration for series section electron microscopy. Several software packages have been developed for 3D registration [39, 38, 66]. Among them, TrakEM2 [56] is one of the most widely used tools, performing 3D registration by iteratively optimizing a spring-connected particle system. Recently, deep learning-based methods have emerged as promising alternatives for 3D registration [80, 86, 42]. SEAMLeSS [53] improves robustness through vector voting and achieves global registration via combined attenuation transformations. Recently, Zhang et al. [82] proposed a Gaussian filter-based 3D registration method that approaches the problem from a frequency-domain perspective.

Medical image registration. Medical image registration [85, 48, 24, 16], closely related to 3D registration, has achieved notable progress [85, 52, 5, 83, 20, 22]. Existing methods are typically categorized by their transformation models. Dense models [2, 9, 37, 8] estimate voxel-wise mappings to form dense deformation fields, whereas interpolation-based models [30, 70, 58, 61] approximate deformations using basis functions (e.g., B-splines) over spatial grids. Despite their success, these approaches mainly address image pairs. In contrast, 3D registration demands handling nonlinear distortions while maintaining axial continuity.

Neural ordinary differential equations. Neural Ordinary Differential Equations (Neural ODEs) [6] integrate differential equation solvers into neural networks, enabling continuous representations with adaptive computation and enhanced parameter efficiency. Formally, given hidden state $\mathbf{h}(t)$, a Neural ODEs defines its dynamics through a parameterized function f_θ :

$$\frac{d\mathbf{h}(t)}{dt} = f_\theta(\mathbf{h}(t), t), \quad (1)$$

and the final state is obtained by solving $\mathbf{h}(t_1) = \mathbf{h}(t_0) + \int_{t_0}^{t_1} f_\theta(\mathbf{h}(t), t) dt$. Numerous variants of Neural ODEs have emerged to capture more complex transformations [57, 75, 49, 44, 35, 27]. Neural ODEs have been widely applied across various fields [50, 31, 55, 68]: Vid-ODE [50] enables continuous-time video synthesis, NODEO [74] adapts it for deformable image registration, and Latent ODE [55] models continuous-time sequences.

Reference line smoothing algorithms. In autonomous driving [81, 65], the reference line smoothing algorithm [12, 60, 17] geometrically optimizes global paths to generate smooth trajectories with curvature continuity, kinematic feasibility, and minimal deviation from the original reference. Its primary role involves eliminating curvature discontinuities between discrete waypoints, constraining maximum curvature within vehicle kinematic limits. The loss function is formulated as:

$$J = \underbrace{\sum_{i=2}^{n-1} \|\mathbf{p}_{i-1} - 2\mathbf{p}_i + \mathbf{p}_{i+1}\|^2}_{\text{Smoothness}} + \lambda \underbrace{\sum_{i=1}^n \|\mathbf{p}_i - \mathbf{q}_i\|^2}_{\text{Fidelity}} + \underbrace{\sum_{i=1}^n \delta(|\kappa_i| \leq \kappa_{\max})}_{\text{Constraints}}, \quad (2)$$

where the first term penalizes curvature variation via second-order differencing, the second enforces proximity to original waypoints \mathbf{q}_i with adaptive weight λ , and the third applies indicator function $\delta(\cdot)$ to enforce maximum curvature κ_{\max} . Optimization strategies include convex quadratic programming (QP) for real-time computation, sequential QP (SQP) for nonlinear constraint handling [12, 17], spline curves [72, 14], and mass-spring physical simulation [34, 23].

3 Proposed method: NeuroTrajReg

For the problem of 3D registration, we propose a dynamics-based nonlinear distortion correction framework for ssEM image sequences. Given a stack of microscopic images $\{I_z\}_{z=0}^{N-1}$ affected by complex nonlinear distortions, our objective is to reconstruct a 3D structure that faithfully preserves the biological morphology by establishing a registration model with biophysical priors. The core challenge lies in dynamically balancing two conflicting demands: on the one hand, eliminating nonlinear spatial distortions induced by external factors, and on the other hand, preserving the intrinsic morphological characteristics of the biological specimen, such as tissue deformation and developmental topological changes.

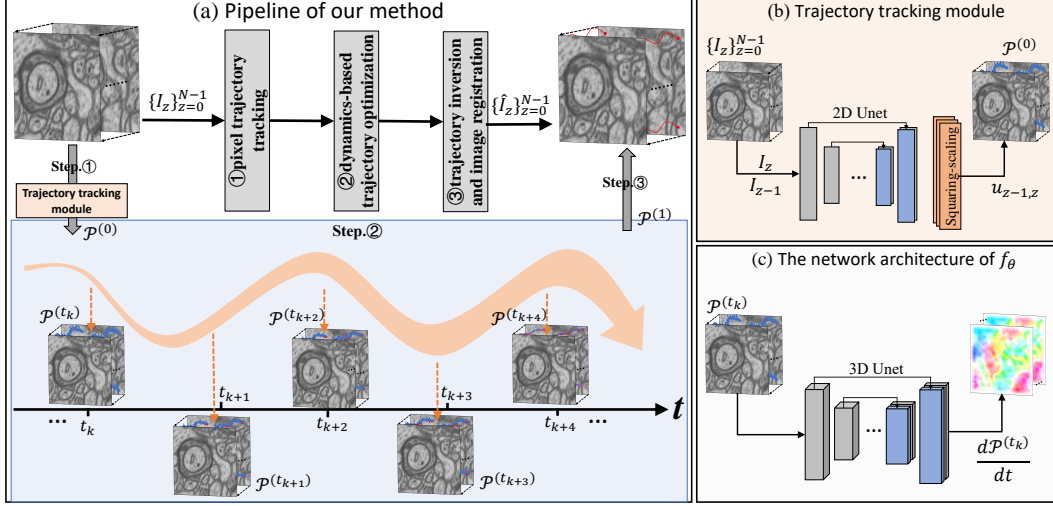


Figure 2: (a) The overall pipeline of NeuroTrajReg. Our approach can be divided into three main components: 1) Pixel trajectory tracking, 2) Dynamics-based trajectory optimization, and 3) Trajectory inversion and image registration. (b) Illustration of the trajectory tracking module. (c) The network architecture of f_θ (Neural ODEs) in dynamics-based trajectory optimization module.

3.1 Problem formulation

In this section, we introduce the concept of pixel-wise motion trajectory tracking and smoothing. Drawing inspiration from the classical Laplace equation [7] and the reference line smoothing algorithms [12], we formulate the 3D registration as a continuous manifold trajectory optimization problem. Specifically, each pixel $p_i \in \mathbb{R}^2$ is treated as a particle within a continuum medium, whose displacement along the slice axis $z \in [z_0, z_{N-1}]$ generates a parameterized trajectory \mathcal{P}_i . While this trajectory ideally follows natural morphological variations, nonlinear distortions induce local curvature abruptness. To address this, we establish spatiotemporal smoothness constraints on pixel trajectories, transforming traditional 3D registration into a physically interpretable continuous trajectory optimization problem. NeuroTrajReg is grounded in three key principles: 1) physical motion coherence in biological tissues; 2) Laplace-based trajectory smoothing; 3) spatiotemporal continuity in microscopy imaging. We model pixel-wise motion from individual dynamics to global trajectory constraints with strict mathematical consistency.

Single-particle dynamics. Let \mathcal{P}_i denote the parameterized trajectory. We formulate the single-particle optimization problem as follows:

$$\min_{\mathcal{P}_i} \left\{ \underbrace{\int_{z_0}^{z_{N-1}} \|\mathcal{P}_i(z) - \mathcal{P}_i^{(0)}(z)\|^2 dz}_{\text{Data fidelity}} + \lambda \underbrace{\int_{z_0}^{z_{N-1}} \|\nabla^2 \mathcal{P}_i(z)\|^2 dz}_{\text{Laplacian smoothness}} \right\}, \quad (3)$$

where $\mathcal{P}_i^{(0)}$ denotes the observed origin trajectory, ∇^2 is the axial Laplacian operator, and $\lambda > 0$ controls the regularization strength. The Laplacian term enforces second-order temporal smoothness by penalizing acceleration discontinuities, effectively filtering biologically implausible instantaneous acceleration changes.

Holistic trajectory constraint. For image stack $\{I_z\}_{z=0}^{N-1}$, we formulate the holistic optimization:

$$\min_{\{\mathcal{P}_i\}} \sum_{i=1}^M \left[\int_{z_0}^{z_{N-1}} \|\mathcal{P}_i(z) - \mathcal{P}_i^{(0)}(z)\|^2 dz + \lambda \int_{z_0}^{z_{N-1}} \|\nabla^2 \mathcal{P}_i(z)\|^2 dz + \mu S_i \right], \quad (4)$$

where M is the total number of trajectories, corresponding to the number of pixels in image I_z , and the spatial coherence term S_i is defined as:

$$S_i := \sum_{j \in \mathcal{N}(i)} \int_{z_0}^{z_{N-1}} \|\mathcal{P}_i(z) - \mathcal{P}_j(z)\|^2 dz, \quad (5)$$

and weighted by $\mu > 0$ for preserving local topologies within each slice, where $\mathcal{N}(i)$ denotes the set of spatial neighbors of point i within the same slice. This dual regularization strategy embeds two fundamental biological principles:

- **Temporal smoothness.** It preserves smooth and stable tissue morphology evolution by constraining axial acceleration discontinuities through Laplacian regularization, preventing unreliable instantaneous trajectory changes.
- **Spatial consistency.** It maintains the local topology between slice pixels by ensuring the consistency of trajectory smoothing for neighboring pixels.

3.2 Trajectory optimization for 3D registration

Our approach, as shown in Figure 2(a), mainly consists of three components: 1) pixel trajectory tracking, 2) dynamics-based trajectory optimization, and 3) trajectory inversion and image registration. Specifically, we perform trajectory tracking for each pixel in the image stack. Then, we apply our proposed trajectory optimization algorithm to obtain smooth pixel trajectories. Afterward, we compute the trajectory displacements and invert them back to the pixel grid to generate a deformation field, which is then used to register the image stack. We will now elaborate on the implementation and contributions of each component.

Pixel trajectory tracking. Given image stack $\{I_z\}_{z=0}^{N-1} \in \mathbb{R}^{N \times 1 \times H \times W}$, our goal is to track the trajectory of each pixel over time, accounting for subpixel displacements. We use the network shown in Figure 2(b) for estimating the displacement field $u_{z-1,z} \in \mathbb{R}^{H \times W \times 2}$ in x -axis and y -axis between adjacent slices. To model the temporal trajectory of every pixel, we introduce a trajectory volume $\mathcal{P} \in \mathbb{R}^{N \times H \times W \times 2}$, where $\mathcal{P}(z)$ stores the spatial coordinate in slice z . This trajectory is initialized as:

$$\mathcal{P}(0, y, x) = (y^0, x^0), \quad (6)$$

indicating that the trajectory of each pixel starts from its position in the first slice. Here, $\mathcal{P}(z, y, x)$ denotes the displacement vector at index (y, x) in slice z .

We recursively propagate the position using the displacement field from the previous slice. But the pixel location $\mathcal{P}(z-1, y, x)$ may not be aligned with the discrete grid of $u_{z-1,z}$, as accumulated displacements often lead to non-integer coordinates. As a result, bilinear interpolation of $u_{z-1,z}$ is required to accurately estimate the flow at subpixel positions. Formally, the trajectory propagation is governed by the transport equation:

$$\mathcal{P}(z, y, x) = \mathcal{P}(z-1, y, x) + \tilde{u}_{z-1,z}(\mathcal{P}(z-1, y, x)), \quad (7)$$

where $\tilde{u}_{z-1,z}(\cdot)$ denotes the interpolated displacement field.

Dynamics-based trajectory optimization. According to Eq. (3), we model trajectory smoothing as the weighted minimization of discrete curvature energy and deviations from the original trajectory. To solve this optimization objective, we employ the Gauss-Seidel method [18] to iteratively smooth the trajectory. The Gauss-Seidel method is a classical algorithm for solving linear systems. Its key characteristic lies in accelerating convergence by immediately incorporating the most recently updated values. The corresponding Gauss-Seidel iteration is given by:

$$\mathcal{P}^{(k+1)}(z) = \frac{\mathcal{P}^{(k+1)}(z-1) + \mathcal{P}^{(k)}(z+1) + \lambda \mathcal{P}^{(0)}(z)}{2 + \lambda}, \quad z = 1, \dots, N-1. \quad (8)$$

with boundary values fixed as $\mathcal{P}^{(k)}(0) \equiv \mathcal{P}^{(0)}(0)$ and $\mathcal{P}^{(k)}(N-1) \equiv \mathcal{P}^{(0)}(N-1)$. In each iteration k , the internal positions z are updated from the 1 to the $N-1$, where each $\mathcal{P}^{(k+1)}(z)$ is computed using the most recent value of its predecessor $\mathcal{P}^{(k+1)}(z-1)$ and the previous value of its successor $\mathcal{P}^{(k)}(z+1)$. A detailed derivation is provided in the supplementary material.

While the Gauss-Seidel method provides a theoretically grounded approach for solving linear systems through explicit point-wise updates with guaranteed convergence, its practical effectiveness is inherently limited by discrepancies between idealized theoretical models and real-world scenarios. Contrastingly, Neural ODEs [6] excel in learning continuous dynamical systems through implicit dynamic modeling, effectively capturing complex nonlinear deviations.

Given the exceptional capability of Neural ODEs in modeling continuous dynamic systems [69, 31, 46], we use Neural ODEs to reformulate trajectory smoothing as the dynamic evolution of a continuous trajectory. The network architecture of f_θ in Neural ODEs is shown in Figure 2(c). The trajectory constraints are enforced through integration of velocity field $\mathcal{V} = \{v_z\}_{z=1}^{N-1}$ satisfying:

$$\mathcal{V}^{(t)} = \frac{d\mathcal{P}^{(t)}}{dt} = f_\theta(\mathcal{P}^{(t)}, t), \quad \mathcal{P}^{(t_{k+1})} = \mathcal{P}^{(t_k)} + \int_{t_k}^{t_{k+1}} \mathcal{V}^{(t)} dt, \quad (9)$$

where f_θ is a neural network with trainable parameters θ .

Inspired by Universal Differential Equations (UDEs) [54], we propose a dynamically adaptive optimization system that synergistically combines the explicit iterative convergence of Gauss-Seidel with the implicit dynamic modeling capabilities of Neural ODEs:

$$\mathcal{P}^{(t_N)} = (1 - \alpha) \underbrace{\mathcal{G}(\mathcal{P}^{(t_0)})}_{\text{Gauss-Seidel core}} + \alpha \underbrace{\text{ODESolver}[f_\theta, \mathcal{P}^{(t_0)}, [t_0, t_N]]}_{\text{Neural compensator}}, \quad (10)$$

where $\mathcal{G}(\cdot)$ denotes the Gauss-Seidel operator, which takes the initial trajectory at time t_0 as input and outputs a smoothed trajectory at time t_N . $\text{ODESolver}[\cdot]$ represents the neural dynamic correction over time interval $[t_0, t_N]$. The adaptive weight $\alpha \in [0, 1]$ evolves through:

$$\alpha = \sigma\left(\beta \cdot \left(1 - \frac{\Delta^{(t_N)}}{\Delta^{(t_0)}}\right)\right), \quad (11)$$

where $\sigma(\cdot)$ denotes the sigmoid activation ensuring smooth transitions, β controls the adaptation rate sensitivity, and $\Delta^{(t_k)} = \|\mathcal{G}(\mathcal{P}^{(t_k)}) - \mathcal{P}^{(t_k)}\|$ quantifies theoretical operator progress.

Therefore, our architecture follows a *theory-guided, data-corrected approach*, incorporating an adaptive blending module that automatically balances theoretical components and corrective terms. It takes advantage of Gauss-Seidel for rapid initial convergence, ensuring numerical stability and accelerating optimization. Neural ODEs are then employed to capture higher-order nonlinear effects and compensate for the gaps between the model and reality.

Trajectory inversion and image registration. After trajectory optimization, we obtain the resulting trajectories $\mathcal{P}^{(t_N)} \in \mathbb{R}^{N \times H \times W \times 2}$, where each element $\mathcal{P}_i^{(t_N)}(z) \in \mathbb{R}^2$ represents the evolved coordinate of the i -th trajectory $\mathcal{P}_i^{(t_N)}$ at slice z . Based on this, the displacement $\phi \in \mathbb{R}^{N \times H \times W \times 2}$ can be computed by comparing it with the observed original trajectory $\mathcal{P}^{(0)}$:

$$\phi = \mathcal{P}^{(t_N)} - \mathcal{P}^{(0)}. \quad (12)$$

The displacement ϕ encodes the offset information of $\mathcal{P}^{(t_N)}$. However, it is important to note that this does not exactly correspond to the displacement of pixel grid points, as the coordinates in $\mathcal{P}^{(t_N)}$ are not necessarily integers. Therefore, inspired by the surface splatting [87, 33] in computer graphics, we propose an efficient bilinear splatting to invert the trajectories and obtain accurate displacements at pixel grid locations (detailed in the supplementary material):

$$\{\Phi_z\}_{z=0}^{N-1} = \mathcal{I}_{\text{bs}}(\mathcal{P}^{(t_N)}, \phi), \quad (13)$$

where \mathcal{I}_{bs} means bilinear splatting and $\{\Phi_z\}_{z=0}^{N-1} \in \mathbb{R}^{N \times H \times W \times 2}$ represents the deformation fields corresponding to the image stack $\{I_z\}_{z=0}^{N-1}$. These deformation fields are subsequently used to register the image stack:

$$I_z = \Phi_z \circ I_z, \quad z = 0, \dots, N-1, \quad (14)$$

where \circ denotes the warping operation.

3.3 Loss functions

The trajectory tracking network is optimized using a composite unsupervised loss function comprising a normalized cross-correlation (NCC) data fidelity term and a diffusion regularizer applied to the displacement fields, with balancing coefficient λ . The loss function is formally expressed as:

$$\mathcal{L}_{\text{traj}} = \text{NCC}(I_z, I_{z-1} \circ u_{z-1,z}) + \lambda \|\nabla u_{z-1,z}\|_2^2, \quad (15)$$

where \circ indicates the spatial warping operator, and ∇ computes first-order spatial gradients via finite difference approximation.

