Sparse Graph Reconstruction and Seriation for Large-Scale Image Stacks

Fuming Yang
Harvard University
fumingyang@fas.harvard.edu

Yaron Meirovitch
Harvard University
yaron.mr@gmail.com

Jeff W. Lichtman
Harvard University
jeff@mcb.harvard.edu

Abstract

We study the problem of recovering 1D order from noisy, local pairwise comparisons matrix while minimizing oracle queries. We cast this as reconstructing a sparse, noisy line graph and to our knowledge give the first algorithm that provably builds a sparse graph containing all edges needed for exact seriation using only $O\big(N(\log N + K)\big)$ queries, near-linear in N for fixed window width K. The method is parallelizable and applies to both binary and bounded-noise distance oracles. Our five-stage pipeline combines: (i) a random-hook Borůvka phase that connects components via short-range edges in $O(N \log N)$ queries; (ii) iterative condensation to bound the graph diameter; (iii) a double-sweep BFS over the sparse graph to obtain a provisional global order; (iv) densification restricted to a K-window around this provisional sequence; and (v) a greedy SUPERCHAIN *that assembles the final order. Under a simple top-1 margin and bounded relative noise, we prove exact recovery and show that* SUPERCHAIN *succeeds even when only about $2N/3$ true adjacencies are present. We apply the approach to wafer-scale section ordering in serial section electron microscopy, which motivates the problem setting. Our method outperforms spectral, MST, and TSP baselines while requiring far fewer comparisons across diverse domains, and note that the algorithm is broadly applicable to sequencing tasks with local structure, including temporal snapshot ordering, archaeological seriation, playlist/tour construction, and other path-like data.*
Demo and code: YouTube | GitHub.

1. Motivation

A complete map of synaptic connections is essential for developing mechanistic models of perception, learning, and neuropsychiatric disease. Over the past decades, connectomics has progressed from the 302 neurons *C. elegans* map [21], to a fruit-fly brain containing more than 140 thousand neurons [5, 16], to a petavoxel-scale (1 mm^3) connectomes of the mouse visual cortex (MICrONS) [20] and human temporal cortex (H01) at nanometre resolution [17]. The next major milestone, a synapse-level connectome of an entire adult mouse brain (500 mm^3) is expected to generate exabytes of raw data.

A typical connectomic pipeline starts with cutting resin-embedded brain tissue into thin sections (usually 30 nm), which are collected on a substrate and then imaged with EM. Section collection can be automated by using tape-collection, such as with the Automated Tape-collecting Ultramicrotome (ATUM [10]), or by collecting sections directly on wafers (e.g. MagC [19], Gauss-EM [8]).

With the ATUM method, sections are collected in a sequence on a continuous tape, with the benefit that their order is preserved before EM begins. The tape is then cut into strips and mounted onto silicon wafers for imaging. While this method is robust, a single 100 mm silicon wafer will typically accommodate a relatively small number of sections (100-200 sections).

In contrast, tape-free methods can pack sections from hundreds to thousands per wafer. In MagC and Gauss-EM systems, make it more densely. the sections are floated in a water bath and then pooled together over a silicon wafer. The water is withdrawn, depositing the sections with high packing density on the wafer surface. These direct-to-wafer approaches have a number of advantages, including the above-mentioned high packing density, but also sections' flatness, reduced wrinkles owing to the annealing procedure, and the ability to collect very large sections, which make them more compatible with the requirements of imaging a whole-mouse brain with EM.

Such large collections of images can be achieved using a IBEAM-mSEM pipeline in which semi-thin sections (250-500 nm) are subjected to multiple cycles of SEM imaging and ion-beam milling. [8, 10, 19]. However, a key distinction of these tape-free methods compared to tape-collecting methods, is that the original cutting section order is not preserved. Therefore, section order needs to be computationally recovered later to allow for volumetric alignment of the EM images [10, 19]. This introduces several challenges due to the excessive number of large and highly variable sections.