According to Eq. (4), the Neural ODEs training objective is formulated as a composite unsupervised loss function comprising three components:

$$\mathcal{L} = \sum_{i=1}^M \left(\underbrace{\sum_{z=z_0}^{z_{N-1}} \|\mathcal{P}_i(z) - \mathcal{P}_i^{(0)}(z)\|^2}_{\text{Data fidelity term}} + \lambda \underbrace{\sum_{z=z_0}^{z_{N-1}} \|\nabla^2 \mathcal{P}_i(z)\|^2}_{\text{Trajectory smoothness term}} + \mu \underbrace{\sum_{z=z_0}^{z_{N-1}} \sum_{j \in \mathcal{N}(i)} \|\mathcal{P}_i(z) - \mathcal{P}_j(z)\|^2}_{\text{Spatial consistency term}} \right), \quad (16)$$

where $\mathcal{P}_i^{(0)}$ denotes the observed origin trajectory. The trajectory smoothness term, weighted by $\lambda > 0$, penalizes acceleration discontinuities through the temporal Laplacian operator ∇^2 , discretized with second-order differences. The spatial consistency term, weighted by $\mu > 0$, preserves local topological relationships within each slice.

4 Experiments

4.1 Datasets

We evaluate our algorithm on six publicly available datasets from the OpenOrganelle platform [77], covering a variety of mouse tissues such as heart, kidney, liver, skin, and pancreas. These allow for a comprehensive evaluation of NeuroTrajReg’s applicability and robustness across diverse tissue types. For real-world data, we utilized the female fruit fly brain neural dataset (FemFlyBrain) [63]. More details can be found in the supplementary material.

4.2 Implementation details

We implement our method using PyTorch, and all experiments are conducted on an NVIDIA A800 GPU with 80GB of memory. For the pixel trajectory tracking, we employ a 2D U-Net architecture to capture the displacements between adjacent slices. Neural ODEs employ a 3D U-Net architecture to parameterize f_θ . The network takes pixel trajectories \mathcal{P} as input and generates a time-dependent velocity field \mathcal{V} . More details can be found in the supplementary material.

4.3 Results

In Table 1, we compare NeuroTrajReg with advanced 3D registration techniques, including EMReg [42], EFSR [76], TrakEM2 [56], SEMLeSS [53], and GaussReg [82], using synthetic datasets spanning six diverse tissue types. Our approach, along with the supervised GaussReg method, demonstrates superior performance due to consideration of natural deformations, which other methods typically overlook. Methods overly focused on pixel-level similarity can neglect biologically relevant movements such as cellular dynamics, as shown in Figure 7. Moreover, compared to the supervised-manner GaussReg, our method uniquely benefits from global trajectory optimization, enabling the computation of a single interpolation deformation across slices. Conversely, GaussReg is constrained by a local receptive field, necessitating multiple interpolation steps. For the performance on real-world datasets, we selected several volumetric samples from the FemFlyBrain dataset [63]. The registration results are visualized in Figure 5, where we employed the interpolation and visualization tools provided by FIJI [59] to display the side views of the image stacks. Although GaussReg reduces nonlinear distortions, it also erroneously removes some structural textures, leading to a noticeable deviation from the original data. In contrast, our method not only effectively corrects the nonlinear distortions in the image stacks but also better preserves local structural details.

To further evaluate our approach, Figure 3 compares the error accumulation across approximately 1000 slices for SEMLeSS [53], GaussReg [82], and our method. SEMLeSS suffers significantly from error accumulation over long sequences. Although GaussReg performs reasonably well, it exhibits lower precision and higher variance compared to our method, indicating less stable performance. Our method achieves superior average SSIM accuracy and demonstrates substantially greater stability, maintaining consistent performance without noticeable degradation even after 1000 slices. Additional results are provided in the supplementary material.

Table 1: The performance of different 3D registration methods in six synthetic datasets [77].

		Mus Heart	Mus Kidney	Mus Liver-3	Mus Liver	Mus Pancreas	Mus Skin
EMReg [47]	MI	0.73 \pm 0.14	0.65 \pm 0.16	0.89 \pm 0.17	0.79 \pm 0.18	0.76 \pm 0.14	0.66 \pm 0.18
	SSIM	0.55 \pm 0.03	0.44 \pm 0.04	0.52 \pm 0.04	0.43 \pm 0.03	0.43 \pm 0.02	0.50 \pm 0.07
	NCC	0.90 \pm 0.03	0.81 \pm 0.05	0.86 \pm 0.05	0.78 \pm 0.05	0.81 \pm 0.03	0.87 \pm 0.04
EFSR [76]	MI	1.15 \pm 0.10	1.26 \pm 0.15	1.45 \pm 0.14	1.36 \pm 0.16	1.33 \pm 0.13	1.03 \pm 0.20
	SSIM	0.67 \pm 0.04	0.63 \pm 0.05	0.65 \pm 0.04	0.56 \pm 0.04	0.57 \pm 0.04	0.60 \pm 0.07
	NCC	0.98 \pm 0.00	0.97 \pm 0.01	0.97 \pm 0.00	0.95 \pm 0.01	0.95 \pm 0.01	0.97 \pm 0.01
TrakEM2 [56]	MI	1.01 \pm 0.03	1.24 \pm 0.10	1.48 \pm 0.07	1.47 \pm 0.08	1.48 \pm 0.06	1.08 \pm 0.15
	SSIM	0.57 \pm 0.02	0.61 \pm 0.03	0.66 \pm 0.03	0.61 \pm 0.03	0.65 \pm 0.03	0.64 \pm 0.06
	NCC	0.98 \pm 0.00	0.97 \pm 0.00	0.98 \pm 0.00	0.97 \pm 0.00	0.97 \pm 0.00	0.98 \pm 0.01
SEMLeSS [53]	MI	1.00 \pm 0.12	1.07 \pm 0.14	1.35 \pm 0.17	1.29 \pm 0.19	1.18 \pm 0.15	0.89 \pm 0.18
	SSIM	0.58 \pm 0.04	0.52 \pm 0.04	0.61 \pm 0.05	0.52 \pm 0.06	0.50 \pm 0.05	0.51 \pm 0.07
	NCC	0.97 \pm 0.01	0.96 \pm 0.01	0.97 \pm 0.01	0.95 \pm 0.01	0.95 \pm 0.01	0.96 \pm 0.01
GaussReg [82]	MI	1.43 \pm 0.05	1.54 \pm 0.11	1.83 \pm 0.07	1.83 \pm 0.08	1.76 \pm 0.06	1.40 \pm 0.16
	SSIM	0.83 \pm 0.02	0.78 \pm 0.03	0.82 \pm 0.02	0.80 \pm 0.02	0.80 \pm 0.02	0.82 \pm 0.03
	NCC	0.99 \pm 0.00	0.99 \pm 0.00	0.99 \pm 0.00	0.99 \pm 0.00	0.99 \pm 0.00	0.99 \pm 0.00
Ours	MI	1.47 \pm 0.08	1.67 \pm 0.14	1.88 \pm 0.11	1.85 \pm 0.21	1.81 \pm 0.12	1.52 \pm 0.17
	SSIM	0.87 \pm 0.02	0.85 \pm 0.02	0.87 \pm 0.01	0.84 \pm 0.15	0.86 \pm 0.18	0.86 \pm 0.03
	NCC	0.99 \pm 0.00	0.99 \pm 0.00	0.99 \pm 0.00	0.99 \pm 0.00	0.99 \pm 0.00	0.99 \pm 0.00

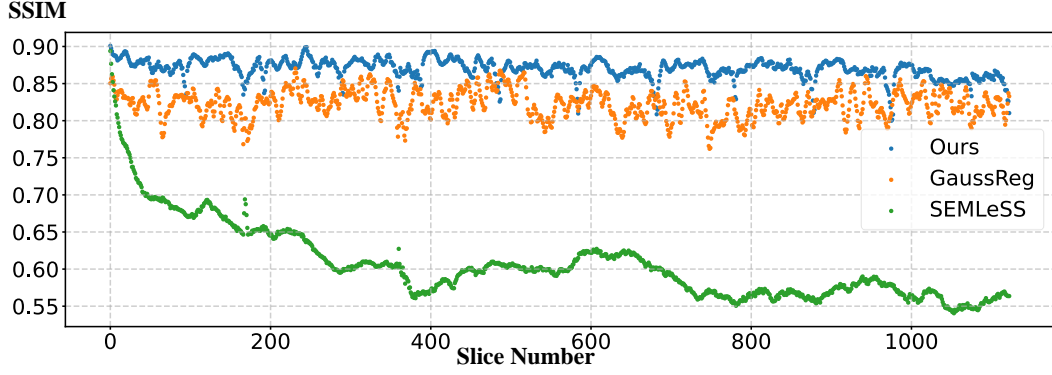


Figure 3: Error accumulation comparison on Mus Liver-3 dataset.

4.4 Ablation study

Loss Terms. We tested the impact of different components of the loss function (Eq. 16) on the results. The results are shown in Table 2, where TS, DF, and SC correspond to the Trajectory Smoothness term, Data Fidelity term, and Spatial Consistency term, respectively. Removing the Data Fidelity term decreases SSIM, as this leads to the trajectory deviating excessively from the original, with a risk of over-smoothing. Removing the Spatial Consistency term results in a decrease in SSIM, as failing to preserve local topology can lead to local pixel distortions. As for the TS term, it is key to trajectory smoothing, as it imposes constraints on the trajectory’s acceleration, preventing unreasonable instantaneous variations. In conclusion, all components are essential and work together to ensure optimal performance.

Hyperparameters. We tested the impact of different hyperparameters λ and μ in Eq. (16) on registration accuracy. λ controls the smoothness of the trajectory to filter unreasonable instantaneous velocity changes, while μ regulates the local consistency of the displacement field to ensure topological consistency before and after registration. λ has an important impact on registration performance; if it is too small, it fails to eliminate noise completely, while if it is too large, it may cause excessive deviation of the trajectory. In contrast, μ is more robust and has a lesser effect on registration performance.

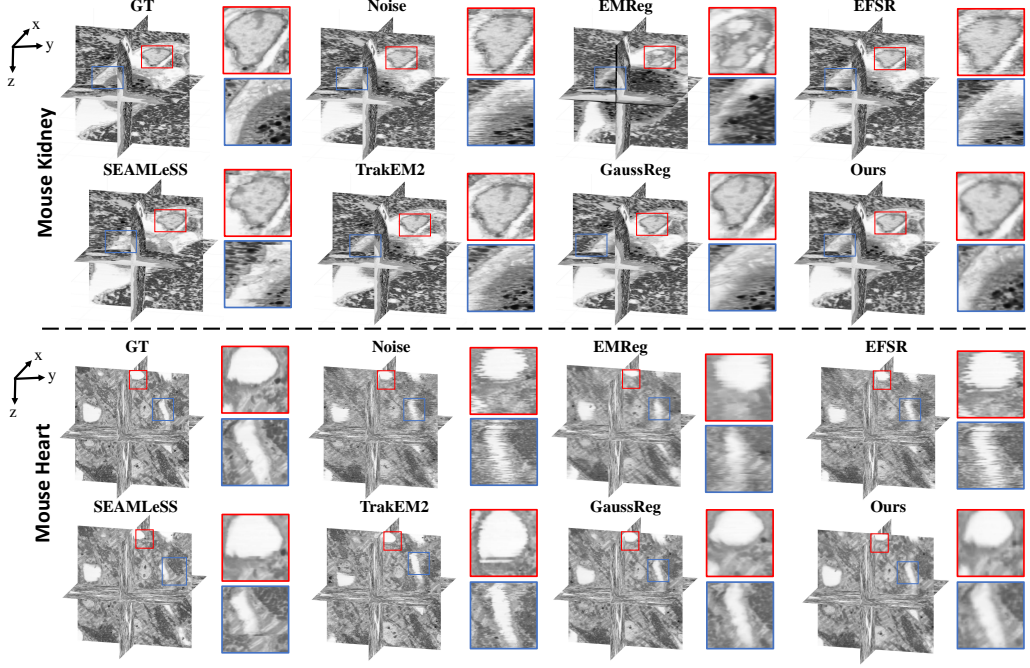


Figure 4: 3D visualization of registration results on the Mus Kidney and Mus Heart datasets.

Table 2: The ablation study of different loss terms.

	TS Loss term	DF Loss term	SC Loss term	SSIM
Mus Heart	✓	✗	✗	0.721
	✓	✓	✗	0.855
	✓	✗	✓	0.793
	✓	✓	✓	0.872
Mus Kidney	✓	✗	✗	0.698
	✓	✓	✗	0.822
	✓	✗	✓	0.774
	✓	✓	✓	0.851

Table 3: The ablation study of different hyper-parameter in the loss.

	SSIM	$\lambda \setminus \mu$	0.0001	0.001	0.01	0.1
Mus Heart		0.001	0.775	0.782	0.771	0.762
		0.01	0.795	0.806	0.783	0.779
		0.1	0.852	0.872	0.841	0.822
		1	0.811	0.819	0.805	0.787
Mus Kidney		0.001	0.754	0.762	0.752	0.741
		0.01	0.781	0.794	0.772	0.766
		0.1	0.833	0.851	0.821	0.815
		1	0.768	0.773	0.758	0.743

Table 4: Ablation study on the number of iterations in the Gauss-Seidel method.

SSIM/Iterations	50	100	150	200	250	300
Mus Heart	0.719	0.802	0.855	0.872	0.874	0.874
Mus Kidney	0.713	0.796	0.828	0.851	0.851	0.851
Mus Liver	0.695	0.774	0.811	0.842	0.844	0.844
Mus Skin	0.682	0.788	0.842	0.864	0.868	0.866

Number of Iterations. Table 4 investigates the effect of varying the number of iterations in the Gauss-Seidel method. From 0 to 200 iterations, the registration performance improves significantly and then gradually converges. Considering the trade-off between accuracy and efficiency, we select 200 iterations in practice to achieve a good balance.

Module Analysis. Our method can be decomposed into two main modules: the Gauss-Seidel method and Neural ODEs. The Gauss-Seidel method achieves fast convergence through iterative updates, while Neural ODEs effectively capture complex nonlinear distortions by modeling implicit dynamics. As shown in Table 5, we analyze the contribution of these key components in the trajectory smoothing scheme, and the results demonstrate the necessity and effectiveness of both modules.

Different Solver. Table 6 further studies the effect of different solvers for Neural ODEs. To avoid additional computational overhead, we adopt the Euler solver to balance performance and computational cost. More ablation experiments are provided in the supplementary material.

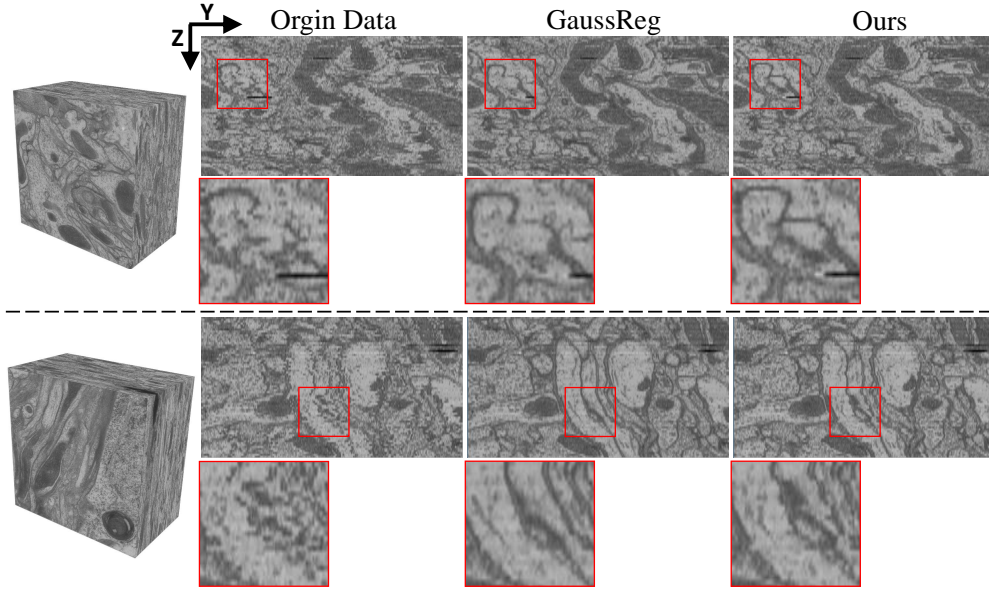


Figure 5: Side views of the original data, the registration results of GaussReg [82], and our method on two sampled volumes from the FemFlyBrain dataset [63].

Table 5: Ablation study of the components in our method.

	Gauss-Seidel	Neural ODEs	SSIM
Mus Heart	✓	✗	0.847
	✗	✓	0.835
	✓	✓	0.872
	✓	✗	0.833
Mus Kidney	✗	✓	0.827
	✓	✓	0.851

Table 6: Ablation study of different solvers for Neural ODEs.

	Euler		Dopri5		RK4	
	Time	SSIM	Time	SSIM	Time	SSIM
Mus Heart	2.45	0.872	8.33	0.877	7.68	0.876
Mus Kidney	2.41	0.851	8.52	0.855	7.66	0.854
Mus Liver	2.5	0.842	8.71	0.848	7.72	0.847
Mus Skin	2.46	0.864	8.39	0.869	7.25	0.868

5 Conclusion

In this paper, we present a novel 3D registration method for series section electron microscopy. We analyze the inherent characteristics of nonlinear distortion from the perspective of biological motion coherence, and reconsider 3D registration from the viewpoint of trajectory optimization. We introduce a dual optimization objective that balances global trajectory smoothness and local structural preservation, and we develop a solver that combines Gauss-Seidel iteration with Neural ODEs. Extensive experiments demonstrate that our method excels in structural restoration accuracy and cross-tissue robustness.

6 Acknowledgements

This research was supported by the National Key Research and Development Program of China [2021YFF0704300], Dubai Future Foundation (Award No. 2024CANAD-MES-061), and the Natural Science Foundation of Shandong Province [ZR2023YQ057].

References

- [1] Functional connectomics spanning multiple areas of mouse visual cortex. *Nature*, 640(8058):435–447, 2025.
- [2] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. An unsupervised learning model for deformable medical image registration. In *Proceedings of the IEEE conference on*

- computer vision and pattern recognition*, pages 9252–9260, 2018.
- [3] Mario Botsch, Alexander Hornung, Matthias Zwicker, and Leif Kobbelt. High-quality surface splatting on today’s gpus. In *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005.*, pages 17–141. IEEE, 2005.
 - [4] Brendan Celii, Stelios Papadopoulos, Zhuokun Ding, Paul G Fahey, Eric Wang, Christos Papadopoulos, Alexander B Kunin, Saumil Patel, J Alexander Bae, Agnes L Bodor, et al. Neurdf offers automated proofreading and feature extraction for connectomics. *Nature*, 640(8058):487–496, 2025.
 - [5] Jianchun Chen, Lingjing Wang, Xiang Li, and Yi Fang. Arbicon-net: Arbitrary continuous geometric transformation networks for image registration. *Advances in neural information processing systems*, 32, 2019.
 - [6] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
 - [7] Christopher I Connolly, J Brian Burns, and Rich Weiss. Path planning using laplace’s equation. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 2102–2106. IEEE, 1990.
 - [8] Adrian Dalca, Marianne Rakic, John Guttag, and Mert Sabuncu. Learning conditional deformable templates with convolutional networks. *Advances in neural information processing systems*, 32, 2019.
 - [9] Adrian V Dalca, Guha Balakrishnan, John Guttag, and Mert R Sabuncu. Unsupervised learning for fast probabilistic diffeomorphic registration. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, pages 729–738. Springer, 2018.
 - [10] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022.
 - [11] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023.
 - [12] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *The international journal of robotics research*, 29(5):485–501, 2010.
 - [13] Sven Dorkenwald, Casey M Schneider-Mizell, Derrick Brittain, Akhilesh Halageri, Chris Jordan, Nico Kemnitz, Manual A Castro, William Silversmith, Jeremy Maitin-Shephard, Jakob Troidl, et al. Cave: Connectome annotation versioning engine. *Nature Methods*, pages 1–9, 2025.
 - [14] Mohamed Elbanhawi, Milan Simic, and Reza N Jazar. Continuous path smoothing for car-like robots using b-spline curves. *Journal of Intelligent & Robotic Systems*, 80:23–56, 2015.
 - [15] Clare R Gamlin, Casey M Schneider-Mizell, Matthew Mallory, Leila Elabbady, Nathan Gouwens, Grace Williams, Alice Mukora, Rachel Dalley, Agnes L Bodor, Derrick Brittain, et al. Connectomics of predicted sst transcriptomic types in mouse visual cortex. *Nature*, 640(8058):497–505, 2025.
 - [16] Morteza Ghahremani, Mohammad Khateri, Bailiang Jian, Benedikt Wiestler, Ehsan Adeli, and Christian Wachinger. H-vit: A hierarchical vision transformer for deformable image registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11513–11523, 2024.
 - [17] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
 - [18] Gene H Golub et al. Cf vanloan, matrix computations. *The Johns Hopkins*, 113(10):23–36, 1996.
 - [19] Anjali Gour, Kevin M Boergens, Natalie Heike, Yunfeng Hua, Philip Laserstein, Kun Song, and Moritz Helmstaedter. Postnatal connectomic development of inhibition in mouse barrel cortex. *Science*, 371(6528):eabb4534, 2021.
 - [20] Grant Haskins, Uwe Kruger, and Pingkun Yan. Deep learning in medical image registration: a survey. *Machine Vision and Applications*, 31(1):8, 2020.
 - [21] Bintao He, Yan Zhang, Zhenbang Zhang, Yiran Cheng, Fa Zhang, Fei Sun, and Renmin Han. vemstitch: an algorithm for fully automatic image stitching of volume electron microscopy. *GigaScience*, 13:giae076, 2024.

- [22] Alessa Hering, Lasse Hansen, Tony CW Mok, Albert CS Chung, Hanna Siebert, Stephanie Häger, Annkristin Lange, Sven Kuckertz, Stefan Heldmann, Wei Shao, et al. Learn2reg: comprehensive multi-task medical image registration challenge, dataset and evaluation in the era of deep learning. *IEEE Transactions on Medical Imaging*, 42(3):697–712, 2022.
- [23] Jens Hilgert, Karina Hirsch, Torsten Bertram, and Manfred Hiller. Emergency path planning for autonomous vehicles using elastic band theory. In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, volume 2, pages 1390–1395. IEEE, 2003.
- [24] Bo Hu, Shenglong Zhou, Zhiwei Xiong, and Feng Wu. Recursive decomposition network for deformable image registration. *IEEE Journal of Biomedical and Health Informatics*, 26(10):5130–5141, 2022.
- [25] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.
- [26] Bernd Jähne. *Digital image processing*. Springer Science & Business Media, 2005.
- [27] Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems*, 32, 2019.
- [28] Xi Jia, Joseph Bartlett, Tianyang Zhang, Wenqi Lu, Zhaowen Qiu, and Jinming Duan. U-net vs transformer: Is u-net outdated in medical image registration? In *International Workshop on Machine Learning in Medical Imaging*, pages 151–160. Springer, 2022.
- [29] Brett E Johnson, Allison L Creason, Jayne M Stommel, Jamie M Keck, Swapnil Parmar, Courtney B Betts, Aurora Blucher, Christopher Boniface, Elmar Bucher, Erik Burlingame, et al. An omic and multidimensional spatial atlas from serial biopsies of an evolving metastatic breast cancer. *Cell Reports Medicine*, 3(2), 2022.
- [30] Christoph Jud, Nadia Möri, Benedikt Bitterli, and Philippe C Cattin. Bilateral regularization in reproducing kernel hilbert spaces for discontinuity preserving image registration. In *Machine Learning in Medical Imaging: 7th International Workshop, MLMI 2016, Held in Conjunction with MICCAI 2016, Athens, Greece, October 17, 2016, Proceedings 7*, pages 10–17. Springer, 2016.
- [31] Donggoo Jung, Daehyun Kim, and Tae Hyun Kim. Continuous exposure learning for low-light image enhancement using neural odes. In *The Thirteenth International Conference on Learning Representations*.
- [32] Narayanan Kasthuri, Kenneth Jeffrey Hayworth, Daniel Raimund Berger, Richard Lee Schalek, José Angel Conchello, Seymour Knowles-Barley, Dongil Lee, Amelio Vázquez-Reina, Verena Kaynig, Thouis Raymond Jones, et al. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015.
- [33] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- [34] Maher Khatib, Hazem Jaouni, Raja Chatila, and Jean-Paul Laumond. Dynamic path modification for car-like nonholonomic mobile robots. In *Proceedings of international conference on robotics and automation*, volume 4, pages 2920–2925. IEEE, 1997.
- [35] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33:6696–6707, 2020.
- [36] 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.
- [37] 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.
- [38] James R Kremer, David N Mastronarde, and J Richard McIntosh. Computer visualization of three-dimensional image data using imod. *Journal of structural biology*, 116(1):71–76, 1996.
- [39] Wei-Chung Allen Lee, Vincent Bonin, Michael Reed, Brett J Graham, Greg Hood, Katie Glattfelder, and R Clay Reid. Anatomy and function of an excitatory network in the visual cortex. *Nature*, 532(7599):370–374, 2016.
- [40] Xiaoyu Liu, Bo Hu, Mingxing Li, Wei Huang, Yueyi Zhang, and Zhiwei Xiong. A soma segmentation benchmark in full adult fly brain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7402–7411, 2023.

- [41] Xiaoyu Liu, Wei Huang, Zhiwei Xiong, Shenglong Zhou, Yueyi Zhang, Xuejin Chen, Zheng-Jun Zha, and Feng Wu. Learning cross-representation affinity consistency for sparsely supervised biomedical instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21107–21117, 2023.
- [42] Xinzhaoh Liu, Yueyi Zhang, Shenglong Zhou, Zhiwei Xiong, and Xiaoyan Sun. Electron microscopy image registration using correlation volume. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2023.
- [43] Sahil Loomba, Jakob Straehle, Vijayan Gangadharan, Natalie Heike, Abdelrahman Khalifa, Alessandro Motta, Niansheng Ju, Meike Sievers, Jens Gempt, Hanno S Meyer, et al. Connectomic comparison of mouse and human cortex. *Science*, 377(6602):eabo0924, 2022.
- [44] Aaron Lou, Derek Lim, Isay Katsman, Leo Huang, Qingxuan Jiang, Ser Nam Lim, and Christopher M De Sa. Neural manifold ordinary differential equations. *Advances in Neural Information Processing Systems*, 33:17548–17558, 2020.
- [45] Ao Luo, Fan Yang, Xin Li, Lang Nie, Chunyu Lin, Haoqiang Fan, and Shuaicheng Liu. GafLOW: Incorporating gaussian attention into optical flow. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9642–9651, 2023.
- [46] Jianqin Luo, Zhexiong Wan, Bo Li, Yuchao Dai, et al. Continuous parametric optical flow. *Advances in Neural Information Processing Systems*, 36:23520–23532, 2023.
- [47] Naisong Luo, Rui Sun, Yuwen Pan, Tianzhu Zhang, and Feng Wu. Electron microscopy images as set of fragments for mitochondrial segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3981–3989, 2024.
- [48] Mingyuan Meng, Dagan Feng, Lei Bi, and Jinman Kim. Correlation-aware coarse-to-fine mlps for deformable medical image registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9645–9654, 2024.
- [49] Ho Huu Nghia Nguyen, Tan Nguyen, Huyen Vo, Stanley Osher, and Thieu Vo. Improving neural ordinary differential equations with nesterov’s accelerated gradient method. *Advances in Neural Information Processing Systems*, 35:7712–7726, 2022.
- [50] Sunghyun Park, Kangyeol Kim, Junsoo Lee, Jaegul Choo, Joonseok Lee, Sookyoung Kim, and Edward Choi. Vid-ode: Continuous-time video generation with neural ordinary differential equation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2412–2422, 2021.
- [51] 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.
- [52] Nicolas Pielawski, Elisabeth Wetzer, Johan Öfverstedt, Jiahao Lu, Carolina Wählby, Joakim Lindblad, and Natasa Sladoje. Comir: Contrastive multimodal image representation for registration. *Advances in neural information processing systems*, 33:18433–18444, 2020.
- [53] Sergiy Popovych, Thomas Macrina, Nico Kemnitz, Manuel Castro, Barak Nehoran, Zhen Jia, J Alexander Bae, Eric Mitchell, Shang Mu, Eric T Trautman, et al. Petascale pipeline for precise alignment of images from serial section electron microscopy. *Nature Communications*, 15(1):289, 2024.
- [54] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.
- [55] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.
- [56] 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.
- [57] Michael Sander, Pierre Ablin, and Gabriel Peyré. Do residual neural networks discretize neural ordinary differential equations? *Advances in Neural Information Processing Systems*, 35:36520–36532, 2022.
- [58] Robin Sandkühler, Simon Andermatt, Grzegorz Bauman, Sylvia Nyilas, Christoph Jud, and Philippe C Cattin. Recurrent registration neural networks for deformable image registration. *Advances in Neural Information Processing Systems*, 32, 2019.

- [59] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–682, 2012.
- [60] Omveer Sharma, Nirod C Sahoo, and Niladri B Puhon. A survey on smooth path generation techniques for nonholonomic autonomous vehicle systems. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 5167–5172. IEEE, 2019.
- [61] Zhengyang Shen, François-Xavier Vialard, and Marc Niethammer. Region-specific diffeomorphic metric mapping. *Advances in Neural Information Processing Systems*, 32, 2019.
- [62] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1599–1610, 2023.
- [63] Shin-ya Takemura, Arjun Bharioke, Zhiyuan Lu, Aljoscha Nern, Shiv Vitaladevuni, Patricia K Rivlin, William T Katz, Donald J Olbris, Stephen M Plaza, Philip Winston, et al. A visual motion detection circuit suggested by drosophila connectomics. *Nature*, 500(7461):175–181, 2013.
- [64] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.
- [65] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, et al. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 8(6):3692–3711, 2023.
- [66] Philippe Thevenaz, Urs E Ruttimann, and Michael Unser. A pyramid approach to subpixel registration based on intensity. *IEEE transactions on image processing*, 7(1):27–41, 1998.
- [67] Jeya Maria Jose Valanarasu and Vishal M Patel. Unext: Mlp-based rapid medical image segmentation network. In *International conference on medical image computing and computer-assisted intervention*, pages 23–33. Springer, 2022.
- [68] Yogesh Verma, Markus Heinonen, and Vikas Garg. Climode: Climate and weather forecasting with physics-informed neural odes. In *The Twelfth International Conference on Learning Representations*.
- [69] Yogesh Verma, Markus Heinonen, and Vikas Garg. Climode: Climate and weather forecasting with physics-informed neural odes. *arXiv preprint arXiv:2404.10024*, 2024.
- [70] Valery Vishnevskiy, Tobias Gass, Gabor Szekely, Christine Tanner, and Orcun Goksel. Isotropic total variation regularization of displacements in parametric image registration. *IEEE transactions on medical imaging*, 36(2):385–395, 2016.
- [71] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, pages 36–54. Springer, 2024.
- [72] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE international conference on robotics and automation*, pages 987–993. IEEE, 2010.
- [73] Grady Barrett Wright. *Radial basis function interpolation: numerical and analytical developments*. University of Colorado at Boulder, 2003.
- [74] Yifan Wu, Tom Z Jiahao, Jiancong Wang, Paul A Yushkevich, M Ani Hsieh, and James C Gee. Nodeo: A neural ordinary differential equation based optimization framework for deformable image registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20804–20813, 2022.
- [75] Hedi Xia, Vai Suliafu, Hangjie Ji, Tan Nguyen, Andrea Bertozzi, Stanley Osher, and Bao Wang. Heavy ball neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 34:18646–18659, 2021.
- [76] 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.

- [77] C Shan Xu, Song Pang, Gleb Shtengel, Andreas Müller, Alex T Ritter, Huxley K Hoffman, Shin-ya Takemura, Zhiyuan Lu, H Amalia Pasolli, Nirmala Iyer, et al. An open-access volume electron microscopy atlas of whole cells and tissues. *Nature*, 599(7883):147–151, 2021.
- [78] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezaatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022.
- [79] Wenjing Yin, Derrick Brittain, Jay Borseth, Marie E Scott, Derric Williams, Jedediah Perkins, Christopher S Own, Matthew Murfitt, Russel M Torres, Daniel Kapner, et al. A petascale automated imaging pipeline for mapping neuronal circuits with high-throughput transmission electron microscopy. *Nature communications*, 11(1):4949, 2020.
- [80] Inwan Yoo, David GC 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: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pages 249–257. Springer, 2017.
- [81] Yajia Zhang, Hongyi Sun, Jinyun Zhou, Jiacheng Pan, Jiangtao Hu, and Jinghao Miao. Optimal vehicle path planning using quadratic optimization for baidu apollo open platform. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 978–984. IEEE, 2020.
- [82] Zhenbang Zhang, Hongjia Li, Zhiqiang Xu, Wenjia Meng, and Renmin Han. A gaussian filter-based 3d registration method for series section electron microscopy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1156–1164, 2025.
- [83] Shengyu Zhao, Yue Dong, Eric I Chang, Yan Xu, et al. Recursive cascaded networks for unsupervised medical image registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10600–10610, 2019.
- [84] Zhihao Zheng, J Scott Lauritzen, Eric Perlman, Camenzind G Robinson, Matthew Nichols, Daniel Milkie, Omar Torrens, John Price, Corey B Fisher, Nadiya Sharifi, et al. A complete electron microscopy volume of the brain of adult drosophila melanogaster. *Cell*, 174(3):730–743, 2018.
- [85] Shenglong Zhou, Bo Hu, Zhiwei Xiong, and Feng Wu. Self-distilled hierarchical network for unsupervised deformable image registration. *IEEE Transactions on Medical Imaging*, 2023.
- [86] Shenglong Zhou, Zhiwei Xiong, Chang Chen, Xuejin Chen, Dong Liu, Yueyi Zhang, Zheng-Jun Zha, and Feng Wu. Fast and accurate electron microscopy image registration with 3d convolution. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22*, pages 478–486. Springer, 2019.
- [87] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378, 2001.

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction accurately reflect the paper's contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the limitations of the work in the Appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not involve theoretical proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: This paper explicitly details the algorithm reproduction and dataset specifics in the appendix. In the future, we will release the code after the paper is accepted.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the code in the supplementary material and offer detailed instructions on how to reproduce the algorithm in the appendix.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide all the training and test details in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We present the error ranges of the evaluation metrics in the tables of both the main text and the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide the details of experiments compute resources in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss both potential positive societal impacts and negative societal impacts in the Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes. All external assets used in the paper, including code, datasets, and models, are properly credited. Their licenses and terms of use have been explicitly mentioned and strictly respected in accordance with their respective guidelines.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

A1 Dataset Details and Metrics

A1.1 Dataset Details

We selected six publicly available datasets from the OpenOrganelle platform [77] for simulation experiments. These datasets encompass high-resolution electron microscopy images of various mouse tissues, including the heart, kidney, liver, skin, and pancreas. The availability of ground-truth annotations in these datasets provides strong support for validating the applicability and robustness of our method across different types of biological tissues. For real-world data, we utilized the female fruit fly brain neural dataset (FemFlyBrain) [63]. These extensive datasets enabled us to validate the robustness of our method against real-world data.

P7 Mouse Heart. This dataset consists of heart tissue extracted from a wild-type C57BL/6J mouse at postnatal day 7. The experimental procedure followed IACUC guidelines for animal anesthesia. The tissue was fixed via perfusion with glutaraldehyde and sectioned using a vibratome. Subsequently, the sample underwent low-temperature reducing OTO staining, dehydration through a graded ethanol series, infiltration with Durcupan resin, and polymerization in an oven at 60°C.

Mouse Kidney. This dataset consists of kidney tissue extracted from an 8-week-old wild-type C57BL/6 mouse. The tissue was perfused with glutaraldehyde fixative and sectioned using a vibratome. After staining with reducing OTO at room temperature, the sample underwent dehydration, infiltration with a graded ethanol series, and Durcupan resin. Finally, the sample was polymerized in an oven at 60°C.