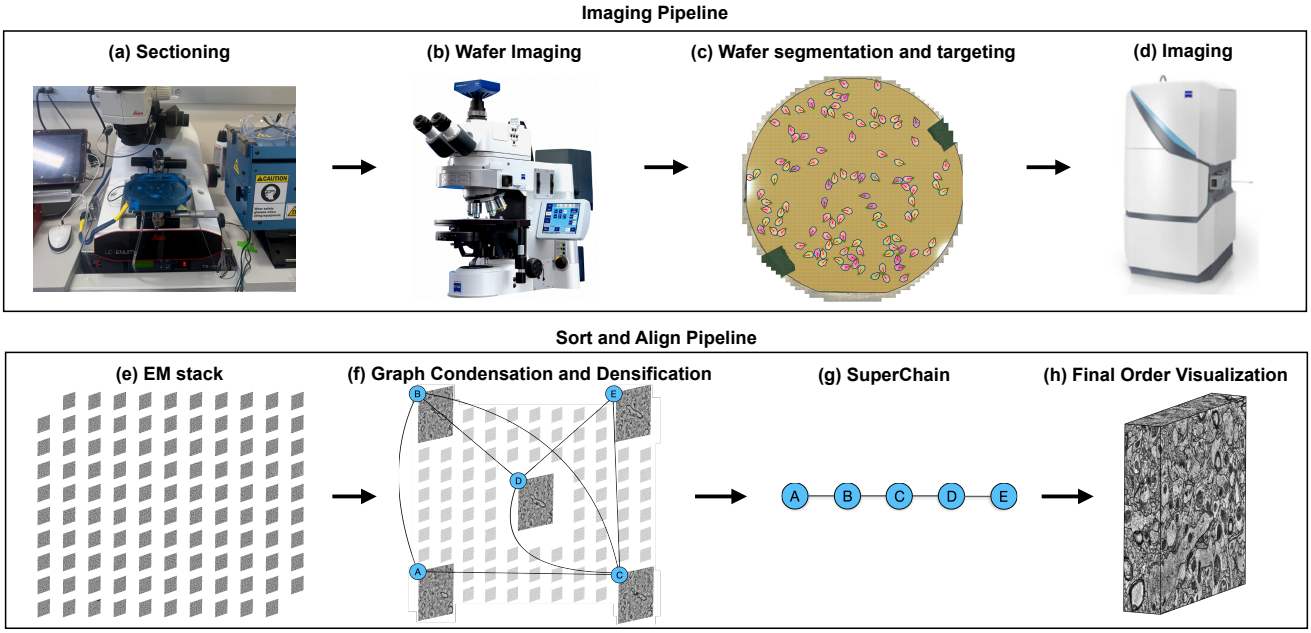

Figure 1. **WaferTools pipeline for wafer-to-volume EM sorting.** (a) Starting from unordered section collection method (e.g., MagC [19]), our WaferTools performs automatic section detection, (b–d) ROI unification with Procrustes alignment, and metadata generation for EM targeting. (e–g) The acquired EM stacks are then ordered via hybrid pipeline or near-linear Graph Condensation–Densification with SuperChain (hundreds to thousands sections). (f) Visualizing the final order result. Shown here is a wafer from the unpublished HI–MC dataset.

Current ordering methods fail to address three key challenges in EM section sorting: First, EM images usually yield a similarity pattern with a small set of strong matches limited to truly nearby sections, and a large noisy background elsewhere. As the distance between sections increases, pairwise scores based on intensity, SSIM, or SIFT inliers quickly drop to the noise level and fluctuate. In practice, since even nearby similarities are imperfect, incorporating noisy long-range comparisons does not resolve ambiguities but instead increases error rates [7, 9].

Second, classic spectral seriation assumes that similarity decays monotonically with serial distance; the Fiedler vector of the graph Laplacian should therefore vary smoothly along the true sequence. However, when the leading band is flat and many off-diagonal blocks are missing at random, the second eigenvector can split into two nearly equal plateaux, and sorting its components produces block inversions [2, 4, 7].