P7 Mouse Liver. This dataset comprises liver tissue collected from a wild-type, postnatal day 7 C57BL/6J mouse. Following anesthesia in accordance with IACUC guidelines, the sample was perfused with glutaraldehyde fixative and sectioned using a vibratome. After low-temperature reducing OTO staining, the tissue underwent dehydration, infiltration with graded ethanol and Durcupan resin, and was subsequently polymerized in a 60°C oven.

Mouse Liver. This dataset contains liver tissue from a wild-type, 8-week-old C57BL/6 mouse. The sample was fixed via perfusion with glutaraldehyde and sectioned using a vibratome. Subsequently, reducing OTO staining was performed at room temperature. The sample was then dehydrated, infiltrated with graded ethanol and Durcupan resin, and polymerized in a 60°C oven.

P7 Mouse Skin. This dataset includes skin tissue from a wild-type, postnatal day 7 C57BL/6J mouse. Mouse anesthesia was performed in accordance with IACUC guidelines, followed by perfusion fixation using glutaraldehyde and vibratome sectioning. The sample underwent low-temperature reducing OTO staining, dehydration, infiltration with graded ethanol and Durcupan resin, and was polymerized in a 60°C oven.

P7 Mouse Pancreas. This dataset consists of pancreas tissue from a wild-type, postnatal day 7 C57BL/6 mouse. The sample was fixed via perfusion with glutaraldehyde and sectioned using a vibratome. It underwent low-temperature reducing OTO staining, followed by dehydration, infiltration with graded ethanol and Durcupan resin, and polymerization in a 60°C oven.

Female Fruit Fly Brain Neural Dataset. FemFlyBrain [63] consists of the right hemisphere of a wild-type Oregon R female fruit fly brain. The brain was continuously sectioned at a thickness of 40 nanometers, covering regions including the medulla and downstream neuropils. The sections were imaged at a magnification of 35,000. The connectome within the medulla includes 379 neurons and 8,637 chemical synaptic contacts.

Full Adult Fly Brain Dataset. Full Adult Fly Brain Dataset (FAFB) [84] covers the entire depth of an adult fruit fly brain (approximately 250 μm). From the optimized sample, 7,062 consecutive sections of about 40 nm thickness were collected, optimized for high membrane contrast and fine ultrastructural preservation. A total of 7,050 sections (99.8%) were successfully imaged.

Mouse Cortical Dataset. The Mouse Cortical Dataset [32] provides a saturated reconstruction of a mouse neocortical sub-volume, in which all cellular elements (axons, dendrites, and glia) and various subcellular components (synapses, vesicles, spines, postsynaptic densities, and mitochondria) are fully annotated.

Table 7: Details of the synthetic datasets for testing phase.

Datasets	Shape	Numbers	URL
P7 Mouse Heart	1184×1184	1000	https://openorganelle.janelia.org/datasets/jrc_mus-heart-1
Mouse Kidney	1184×1184	1000	https://openorganelle.janelia.org/datasets/jrc_mus-kidney
P7 Mouse Liver	1184×1184	1127	https://openorganelle.janelia.org/datasets/jrc_mus-liver-3
Mouse Liver	1184×1184	558	https://openorganelle.janelia.org/datasets/jrc_mus-liver
P7 Mouse Pancreas	1184×1184	898	https://openorganelle.janelia.org/datasets/jrc_mus-pancreas-4
P7 Mouse Skin	1184×1184	1231	https://openorganelle.janelia.org/datasets/jrc_mus-skin-1
FemFlyBrain	1024×1024	1299	https://neurodata.io/data/takemura13/
FAFB	1024×1024	700	https://temca2data.org/
Mouse Cortex	1024×1024	700	https://neurodata.io/data/kasthuri15/

A1.2 Metrics

To quantitatively evaluate the performance of our method, we calculated several metrics, including Normalized Cross-Correlation (NCC), Mutual Information (MI), and Structural Similarity Index (SSIM) between the registered image series and the ground truth (GT). These metrics provide a comprehensive evaluation of the accuracy of our alignment and registration results from different perspectives. Specifically, NCC measures the similarity between the result images and the GT, MI reflects the mutual information between the two, and SSIM evaluates the accuracy based on structural information. Dice is a set similarity metric commonly used to measure the similarity between two samples. Here, we use the Dice score to evaluate the segmentation accuracy of 3D segmentation results on the registered data.

A2 Experimental Details

A2.1 Dataset Setup

For training the trajectory tracking module, we split the original image data into 3000 image pairs $\{I_0, I_1\}$, where I_0 and I_1 represent adjacent images. These images are cropped to a size of 1184×1184 . During training, the data is normalized to the range of $[-1, 1]$. The training-validation split is set to 0.95. Additionally, we randomly select one of the images from $\{I_0, I_1\}$ and apply random elastic deformation to simulate nonlinear distortions. Specifically, we first generate a random deformation field D_{rand} , which indicates the pixel displacement matrix. This matrix is then smoothed using a Gaussian filter, and the resulting deformation is applied to create the deformed images. The deformation process is described by the following formulas:

$$\begin{cases} \phi_i = \alpha \cdot \text{size}(I_i) \cdot \text{Gauss}(D_{\text{rand}}, \sigma \cdot \text{size}(I_i)), \\ I_i = \phi_i \circ I_i, \end{cases} \quad (17)$$

where ϕ_i represents the random deformation field applied to image I_i , \circ denotes the deformation operation, D_{rand} is the generated random displacement field, $\text{Gauss}(\cdot)$ is the 2D Gaussian filter operator, α controls the displacement magnitude, and σ determines the smoothing extent. In our experiments, we set $\sigma = 0.08$ and $\alpha = 1.0$.

During the testing phase, we cropped all six datasets into image stacks of size $1184 \times 1184 \times N$, where N denotes the number of slices. Additionally, we applied random elastic deformations to all images except the first one to simulate nonlinear distortions. The deformation parameters were set to $\sigma = 0.08$ and $\alpha = 1.0$. Detailed information regarding the datasets can be found in Table 7. Our test data consists of several hundreds to around 1000 samples, which fully demonstrates our method’s capability in handling long sequences and addressing the challenge of error accumulation.

A2.2 Network Architecture and Training Details

The trajectory tracking network adopts a 2D U-Net architecture with residual multi-kernel fusion, consisting of 1) a cascaded feature encoder with hybrid convolutional blocks, and 2) a dense feature decoder with transposed upsampling. The encoder progressively reduces spatial resolution through four strided convolutions (stride=2), doubling the number of channels from an initial 8 to 64. Specifically, the encoder is composed of four convolutional layers with output channels of 8, 16, 32, and 64, respectively. Each convolutional layer includes two convolution operations: the first convolution uses a 3×3 kernel, followed by a second convolution with a larger 7×7 kernel to expand the receptive field, similar to the approach in LKUnet [28]. The decoder consists of four layers and reconstructs dense predictions using four transposed convolutions (kernel size = 2, stride = 2), halving the number of channels from 64 to 2. Skip connections concatenate the encoder features with the corresponding decoder layers to facilitate hierarchical feature fusion. The final displacement field is generated through dual 3×3 convolutions with a Softsign activation. To enable spatial warping, a Spatial Transformer Network (STN) [25] is employed for image registration. The model is optimized using the ADAM optimizer with parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is set to 1×10^{-4} , and training is performed for 500 epochs with a batch size of 16. For the regularization term in the loss function, we set $\lambda = 1.5$ to encourage smoothness in displacement field.

Neural ODEs employ a 3D U-Net architecture to parameterize f_θ . The number of network layers and the number of channels per layer are kept consistent with the 2D U-Net used in the trajectory tracking module, but large convolutional kernels are avoided to reduce memory consumption. For training, we use the ADAM optimizer with parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is set to 1×10^{-4} , and training is conducted for 500 epochs with a batch size of 4. For the loss function, we set $\lambda = 0.1$ and $\mu = 0.005$. For the Gauss-Seidel method used in trajectory smoothing, we set the number of iterations to 200 and $\lambda = 0.1$. For Neural ODEs, we use the simple Euler solver, with all other parameters kept at their default settings.

A2.3 Training and Testing on Real Data

A key advantage of our method lies in its unsupervised learning paradigm, which is particularly valuable for real-world datasets that lack reliable ground-truth correspondences. The training process on real data is divided into two stages: training of the trajectory tracking module and training of the trajectory smoothing module. We begin by cropping a small spatiotemporal block of size $H \times W \times T$ from the real dataset. Training samples are then constructed following the procedure described in Section A2.1, except that no additional elastic deformation is applied. The trajectory tracking network is subsequently trained in an unsupervised manner, identical to the approach used for synthetic data. For training the trajectory smoothing module, we first apply the previously trained network to obtain estimated trajectories. These are then used for the unsupervised training of the smoothing module, again following the same procedure as used for synthetic data.

For testing on real data, the networks trained on cropped data blocks can be directly applied to the full image stack. We adopt a sliding-window registration strategy to perform registration across the entire long image sequence. Specifically, given the full real image stack $\{I_i\}_{i=0}^{N-1}$, we define a receptive field of moderate size (about 100 slices). Within each windowed region, the trajectory tracking network is used to estimate the displacement trajectories. These trajectories are then refined by the trajectory smoothing module. Subsequently, inverse warping is applied based on the smoothed trajectories to perform image registration. The registered image block is then placed back into its corresponding location in the full stack. The window is shifted along the sequence, and the procedure is repeated until the entire image stack has been registered.

A3 Supplementary Mathematical Derivations

A3.1 Background of the Gauss-Seidel Method

The Gauss-Seidel method [18], a classical iterative solver for linear systems $A\mathbf{x} = \mathbf{b}$, that leverages the most recent updates to accelerate convergence, particularly in sparse, tridiagonal systems. This system can be equivalently written as a symmetric tridiagonal linear system:

$$A\mathbf{p} = \mathbf{b}, \quad (18)$$

where: - $\mathbf{p} = [\mathcal{P}(1), \dots, \mathcal{P}(N-1)]^T$ is the unknown trajectory, - $\mathbf{b} = [\lambda\mathcal{P}^{(0)}(1), \dots, \lambda\mathcal{P}^{(0)}(N-1)]^T$, - and $A \in \mathbb{R}^{(N-1) \times (N-1)}$ is a tridiagonal matrix with structure:

$$A = \begin{bmatrix} 4+\lambda & -2 & & & \\ -2 & 4+\lambda & -2 & \ddots & \\ & \ddots & \ddots & \ddots & \\ & & -2 & 4+\lambda & -2 \\ & & & -2 & 4+\lambda \end{bmatrix}. \quad (19)$$

Applying the Gauss-Seidel method to this system, the general update rule for the i -th variable is:

$$x_i^{(k+1)} = \frac{1}{A_{ii}} \left(b_i - \sum_{j<i} A_{ij} x_j^{(k+1)} - \sum_{j>i} A_{ij} x_j^{(k)} \right). \quad (20)$$

For our tridiagonal matrix, each row z only has nonzero entries at $z-1$, z , and $z+1$. Substituting these into the general Gauss-Seidel formula, the update rule for $\mathcal{P}(z)$ becomes:

$$\mathcal{P}^{(k+1)}(z) = \frac{1}{A_{zz}} \left(\lambda\mathcal{P}^{(0)}(z) - A_{z,z-1}\mathcal{P}^{(k+1)}(z-1) - A_{z,z+1}\mathcal{P}^{(k)}(z+1) \right), \quad (21)$$

where: - $A_{zz} = 4 + \lambda$, - $A_{z,z-1} = A_{z,z+1} = -2$.

Substituting these in yields:

$$\mathcal{P}^{(k+1)}(z) = \frac{1}{4+\lambda} \left(2\mathcal{P}^{(k+1)}(z-1) + 2\mathcal{P}^{(k)}(z+1) + \lambda\mathcal{P}^{(0)}(z) \right). \quad (22)$$

For simplicity, we divide both numerator and denominator by 2, resulting in the final explicit Gauss-Seidel update:

$$\mathcal{P}^{(k+1)}(z) = \frac{\mathcal{P}^{(k+1)}(z-1) + \mathcal{P}^{(k)}(z+1) + \frac{\lambda}{2}\mathcal{P}^{(0)}(z)}{2+\lambda}, \quad z = 1, \dots, N-1. \quad (23)$$

A3.2 Derivation of the Gauss-Seidel Iteration Update Formula

For our trajectory smoothing method, the Gauss-Seidel iteration can be adapted by introducing a fidelity constraint. Below is the detailed derivation of the iteration formula. Specifically, to minimize the energy function:

$$E(u) = \lambda \sum_{i=0}^{N-1} (u_i - u_i^{(0)})^2 + \sum_{i=1}^{N-2} (u_{i-1} - 2u_i + u_{i+1})^2, \quad (24)$$

we take the partial derivative of E with respect to each u_i and set it to zero to obtain the optimality condition.

The fidelity term contributes a derivative of:

$$\frac{\partial}{\partial u_i} \left(\lambda(u_i - u_i^{(0)})^2 \right) = 2\lambda(u_i - u_i^{(0)}). \quad (25)$$

The smoothing term consists of a sum of squared second-order differences, where each point u_i appears in multiple overlapping terms, together with u_{i-1} and u_{i+1} . For clarity, we focus on a representative term where u_i is the center, to illustrate how the smoothness constraint acts locally on the trajectory:

$$\frac{\partial}{\partial u_i} (u_{i-1} - 2u_i + u_{i+1})^2 = -4(u_{i-1} - 2u_i + u_{i+1}). \quad (26)$$

Combining the derivatives of both terms, we obtain the total gradient:

$$\frac{\partial E}{\partial u_i} = 2\lambda(u_i - u_i^{(0)}) - 4(u_{i-1} - 2u_i + u_{i+1}) = 0. \quad (27)$$

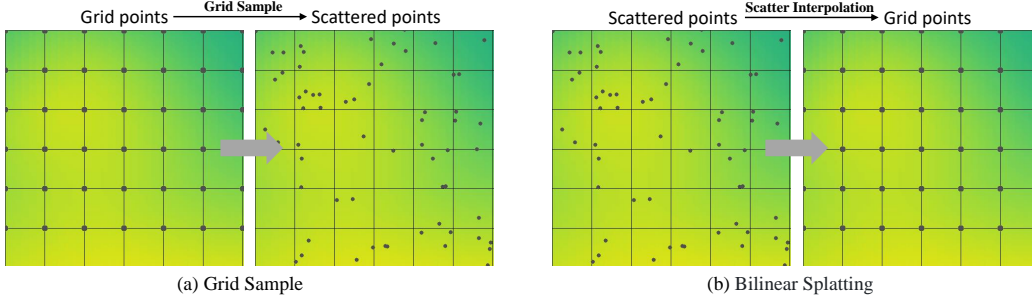


Figure 6: Regular grid sampling and Bilinear splatting. Grid sampling is used to obtain trajectory coordinates at sub-pixel resolution. For trajectory inversion, bilinear splatting is used to estimate the displacement at pixel grid locations.

Expanding and rearranging terms gives:

$$2\lambda(u_i - u_i^{(0)}) - 4u_{i-1} + 8u_i - 4u_{i+1} = 0. \quad (28)$$

Solving for u_i , we derive the following linear equation:

$$(2\lambda + 8)u_i = 2\lambda u_i^{(0)} + 4u_{i-1} + 4u_{i+1}. \quad (29)$$

This leads to the Gauss-Seidel iteration update formula:

$$u_i^{(k+1)} = \frac{1}{2\lambda + 8} \left(2\lambda u_i^{(0)} + 4u_{i-1}^{(k+1)} + 4u_{i+1}^{(k)} \right), \quad (30)$$

Simplifying further, we obtain

$$u_i^{(k+1)} = \frac{1}{\lambda + 2} \left(\lambda u_i^{(0)} + u_{i-1}^{(k+1)} + u_{i+1}^{(k)} \right), \quad (31)$$

where λ is halved from its original value. We use the updated $u_{i-1}^{(k+1)}$ from the current iteration and the old $u_{i+1}^{(k)}$ from the previous iteration to update $u_i^{(k+1)}$.

A4 Algorithmic Details of Bilinear Splatting

The goal of bilinear splatting is to map discrete feature points onto a regular grid using bilinear interpolation. Given a set of normalized coordinates and their associated feature vectors, we need to compute the grid values using the surrounding grid points. The coordinates are normalized to the range $[0, H)$ and $[0, W)$, where H and W are the height and width of the target grid. The feature vectors are associated with these coordinates and are projected to the grid using bilinear interpolation, as shown in Figure 6.

Let the coordinates of the discrete points be represented as $\text{coords} = \{(y_1, x_1), \dots, (y_N, x_N)\}$, where each (y_i, x_i) is a 2D coordinate within the normalized range. The corresponding feature vectors for each point are denoted by values $= \{v_1, \dots, v_N\}$, where $v_i \in \mathbb{R}^C$ is the feature vector for point i , and C is the feature dimension.

To perform bilinear splatting, we begin by identifying the four nearest grid points that surround each continuous coordinate (y_i, x_i) . These grid points form the corners of the smallest axis-aligned square in the discrete grid that encloses the given coordinate. Specifically, we obtain them by applying the floor and ceiling operations to both the y - and x -coordinates:

$$\begin{cases} y_{\text{floor}} = \lfloor y_i \rfloor, & x_{\text{floor}} = \lfloor x_i \rfloor, \\ y_{\text{ceil}} = \lfloor y_i \rfloor + 1, & x_{\text{ceil}} = \lfloor x_i \rfloor + 1. \end{cases} \quad (32)$$

These points form the basis for distributing the value at (y_i, x_i) across the surrounding pixels based on their bilinear interpolation weights.

Next, the weights for bilinear splatting are calculated based on the fractional parts of the coordinates. Let $\Delta y = y_i - \lfloor y_i \rfloor$ and $\Delta x = x_i - \lfloor x_i \rfloor$ represent the fractional parts of the coordinates. The four interpolation weights are given by:

$$\begin{cases} w_1 = (1 - \Delta x)(1 - \Delta y), & w_2 = (1 - \Delta x)\Delta y, \\ w_3 = \Delta x(1 - \Delta y), & w_4 = \Delta x\Delta y. \end{cases} \quad (33)$$

These weights represent the contribution of each neighboring grid point to the interpolated value at the target coordinate (y_i, x_i) . Now, we distribute the feature values to the corresponding grid locations based on these weights. For each coordinate (y_i, x_i) , we update the feature grid by adding the weighted value of the point to the grid locations corresponding to the four neighboring points. This is done using the following scatter operation:

$$\begin{cases} \text{Grid}[y_{\text{floor}}, x_{\text{floor}}] = \text{Grid}[y_{\text{floor}}, x_{\text{floor}}] + v_i \cdot w_1, \\ \text{Grid}[y_{\text{ceil}}, x_{\text{floor}}] = \text{Grid}[y_{\text{ceil}}, x_{\text{floor}}] + v_i \cdot w_2, \\ \text{Grid}[y_{\text{floor}}, x_{\text{ceil}}] = \text{Grid}[y_{\text{floor}}, x_{\text{ceil}}] + v_i \cdot w_3, \\ \text{Grid}[y_{\text{ceil}}, x_{\text{ceil}}] = \text{Grid}[y_{\text{ceil}}, x_{\text{ceil}}] + v_i \cdot w_4. \end{cases} \quad (34)$$

The grid values are accumulated at the corresponding grid positions, and the sum of the weights at each grid location is also accumulated for normalization purposes.

Finally, to ensure proper averaging when multiple points contribute to the same grid position, we normalize the resulting grid by the sum of the weights:

$$\text{Grid}[y, x] = \frac{\sum v_i \cdot w_i}{\sum w_i} \quad (35)$$

To avoid division by zero in case of no contribution to a grid location, a small epsilon $\epsilon = 1e - 8$ is added to the denominator during the normalization step. Thus, the bilinear splatting operation efficiently interpolates the feature values from discrete points to the target grid, with smooth interpolation properties ensured by the bilinear weights.

A5 Limitations and Future Work

Our approach can be divided into three main modules: 1) pixel trajectory tracking, 2) dynamics-based trajectory optimization, and 3) trajectory inversion and image registration. For each module, we conducted a simple exploration in the ablation study (see Sec.B.1) to evaluate its effect on registration performance, including the influence of complex scenarios such as high-noise artifacts and wrinkles that may affect trajectory-tracking accuracy. Future work may investigate 3D registration methods that are robust to artifacts and wrinkles, as well as explore the use of more advanced architectures for improving registration accuracy and generalizability. Furthermore, Table 11 presents the registration accuracy and execution time on the Mus Liver-3 dataset, where the test slice resolution is set to 1024×1024 . Our method achieves the highest SSIM, demonstrating superior accuracy compared to other approaches. However, due to the computational demands of the ODE solver, it does not achieve the fastest runtime.

To further investigate runtime efficiency, we conducted additional experiments focusing on two optimization strategies: (i) sparse trajectory sampling (every $2/3/4$ pixels); and (ii) resolution reduction by downsampling followed by upsampling. Table 9 shows that sparse sampling substantially reduces runtime (from 2.46s to 0.95s) while maintaining state-of-the-art accuracy, with SSIM only slightly decreasing from 0.874 to 0.831. Table 10 demonstrates that reducing the resolution also improves efficiency (from 2.46s to 0.91s), though at the cost of accuracy (SSIM drops from 0.874 to 0.827). Nevertheless, the performance remains superior to the GaussReg baseline. We attribute the observed degradation primarily to the use of a simple bilinear upsampling strategy. We argue that could achieve a more favorable balance between runtime and accuracy.

Table 8: Execution time and performance.

	TrakEM2 [56]	EFSR [76]	EMReg [42]	SEMLeSS [53]	GaussReg [82]	Ours
Time(s)	3.92	3.44	0.28	27.11	0.92	2.46
SSIM	0.66	0.52	0.65	0.61	0.82	0.87

Table 9: Ablation study of sparse sampling.

	All pixels	1/2 pixel	1/3 pixel	1/4 pixel
SSIM	0.874	0.866	0.842	0.831
Time(s)	2.46	1.35	1.12	0.95

Table 10: Ablation study of resolution reduction.

	Full resolution	1/2 resolution	1/4 resolution
SSIM	0.874	0.845	0.827
Time(s)	2.46	1.12	0.91

A6 Broader Impacts and Discussion

Our work focuses on 3D registration for Series Section Electron Microscopy (ssEM). One of the key advantages of our approach is its fully unsupervised training, which eliminates the reliance on ground truth and makes it applicable to real-world ssEM scenarios. For ssEM applications [79, 13, 15], this is particularly important for the increasingly complex and diverse electron microscopy image datasets [15, 77], as acquiring comprehensive ground truth across various tissue and cell types is challenging. Our method can be applied to the modeling and analysis of biological cell tissues, assisting researchers in exploring the structure and function of cellular tissues.

For other downstream tasks in ssEM, such as 3D segmentation [40, 41, 47], our method provides well-structured image data that aligns with natural biological morphology. By improving the axial continuity of the raw image data while preserving the structural integrity of superstructural details, our approach can enhance the performance of downstream tasks.

For the design of 3D registration algorithms, we propose a novel paradigm from the perspective of trajectory optimization. Our approach can be divided into three main components: 1) pixel trajectory tracking, 2) dynamics-based trajectory optimization, and 3) trajectory inversion and image registration. These three modules are decoupled and interchangeable. In theory, each module can be replaced with a more efficient algorithm. For example, more efficient feature point tracking algorithms [11, 10] or optical flow estimation algorithms [62, 78] can be used for trajectory tracking. Low-pass filtering [26] and convex quadratic programming (QP) [12, 17] can be used for trajectory smoothing. Radial basis function interpolation [73] and other splatting techniques [3] can be used for trajectory inversion. This helps researchers explore more efficient 3D registration methods based on this paradigm.

Due to time and resource constraints, we were unable to validate the performance of our method on larger-scale scenarios, particularly on ultra-high-resolution image data. The challenges posed by computational complexity in larger environments and the variability of data noise remain. In future work, we will refine our approach and adapt it into a microscopy image processing tool suitable for large-scale real-world datasets.

Table 11: Performance of different registration methods on downstream segmentation tasks across six datasets. We evaluate segmentation accuracy using the Dice score (Avg %).

	EMReg [42]	EFSR [76]	SEMLeSS [53]	TrakEM2 [56]	GaussReg [82]	Ours
Mus Heart	0.35 \pm 0.06	0.82 \pm 0.02	0.65 \pm 0.05	0.68 \pm 0.03	0.89 \pm 0.01	<u>0.88</u> \pm 0.01
Mus Kidney	0.41 \pm 0.07	0.75 \pm 0.02	0.67 \pm 0.02	0.75 \pm 0.02	<u>0.82</u> \pm 0.01	0.84 \pm 0.01
Mus Liver-3	0.41 \pm 0.08	0.90 \pm 0.02	0.86 \pm 0.02	0.90 \pm 0.02	0.95 \pm 0.01	<u>0.94</u> \pm 0.01
Mus Liver	0.57 \pm 0.09	0.89 \pm 0.04	0.86 \pm 0.05	0.92 \pm 0.04	0.94 \pm 0.03	0.94 \pm 0.03
Mus Pancreas	0.46 \pm 0.04	0.87 \pm 0.02	0.79 \pm 0.02	0.90 \pm 0.01	0.94 \pm 0.01	<u>0.93</u> \pm 0.01
Mus Skin	0.50 \pm 0.07	0.87 \pm 0.03	0.76 \pm 0.04	0.90 \pm 0.02	0.94 \pm 0.01	0.94 \pm 0.03

Table 12: Evaluation of the diffeomorphic property of deformation fields using the Folds metric (% of $|J\phi| \leq 0$).

% of $ J\phi \leq 0$	TrakEM2 [56]	EFSR [76]	EMReg [42]	SEMLeSS [53]	GaussReg [82]	Ours
Mus Heart	/	0.188	0.243	0.135	0.099	0.0171
Mus Kidney	/	0.206	0.379	0.24	0.174	0.0098
Mus Liver	/	0.193	0.31	0.155	0.106	0.0124
Mus Skin	/	0.171	0.252	0.102	0.091	0.0095

B Additional Quantitative Results and Visualization

B.1 More Experimental Results

Robustness to Error Accumulation. In practical applications, serial section electron microscopy (ssEM) datasets often contain hundreds or even thousands of images. This large number of slices poses a challenge for long-sequence registration, as cumulative errors can easily arise, eventually leading to substantial sequence drift and compromising the accurate reconstruction of the biological specimen’s true 3D structure. To systematically evaluate the ability of our method to suppress such cumulative errors, we conducted comparative experiments on six long-sequence datasets listed in Table 7, focusing on the robustness of our method versus GaussReg [82] and SEMLeSS [53].

Specifically, Figures 12 illustrate how the registration accuracy changes as the number of slices increases. The results show that our method consistently achieves higher average registration accuracy across the entire sequence, with smaller fluctuations in the accuracy curve. This indicates superior stability and robustness when handling long sequences. Moreover, Figures 7 and 8 present the 3D reconstruction results on the remaining four datasets. It is evident that our method accurately recovers the spatial structures of various biological tissues, further demonstrating its generalizability and robustness across different types of data.

Performance on 3D Segmentation Tasks. 3D segmentation [40, 41] is a important task in serial section electron microscopy (ssEM) and has been widely applied in various biological image analysis domains [19, 43]. High-quality 3D registration plays a crucial role in segmentation tasks, as it can significantly mitigate artifacts caused by structural deformation, scale variations, and differences in imaging modalities, thereby improving segmentation accuracy. To comprehensively evaluate the applicability of our method in 3D segmentation, we conducted experiments on six datasets involving various organelles, including nuclei and mitochondria. Specifically, we employed a segmentation network [67] to perform segmentation on the images registered by each method. The segmentation performance was quantified by computing the Dice similarity coefficient between the predicted results and the ground-truth (GT) label stacks.

Table 11 summarizes the Dice scores for 3D segmentation results across the six datasets. As shown, our method achieves the best performance on three datasets and the second-best on the remaining three, with overall results slightly below those of the supervised method GaussReg (with differences no greater than 0.01). This discrepancy may be attributed to the segmentation model’s sensitivity to subtle misalignments, while our method focuses on global trajectory optimization and may have limitations in handling local details. For instance, as illustrated in Figure 8 for the Mus Liver-3 dataset, minor local misalignments can still be observed in our registration results. Nevertheless, our method attains accuracy comparable to that of the supervised GaussReg. Furthermore, Figs. 9, 10, and 11 present 3D visualizations of the segmentation results on multiple datasets. These visual results

Table 13: Sensitivity analysis of the hyperparameter λ .

λ (SSIM)	0.15	1.5	4.5	7.5	10
Mus Heart	0.855	0.872	0.87	0.864	0.83
Mus Kidney	0.827	0.851	0.855	0.843	0.809

Table 14: Impact of different network architectures on trajectory tracking performance.

SSIM	RAFT [64]	GAFLOW [45]	SEA-RAFT [71]
Mus Heart	0.892	0.917	0.909
Mus Kidney	0.894	0.933	0.925
Mus Liver	0.871	0.928	0.918
Mus Skin	0.92	0.943	0.932

Table 15: Effect of large deformations on registration performance.

	GaussReg ($\alpha=1.0$)	Ours ($\alpha=1.0$)	GaussReg ($\alpha=1.5$)	Ours ($\alpha=1.5$)	GaussReg ($\alpha=2.0$)	Ours ($\alpha=2.0$)
Mus Heart	0.832	0.872	0.811	0.867	0.772	0.856
Mus Kidney	0.781	0.851	0.763	0.843	0.739	0.823
Mus Liver	0.805	0.842	0.782	0.836	0.758	0.817
Mus Skin	0.827	0.864	0.797	0.851	0.763	0.838

Table 16: Effect of noise levels on registration performance.

	GaussReg (5%)	Orgin (5%)	GaussReg (10%)	Orgin (10%)	GaussReg (15%)	Orgin (15%)
Mus Heart	0.815	0.851	0.788	0.832	0.748	0.771
Mus Kidney	0.762	0.833	0.734	0.822	0.715	0.759
Mus Liver	0.779	0.814	0.74	0.784	0.72	0.714
Mus Skin	0.804	0.837	0.751	0.803	0.728	0.742

demonstrate that our method can reliably reconstruct the correct 3D structure of biological tissues, leading to clearer and more consistent segmentation boundaries, further validating its practicality in downstream tasks.

In addition, we report the Folds metric in Table 12 to further evaluate the diffeomorphic property of the deformation fields. The results demonstrate that our method consistently achieves the lowest Folds values, remaining around 0.01, whereas most baseline methods report values greater than 0.1. This indicates that our approach substantially reduces grid folding in the generated deformation fields, thereby better preserving topological consistency compared to existing baselines.

Results on Real-World Data. To further evaluate the performance of our method on real-world datasets, we present the registration results on three datasets, FemFlyBrain, FAFB³, and the Mouse Cortical Dataset, as shown in Figures 13-17. The results demonstrate that our method achieves accurate and consistent alignment across diverse biological samples, effectively handling complex morphological variations and imaging artifacts.

More Ablation Experiments To gain a deeper understanding of our method, we conducted additional ablation studies. Table 13 presents a sensitivity analysis of the key hyperparameter λ , which primarily controls the smoothness term in the loss function of the trajectory tracking module. Table 14 investigates the impact of adopting more advanced network architectures on the performance of the trajectory tracking module. As shown, replacing the baseline with a more sophisticated network slightly improves the registration accuracy, demonstrating the modular flexibility of our framework and its compatibility with advanced architectures. Tables 15 and 16 further compare registration performance under more challenging conditions, including large deformations and high noise levels. The results indicate a gradual performance degradation as deformation or noise intensity increases, suggesting promising directions for extending our approach to more complex real-world scenarios.

³Due to FAFB data acquisition, only our registration results are shown.

B.2 Explainable Visualization Study

To better understand the performance and explainability of our new paradigm, we conducted a visualization analysis of the intermediate trajectories. Specifically, we performed trajectory tracking on data with nonlinear distortions, ground truth data, and registration results, ensuring consistency between the trajectories across different datas. The visualization results are shown in Figure 18. It is evident from the figure that the trajectories of the nonlinear distorted data exhibit irregular noise jitter, leading to abrupt changes in local curvature, which reflect the impact of distortion on the motion trajectory. In contrast, the trajectory of the ground truth data maintains the natural evolutionary pattern of the organism’s movement, exhibiting smooth and continuous changes. Our method successfully overcame abnormal physical deformations in the registration results, faithfully and precisely restoring the natural motion trajectory.

B.3 Failure Cases

We observe that our method is to some extent dependent on the accuracy of the trajectory tracking module. When the tracking is suboptimal, the subsequent image registration performance may be affected. This issue becomes particularly prominent when handling real-world anisotropic data, where the axial resolution is often significantly lower than the lateral (XY) resolution. Substantial structural and textural differences arise between adjacent slices (as shown in Figure 19). Moreover, real datasets often suffer from high noise levels, imaging artifacts, and missing slices, further complicating reliable trajectory estimation. In Figure 20, we illustrate several representative failure cases, in which the axial resolution of the image stacks is approximately one-tenth of the lateral resolution, with noticeable noise and missing slices. Although our method is still capable of performing registration under such challenging conditions, the visual quality may degrade due to the aforementioned interfering factors. We hope future research will explore more robust and efficient trajectory tracking modules, as well as 3D registration methods that are better suited for anisotropic and noisy real-world data.

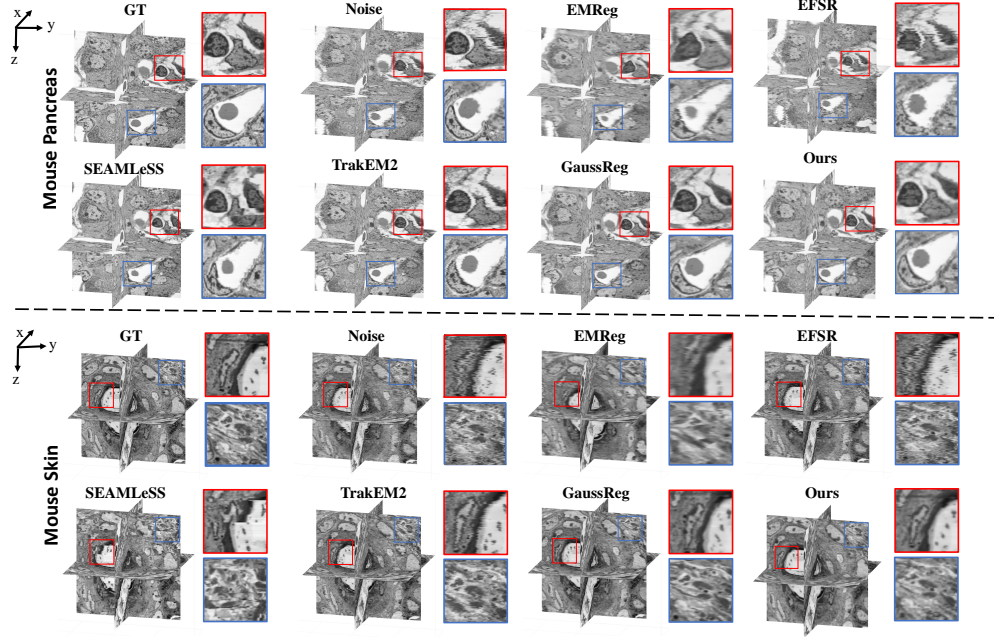


Figure 7: More 3D visualization of registration results on Mus Pancreas and Mus Skin datasets.