Third, comparing all image pairs is computationally expensive: identifying the few reliable short-distance edges would still require scoring a large fraction of the $N(N–1)/2$ pairs. Therefore, we seek a near-linear method that extracts the global order directly, avoiding near-exhaustive pair evaluation.

We address ordering under these constraints with a five-phase algorithm (Fig. 1f,g). In the first phase, a binary graph is constructed using a modified random-hook Borůvka procedure, which first connects the graph using $\Theta(N \log N)$ binary oracle calls (each oracle call queries a single pairwise similarity). In the second phase, iterative farthest-point condensation reduces the diameter of the dynamically updated graph, bringing it near the diameter of the true order graph in $\Theta(N \log N)$ steps. In the third phase, a double-sweep BFS over the sparse graph generates a provisional global seriation scaffold. In the fourth phase, we densify only inside a fixed window of $K = \mathcal{O}(c)$ sections around this provisional sequence, adding edges along its backbone. Theory shows it suffices to add $\Theta(NK)$ edges on that backbone for the recovery of missing true neighbors for each vertex even under considerable image noise (Sec. 5). In the final phase, a greedy routine dubbed SUPERCHAIN exploits the empirical gap between the strongest edge at each vertex and the next-best alternative to assemble the true order permutation in $\Theta(N \log N)$ time. The overall budget $\Theta\big(N(\log N + K)\big)$ is near-linear, reducing the one-million-section problem to a tractable overnight job on a single workstation.

In summary, in this work we formulate the EM seriation problem using concepts from random graph theory. To our knowledge, it introduces the first near-linear-query algorithm for the seriation problem. Experiments confirm zero ordering error and an eight-fold speedup over dense spectral methods. Finally, to aid reproducibility and further work in

this direction, we release a $500\,\text{sections}$ benchmark dataset with open-source code.

1.1. Related Work

We use *seriation* to denote the algorithmic problem of recovering a linear order from pairwise similarities (a sparse, partially observed graph in our case), and *ordering* to refer to the task and its output permutation.

Early algorithms for recovering section order leveraged image-similarity matrices (e.g., Pearson correlation or feature matches) together with manual curation; subsequent work formalized *post–acquisition* correction and jointly estimated section order and z–spacing directly from image-derived similarities [9].

Two common approaches employ Minimum Spanning Tree (MST) and Traveling Salesman Problem (TSP) formulations. MST-based methods build a minimum spanning tree on the similarity (or distance) graph (in polynomial time once the graph is known; see Kruskal [13], Prim [15]) and then derive a sequence from a traversal; they work when reliable short-range edges are present, but single-linkage–style chaining means a few wrong links can propagate errors [18]. TSP-style approaches seek a minimum-cost Hamiltonian path over sections; exact TSP is NP-hard [12], and practical solvers typically rely on dense costs or carefully curated candidate edges [1, 11, 14]. In our data, the informative signal is highly local: the nearest neighbor of a section is typically a true neighbor, the second true neighbor often appears among the next few best matches, and similarities for offsets ≥ 10 are at the noise level. Consequently, MST traversals and TSP heuristics either require scoring a large fraction of pairs to be robust, or they risk local swaps and occasional incorrect links.

In practice, spectral seriation and related convex relaxations (e.g., 2–SUM/SerialRank) provide more principled baselines [2, 7]. Spectral seriation sorts the components of the Fiedler vector of the graph Laplacian; however, when the leading similarity band flattens or the Laplacian admits a *multiple* Fiedler value, the solution becomes non-unique and can induce block inversions/"zig–zag" artifacts [2, 4, 7].

The seriation problem considered here appears in other domains, including archaeology and computational biology. Classical spectral methods [2] and modern convex/spectral formulations such as SerialRank [7] provide robustness to noise and missing entries under appropriate models, but many pipelines still rely on a large number of pairwise measurements, which are prohibitive at $10\,\text{mm}^3$ scale.