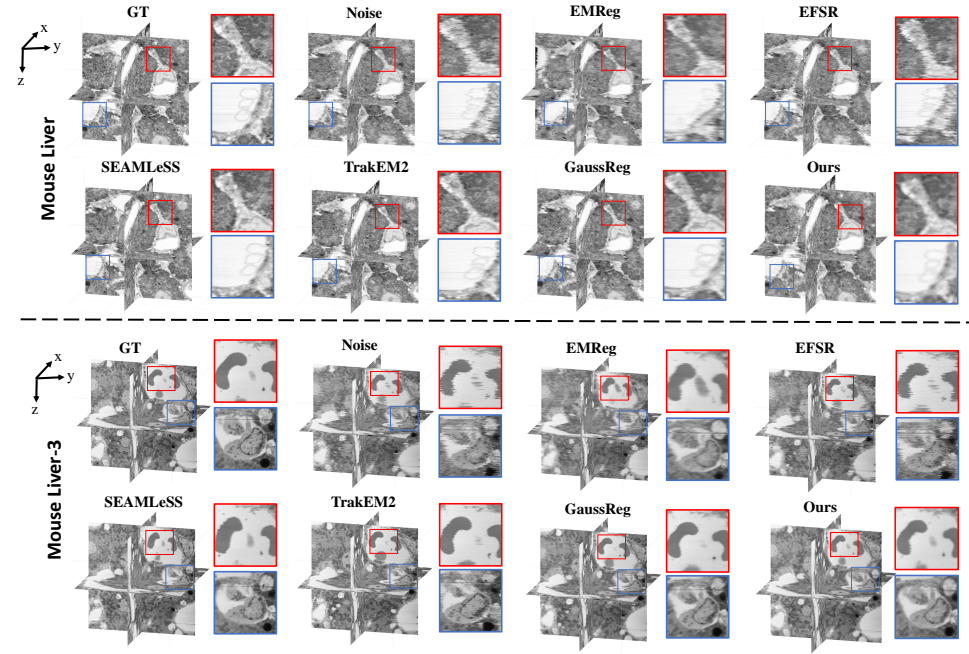


Figure 8: More 3D visualization of registration results on Mus Liver and Mus Liver-3 datasets.

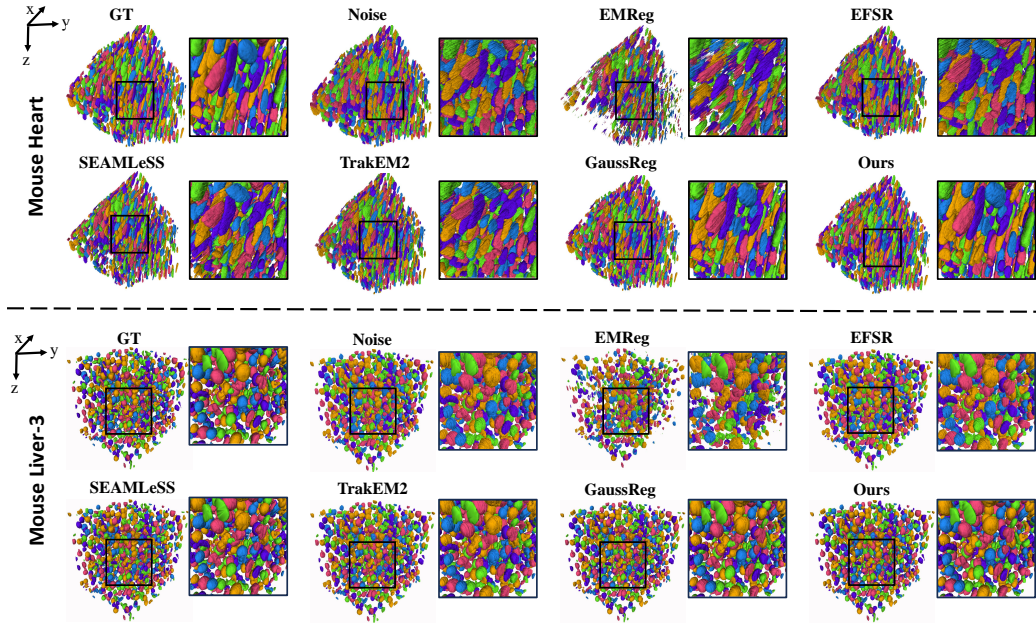


Figure 9: 3D segmentation visualizations of registration results using various methods on the Mus Heart and Mus Liver-3 datasets.

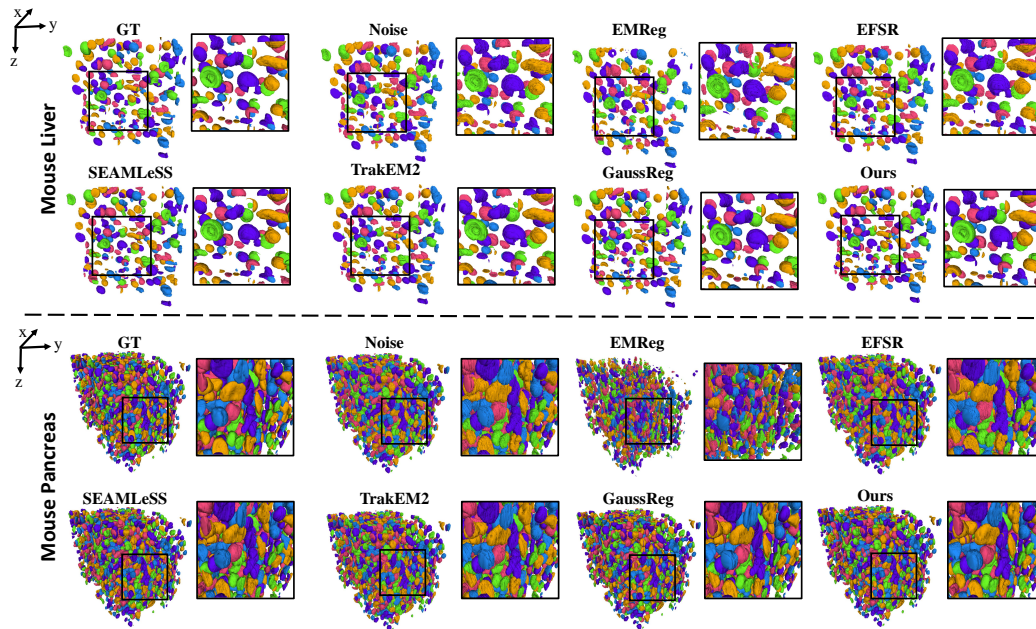


Figure 10: 3D segmentation visualizations of registration results using various methods on the Mus Liver and Mus Pancreas datasets.

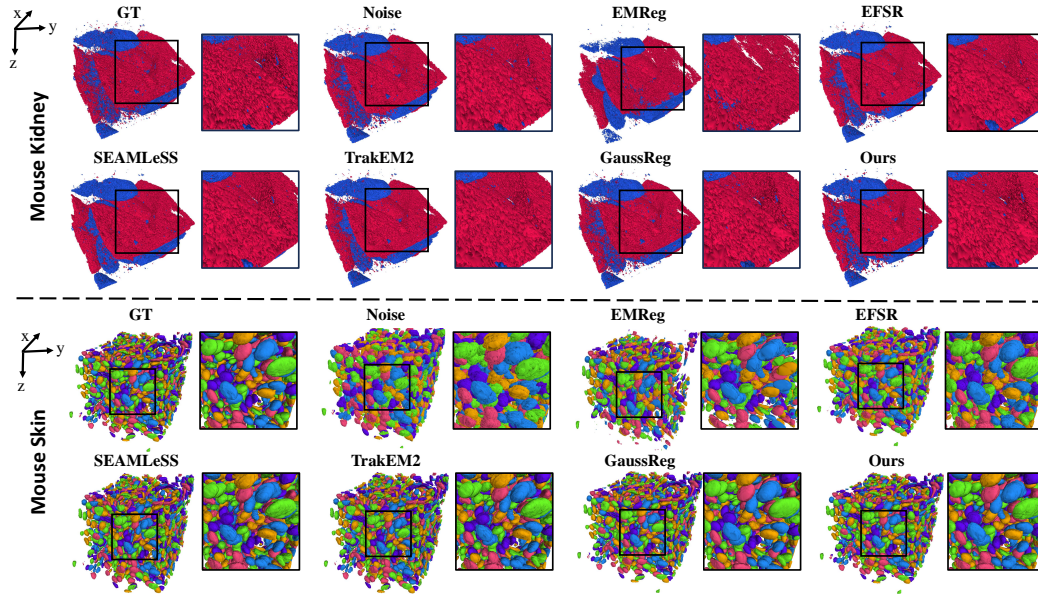


Figure 11: 3D segmentation visualizations of registration results using various methods on the Mus Kidney and Mus Skin datasets.

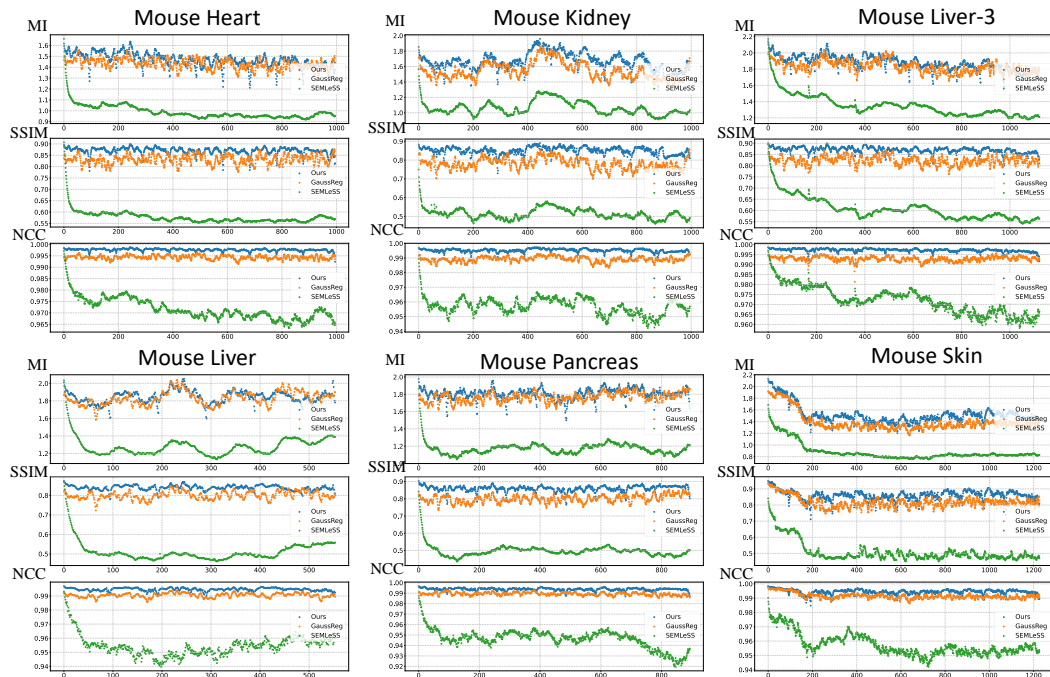


Figure 12: Error accumulation comparison between GaussReg [82] and SEMLeSS [53] and ours on six dataset.

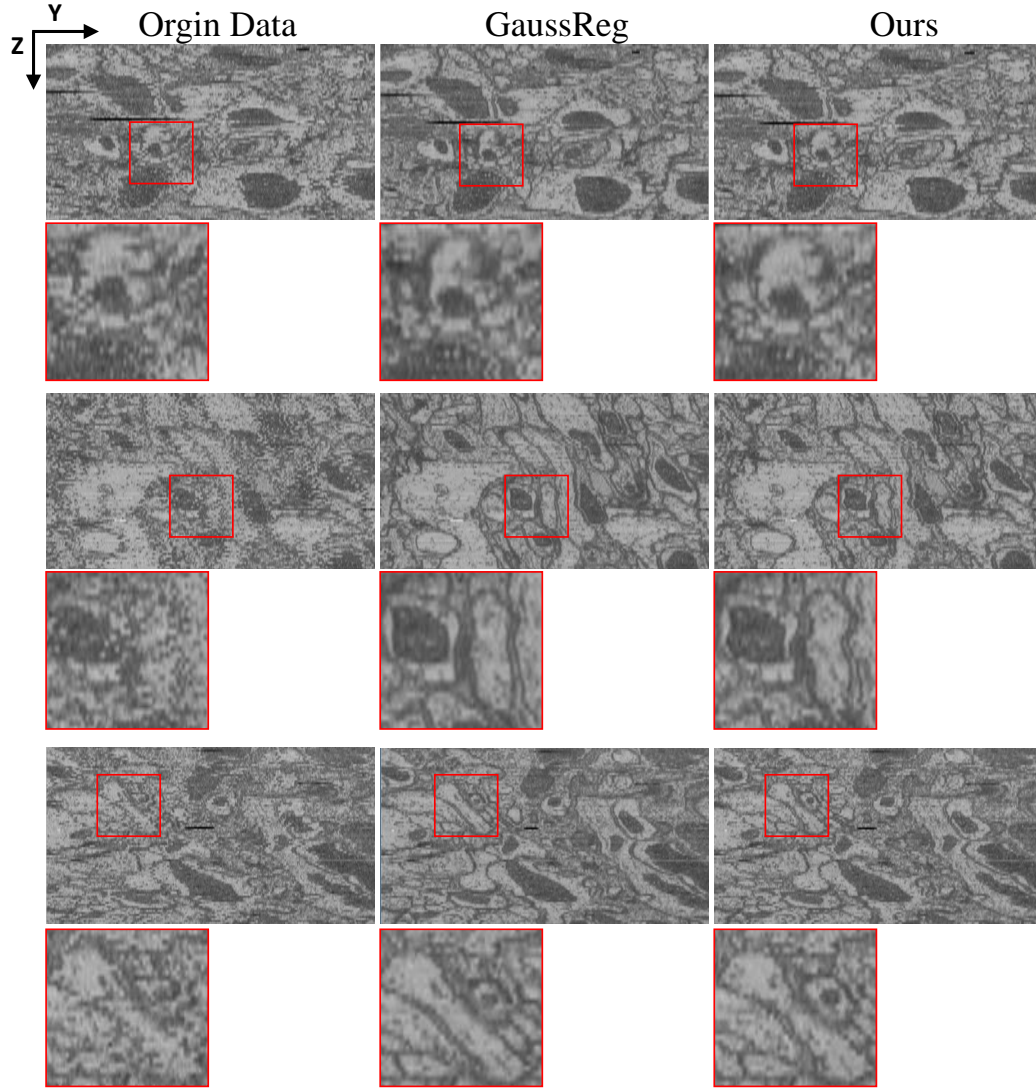


Figure 13: Side views of the original data, the registration results of GaussReg [82], and our method on the FemFlyBrain dataset [63].

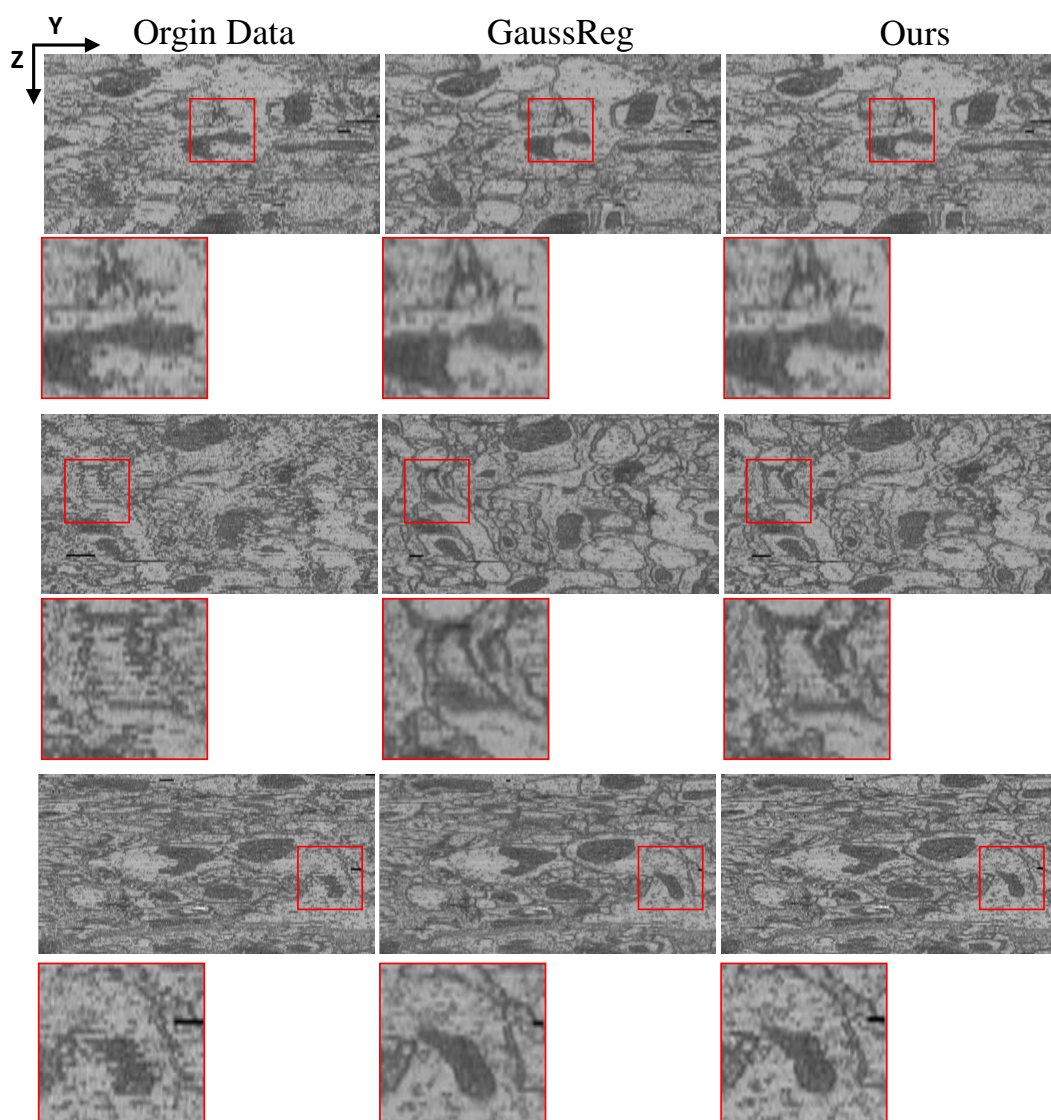


Figure 14: Side views of the original data, the registration results of GaussReg [82], and our method on the FemFlyBrain dataset [63].

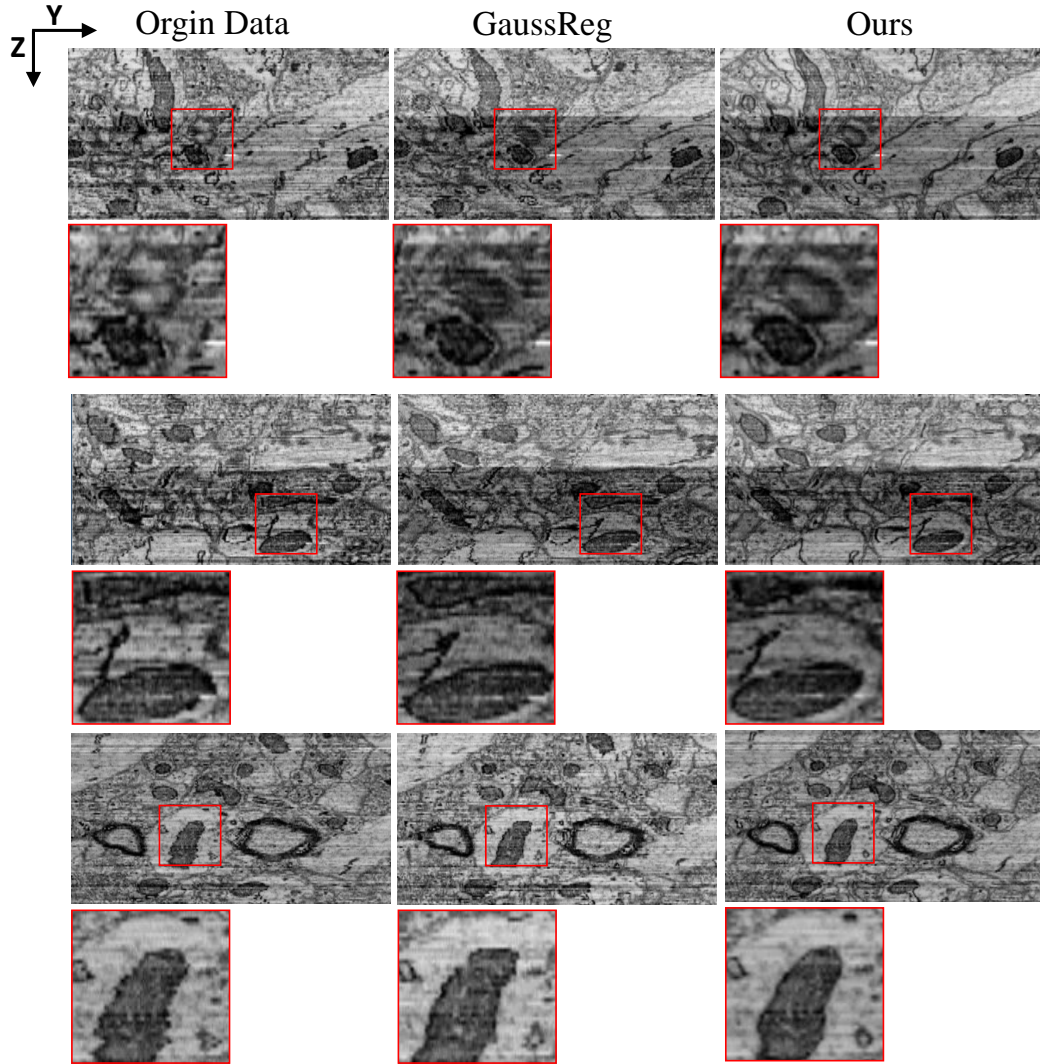


Figure 15: Side views of the original data, the registration results of GaussReg [82], and our method on the Mouse cortical dataset [32].

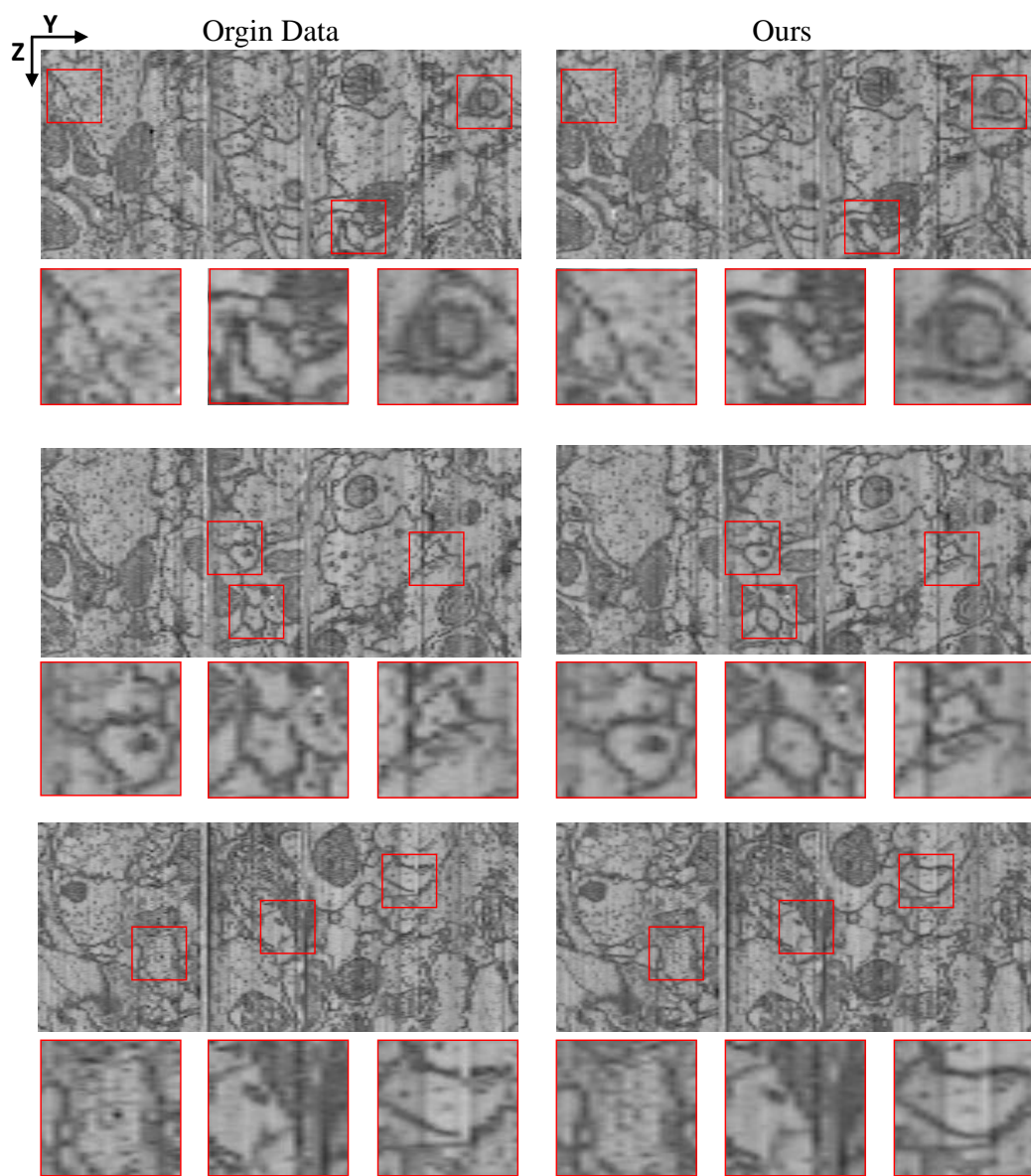


Figure 16: Side views of the original data, the registration results on the FAFB dataset [84].

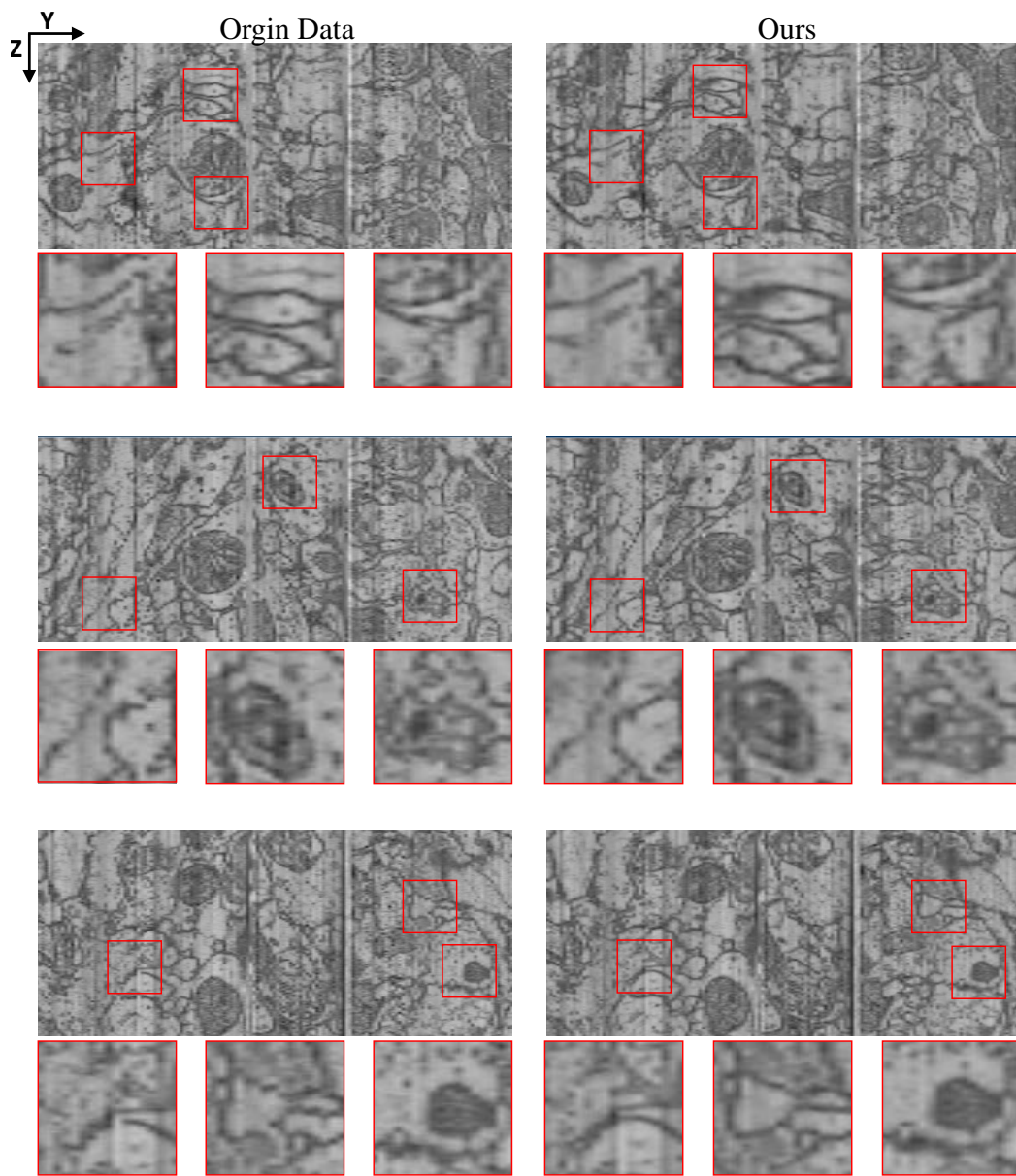


Figure 17: Side views of the original data and the registration results on the FAFB dataset [84].

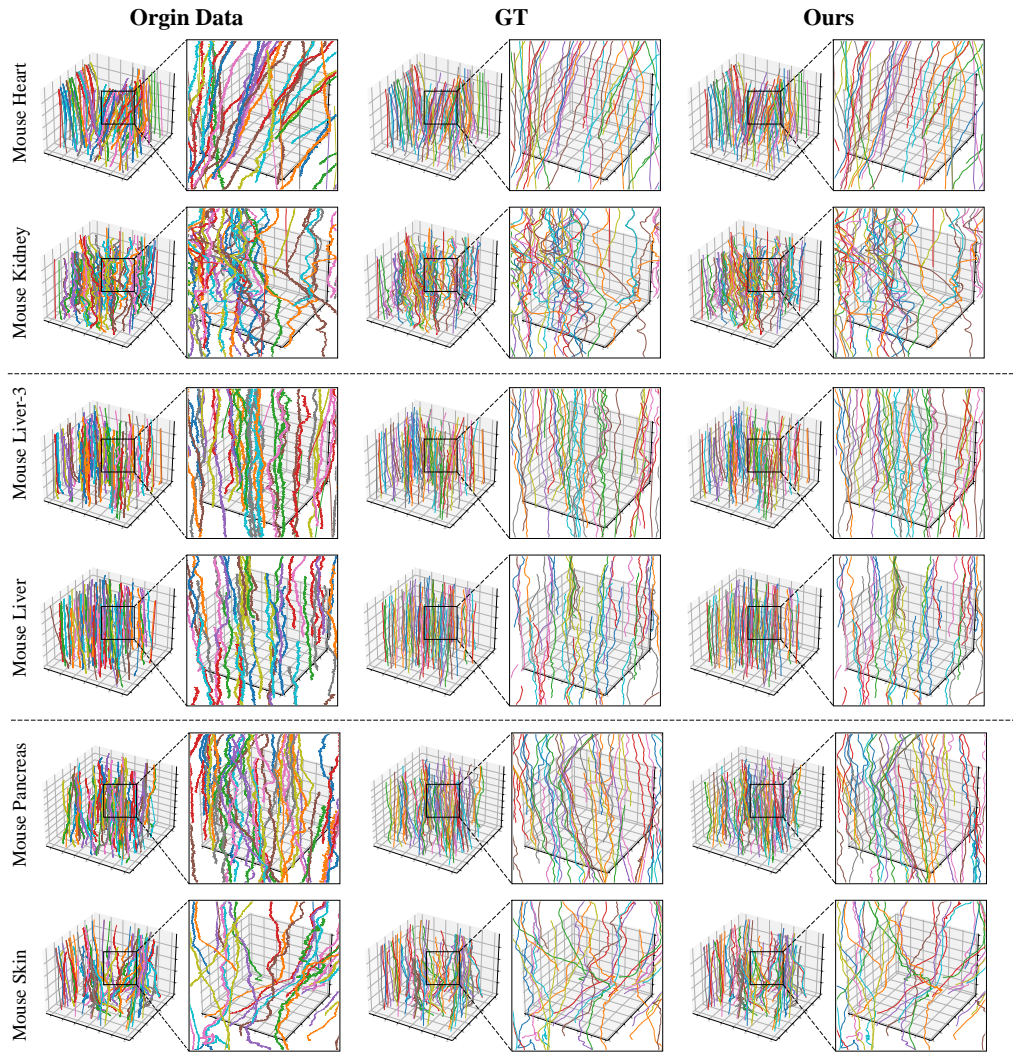


Figure 18: Trajectory visualization of the original data, ground truth, and registration results.

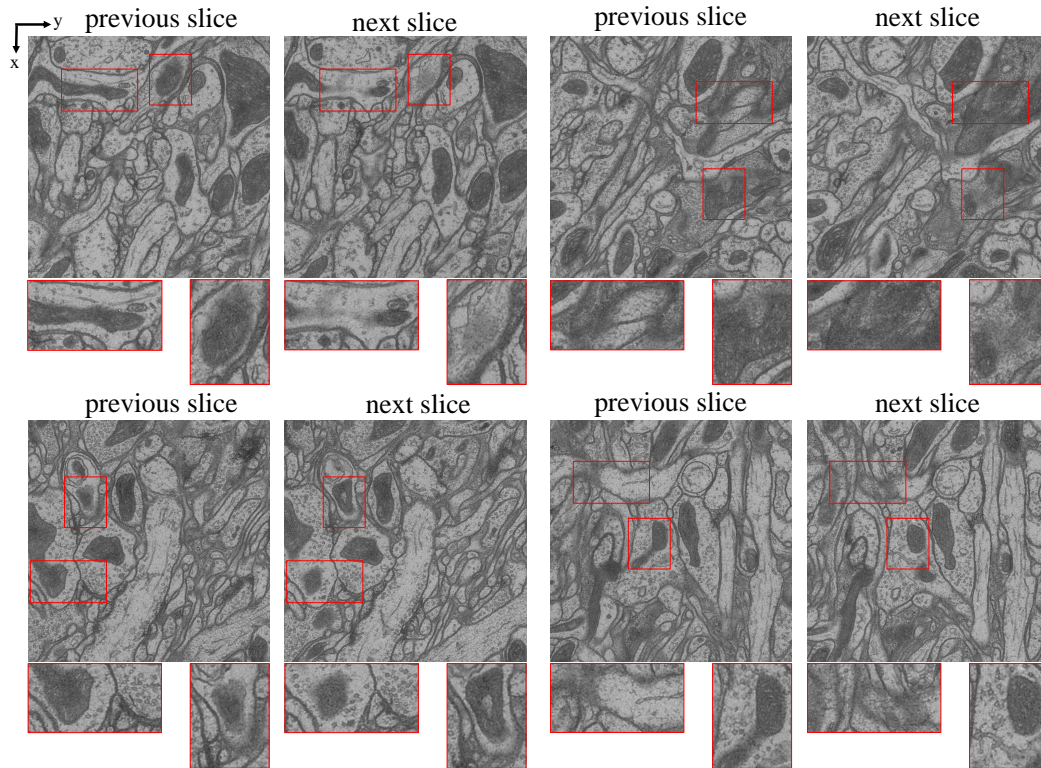


Figure 19: Four examples showing texture structure differences between adjacent slices of anisotropic data.

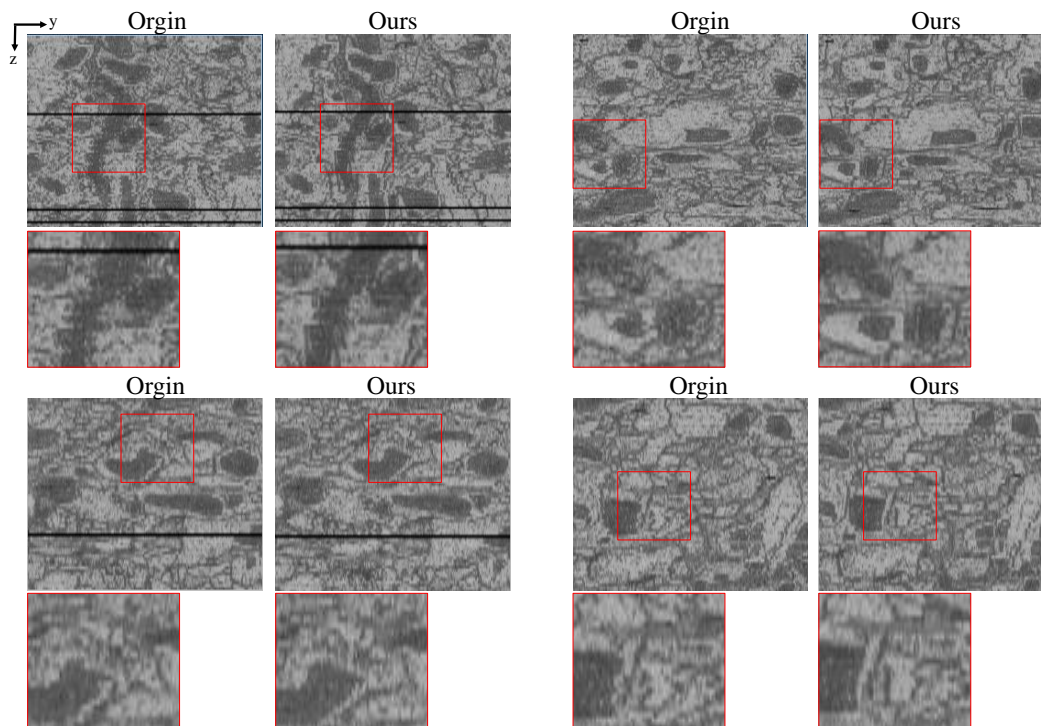


Figure 20: Four failure cases from the FemFlyBrain dataset. These examples are affected by low axial resolution, missing slices, and high noise.

References

- [1] Functional connectomics spanning multiple areas of mouse visual cortex. *Nature*, 640(8058):435–447, 2025.
- [2] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. An unsupervised learning model for deformable medical image registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9252–9260, 2018.
- [3] Mario Botsch, Alexander Hornung, Matthias Zwicker, and Leif Kobbelt. High-quality surface splatting on today’s gpus. In *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005.*, pages 17–141. IEEE, 2005.
- [4] Brendan Celii, Stelios Papadopoulos, Zhuokun Ding, Paul G Fahey, Eric Wang, Christos Papadopoulos, Alexander B Kunin, Saumil Patel, J Alexander Bae, Agnes L Bodor, et al. Neurd offers automated proofreading and feature extraction for connectomics. *Nature*, 640(8058):487–496, 2025.
- [5] Jianchun Chen, Lingjing Wang, Xiang Li, and Yi Fang. Arbicon-net: Arbitrary continuous geometric transformation networks for image registration. *Advances in neural information processing systems*, 32, 2019.
- [6] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- [7] Christopher I Connolly, J Brian Burns, and Rich Weiss. Path planning using laplace’s equation. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 2102–2106. IEEE, 1990.
- [8] Adrian Dalca, Marianne Rakic, John Guttag, and Mert Sabuncu. Learning conditional deformable templates with convolutional networks. *Advances in neural information processing systems*, 32, 2019.
- [9] Adrian V Dalca, Guha Balakrishnan, John Guttag, and Mert R Sabuncu. Unsupervised learning for fast probabilistic diffeomorphic registration. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, pages 729–738. Springer, 2018.
- [10] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022.
- [11] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023.
- [12] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *The international journal of robotics research*, 29(5):485–501, 2010.
- [13] Sven Dorkenwald, Casey M Schneider-Mizell, Derrick Brittain, Akhilesh Halageri, Chris Jordan, Nico Kemnitz, Manual A Castro, William Silversmith, Jeremy Maitin-Shephard, Jakob Troidl, et al. Cave: Connectome annotation versioning engine. *Nature Methods*, pages 1–9, 2025.
- [14] Mohamed Elbanhawi, Milan Simic, and Reza N Jazar. Continuous path smoothing for car-like robots using b-spline curves. *Journal of Intelligent & Robotic Systems*, 80:23–56, 2015.
- [15] Clare R Gamlin, Casey M Schneider-Mizell, Matthew Mallory, Leila Elabbady, Nathan Gouwens, Grace Williams, Alice Mukora, Rachel Dalley, Agnes L Bodor, Derrick Brittain, et al. Connectomics of predicted sst transcriptomic types in mouse visual cortex. *Nature*, 640(8058):497–505, 2025.
- [16] Morteza Ghahremani, Mohammad Khateri, Bailiang Jian, Benedikt Wiestler, Ehsan Adeli, and Christian Wachinger. H-vit: A hierarchical vision transformer for deformable image registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11513–11523, 2024.
- [17] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
- [18] Gene H Golub et al. Cf vanloan, matrix computations. *The Johns Hopkins*, 113(10):23–36, 1996.
- [19] Anjali Gour, Kevin M Boergens, Natalie Heike, Yunfeng Hua, Philip Laserstein, Kun Song, and Moritz Helmstaedter. Postnatal connectomic development of inhibition in mouse barrel cortex. *Science*, 371(6528):eabb4534, 2021.

- [20] Grant Haskins, Uwe Kruger, and Pingkun Yan. Deep learning in medical image registration: a survey. *Machine Vision and Applications*, 31(1):8, 2020.
- [21] Bintao He, Yan Zhang, Zhenbang Zhang, Yiran Cheng, Fa Zhang, Fei Sun, and Renmin Han. vemstitch: an algorithm for fully automatic image stitching of volume electron microscopy. *GigaScience*, 13:giae076, 2024.
- [22] Alessa Hering, Lasse Hansen, Tony CW Mok, Albert CS Chung, Hanna Siebert, Stephanie Häger, Annkristin Lange, Sven Kuckertz, Stefan Heldmann, Wei Shao, et al. Learn2reg: comprehensive multi-task medical image registration challenge, dataset and evaluation in the era of deep learning. *IEEE Transactions on Medical Imaging*, 42(3):697–712, 2022.
- [23] Jens Hilgert, Karina Hirsch, Torsten Bertram, and Manfred Hiller. Emergency path planning for autonomous vehicles using elastic band theory. In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, volume 2, pages 1390–1395. IEEE, 2003.
- [24] Bo Hu, Shenglong Zhou, Zhiwei Xiong, and Feng Wu. Recursive decomposition network for deformable image registration. *IEEE Journal of Biomedical and Health Informatics*, 26(10):5130–5141, 2022.
- [25] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.
- [26] Bernd Jähne. *Digital image processing*. Springer Science & Business Media, 2005.
- [27] Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems*, 32, 2019.
- [28] Xi Jia, Joseph Bartlett, Tianyang Zhang, Wenqi Lu, Zhaowen Qiu, and Jinming Duan. U-net vs transformer: Is u-net outdated in medical image registration? In *International Workshop on Machine Learning in Medical Imaging*, pages 151–160. Springer, 2022.
- [29] Brett E Johnson, Allison L Creason, Jayne M Stommel, Jamie M Keck, Swapnil Parmar, Courtney B Betts, Aurora Blucher, Christopher Boniface, Elmar Bucher, Erik Burlingame, et al. An omic and multidimensional spatial atlas from serial biopsies of an evolving metastatic breast cancer. *Cell Reports Medicine*, 3(2), 2022.
- [30] Christoph Jud, Nadia Möri, Benedikt Bitterli, and Philippe C Cattin. Bilateral regularization in reproducing kernel hilbert spaces for discontinuity preserving image registration. In *Machine Learning in Medical Imaging: 7th International Workshop, MLMI 2016, Held in Conjunction with MICCAI 2016, Athens, Greece, October 17, 2016, Proceedings 7*, pages 10–17. Springer, 2016.
- [31] Donggoo Jung, Daehyun Kim, and Tae Hyun Kim. Continuous exposure learning for low-light image enhancement using neural odes. In *The Thirteenth International Conference on Learning Representations*.
- [32] Narayanan Kasthuri, Kenneth Jeffrey Hayworth, Daniel Raimund Berger, Richard Lee Schalek, José Angel Conchello, Seymour Knowles-Barley, Dongil Lee, Amelio Vázquez-Reina, Verena Kaynig, Thouis Raymond Jones, et al. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015.
- [33] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- [34] Maher Khatib, Hazem Jaouni, Raja Chatila, and Jean-Paul Laumond. Dynamic path modification for car-like nonholonomic mobile robots. In *Proceedings of international conference on robotics and automation*, volume 4, pages 2920–2925. IEEE, 1997.
- [35] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33:6696–6707, 2020.
- [36] 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.
- [37] 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.
- [38] James R Kremer, David N Mastronarde, and J Richard McIntosh. Computer visualization of three-dimensional image data using imod. *Journal of structural biology*, 116(1):71–76, 1996.

- [39] Wei-Chung Allen Lee, Vincent Bonin, Michael Reed, Brett J Graham, Greg Hood, Katie Glattfelder, and R Clay Reid. Anatomy and function of an excitatory network in the visual cortex. *Nature*, 532(7599):370–374, 2016.
- [40] Xiaoyu Liu, Bo Hu, Mingxing Li, Wei Huang, Yueyi Zhang, and Zhiwei Xiong. A soma segmentation benchmark in full adult fly brain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7402–7411, 2023.
- [41] Xiaoyu Liu, Wei Huang, Zhiwei Xiong, Shenglong Zhou, Yueyi Zhang, Xuejin Chen, Zheng-Jun Zha, and Feng Wu. Learning cross-representation affinity consistency for sparsely supervised biomedical instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21107–21117, 2023.
- [42] Xinzhao Liu, Yueyi Zhang, Shenglong Zhou, Zhiwei Xiong, and Xiaoyan Sun. Electron microscopy image registration using correlation volume. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2023.
- [43] Sahil Loomba, Jakob Straehle, Vijayan Gangadharan, Natalie Heike, Abdelrahman Khalifa, Alessandro Motta, Niansheng Ju, Meike Sievers, Jens Gempt, Hanno S Meyer, et al. Connectomic comparison of mouse and human cortex. *Science*, 377(6602):eabo0924, 2022.
- [44] Aaron Lou, Derek Lim, Isay Katsman, Leo Huang, Qingxuan Jiang, Ser Nam Lim, and Christopher M De Sa. Neural manifold ordinary differential equations. *Advances in Neural Information Processing Systems*, 33:17548–17558, 2020.
- [45] Ao Luo, Fan Yang, Xin Li, Lang Nie, Chunyu Lin, Haoqiang Fan, and Shuaicheng Liu. GafLOW: Incorporating gaussian attention into optical flow. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9642–9651, 2023.
- [46] Jianqin Luo, Zhexiong Wan, Bo Li, Yuchao Dai, et al. Continuous parametric optical flow. *Advances in Neural Information Processing Systems*, 36:23520–23532, 2023.
- [47] Naisong Luo, Rui Sun, Yuwen Pan, Tianzhu Zhang, and Feng Wu. Electron microscopy images as set of fragments for mitochondrial segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3981–3989, 2024.
- [48] Mingyuan Meng, Dagan Feng, Lei Bi, and Jinman Kim. Correlation-aware coarse-to-fine mlps for deformable medical image registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9645–9654, 2024.
- [49] Ho Huu Nghia Nguyen, Tan Nguyen, Huyen Vo, Stanley Osher, and Thieu Vo. Improving neural ordinary differential equations with nesterov’s accelerated gradient method. *Advances in Neural Information Processing Systems*, 35:7712–7726, 2022.
- [50] Sunghyun Park, Kangyeol Kim, Junsoo Lee, Jaegul Choo, Joonseok Lee, Sookyung Kim, and Edward Choi. Vid-ode: Continuous-time video generation with neural ordinary differential equation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2412–2422, 2021.
- [51] 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.
- [52] Nicolas Pielawski, Elisabeth Wetzer, Johan Öfverstedt, Jiahao Lu, Carolina Wählby, Joakim Lindblad, and Natasa Sladoje. Comir: Contrastive multimodal image representation for registration. *Advances in neural information processing systems*, 33:18433–18444, 2020.
- [53] Sergiy Popovych, Thomas Macrina, Nico Kemnitz, Manuel Castro, Barak Nehoran, Zhen Jia, J Alexander Bae, Eric Mitchell, Shang Mu, Eric T Trautman, et al. Petascale pipeline for precise alignment of images from serial section electron microscopy. *Nature Communications*, 15(1):289, 2024.
- [54] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.
- [55] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.
- [56] 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.

- [57] Michael Sander, Pierre Ablin, and Gabriel Peyré. Do residual neural networks discretize neural ordinary differential equations? *Advances in Neural Information Processing Systems*, 35:36520–36532, 2022.
- [58] Robin Sandkühler, Simon Andermatt, Grzegorz Bauman, Sylvia Nyilas, Christoph Jud, and Philippe C Cattin. Recurrent registration neural networks for deformable image registration. *Advances in Neural Information Processing Systems*, 32, 2019.
- [59] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–682, 2012.
- [60] Omveer Sharma, Nirod C Sahoo, and Niladri B Puhon. A survey on smooth path generation techniques for nonholonomic autonomous vehicle systems. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 5167–5172. IEEE, 2019.
- [61] Zhengyang Shen, François-Xavier Vialard, and Marc Niethammer. Region-specific diffeomorphic metric mapping. *Advances in Neural Information Processing Systems*, 32, 2019.
- [62] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1599–1610, 2023.
- [63] Shin-ya Takemura, Arjun Bharioke, Zhiyuan Lu, Aljoscha Nern, Shiv Vitaladevuni, Patricia K Rivlin, William T Katz, Donald J Olbris, Stephen M Plaza, Philip Winston, et al. A visual motion detection circuit suggested by drosophila connectomics. *Nature*, 500(7461):175–181, 2013.
- [64] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.
- [65] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, et al. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 8(6):3692–3711, 2023.
- [66] Philippe Thevenaz, Urs E Ruttimann, and Michael Unser. A pyramid approach to subpixel registration based on intensity. *IEEE transactions on image processing*, 7(1):27–41, 1998.
- [67] Jeya Maria Jose Valanarasu and Vishal M Patel. Unext: Mlp-based rapid medical image segmentation network. In *International conference on medical image computing and computer-assisted intervention*, pages 23–33. Springer, 2022.
- [68] Yogesh Verma, Markus Heinonen, and Vikas Garg. Climode: Climate and weather forecasting with physics-informed neural odes. In *The Twelfth International Conference on Learning Representations*.
- [69] Yogesh Verma, Markus Heinonen, and Vikas Garg. Climode: Climate and weather forecasting with physics-informed neural odes. *arXiv preprint arXiv:2404.10024*, 2024.
- [70] Valery Vishnevskiy, Tobias Gass, Gabor Szekely, Christine Tanner, and Orcun Goksel. Isotropic total variation regularization of displacements in parametric image registration. *IEEE transactions on medical imaging*, 36(2):385–395, 2016.
- [71] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, pages 36–54. Springer, 2024.
- [72] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE international conference on robotics and automation*, pages 987–993. IEEE, 2010.
- [73] Grady Barrett Wright. *Radial basis function interpolation: numerical and analytical developments*. University of Colorado at Boulder, 2003.
- [74] Yifan Wu, Tom Z Jiahao, Jiancong Wang, Paul A Yushkevich, M Ani Hsieh, and James C Gee. Nodeo: A neural ordinary differential equation based optimization framework for deformable image registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20804–20813, 2022.
- [75] Hedi Xia, Vai Suliafu, Hangjie Ji, Tan Nguyen, Andrea Bertozzi, Stanley Osher, and Bao Wang. Heavy ball neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 34:18646–18659, 2021.

- [76] 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.
- [77] C Shan Xu, Song Pang, Gleb Shtengel, Andreas Müller, Alex T Ritter, Huxley K Hoffman, Shin-ya Takemura, Zhiyuan Lu, H Amalia Pasolli, Nirmala Iyer, et al. An open-access volume electron microscopy atlas of whole cells and tissues. *Nature*, 599(7883):147–151, 2021.
- [78] Hao-fei Xu, Jing Zhang, Jian-fei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022.
- [79] Wenjing Yin, Derrick Brittain, Jay Borseth, Marie E Scott, Derric Williams, Jedediah Perkins, Christopher S Own, Matthew Murfitt, Russel M Torres, Daniel Kapner, et al. A petascale automated imaging pipeline for mapping neuronal circuits with high-throughput transmission electron microscopy. *Nature communications*, 11(1):4949, 2020.
- [80] Inwan Yoo, David GC 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: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pages 249–257. Springer, 2017.
- [81] Yajia Zhang, Hongyi Sun, Jinyun Zhou, Jiacheng Pan, Jiangtao Hu, and Jinghao Miao. Optimal vehicle path planning using quadratic optimization for baidu apollo open platform. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 978–984. IEEE, 2020.
- [82] Zhenbang Zhang, Hongjia Li, Zhiqiang Xu, Wenjia Meng, and Renmin Han. A gaussian filter-based 3d registration method for series section electron microscopy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1156–1164, 2025.
- [83] Shengyu Zhao, Yue Dong, Eric I Chang, Yan Xu, et al. Recursive cascaded networks for unsupervised medical image registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10600–10610, 2019.
- [84] Zhihao Zheng, J Scott Lauritzen, Eric Perlman, Camenzind G Robinson, Matthew Nichols, Daniel Milkie, Omar Torrens, John Price, Corey B Fisher, Nadiya Sharifi, et al. A complete electron microscopy volume of the brain of adult drosophila melanogaster. *Cell*, 174(3):730–743, 2018.
- [85] Shenglong Zhou, Bo Hu, Zhiwei Xiong, and Feng Wu. Self-distilled hierarchical network for unsupervised deformable image registration. *IEEE Transactions on Medical Imaging*, 2023.
- [86] Shenglong Zhou, Zhiwei Xiong, Chang Chen, Xuejin Chen, Dong Liu, Yueyi Zhang, Zheng-Jun Zha, and Feng Wu. Fast and accurate electron microscopy image registration with 3d convolution. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22*, pages 478–486. Springer, 2019.
- [87] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378, 2001.