2. Methodology

Our approach to the section ordering problem consists of two main components: first, we construct a sparse graph that captures the essential connectivity structure while avoiding quadratic complexity, and second, we recover the exact ordering from this sparse graph. We begin by describing the overall pipeline before providing rigorous theoretical analysis.

2.1. Sparse Graph Construction

The fundamental challenge in section ordering is that exhaustive pairwise comparison of N sections requires $O(N^2)$ similarity computations, which becomes prohibitive for large-scale connectomics projects. We address this by constructing a sparse graph containing only $O(N(\log N + K))$ edges that provably includes all connections necessary for exact seriation, where K is a small constant densification window.

2.1.1. Feature Extraction and Similarity Computation

For each candidate section pair, we implement a multi-stage feature-matching pipeline to determine edge existence and weight. We first extract SIFT keypoints from both sections and establish putative correspondences via FLANN-based nearest neighbor matching. These matches are then filtered using RANSAC-based geometric consistency checks, where correspondences inconsistent with an affine transformation model are rejected as outliers. For pairs passing the minimum inlier threshold, we compute the structural similarity index (SSIM) on the overlapping region after alignment, capturing fine-scale morphological consistency beyond keypoint matching. The final similarity score combines both metrics:

$$M_{uv} = \text{SSIM}(S_u, S_v) \cdot \#\text{Inliers}(S_u, S_v) \tag{1}$$

This multi-modal scoring ensures robust edge detection despite imaging artifacts and registration errors.

2.1.2. Four-Phase Graph Construction Algorithm

Our sparse graph construction proceeds through four carefully designed phases that balance computational efficiency with theoretical guarantees. The key insight is that we can build a graph containing all essential edges without examining most of the $N(N-1)/2$ possible pairs.

Phase 1: Initial Connectivity via Random-Hook Borůvka. We begin by establishing global connectivity through a randomized variant of Borůvka's algorithm. Unlike classical MST algorithms that assume access to all edges, our setting requires discovering which edges exist through oracle queries. Each connected component samples $O(\log N)$ random candidate partners per round and queries the oracle for valid short-range connections. When a valid edge is found, the components are marked for merging. This process continues until a single spanning tree is formed, requiring $O(N \log N)$ queries with high probability under the assumption that short edges between components exist and can be found through random sampling. Under the assumptions in Sec. 5 with a constant K, Phase A

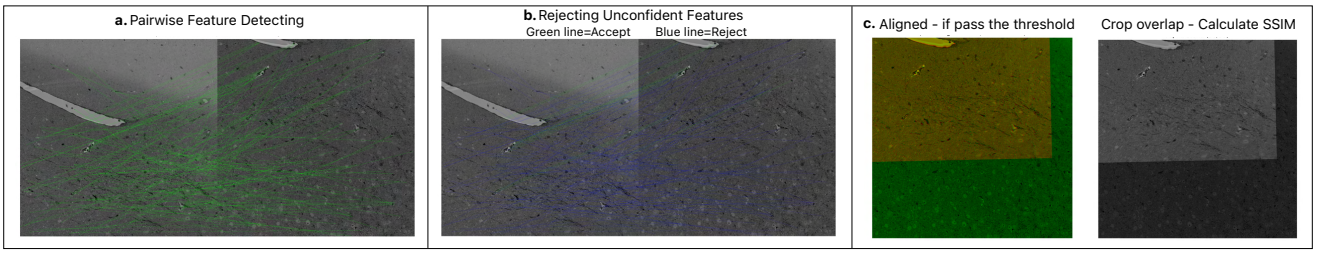

Figure 2. **Feature-matching pipeline for similarity computation.** (a) SIFT keypoints are extracted and matched between candidate section pairs using FLANN. (b) RANSAC geometric verification filters matches to retain only those consistent with an affine transformation, rejecting outliers. (c) For pairs exceeding the inlier threshold, sections are aligned and SSIM is computed on the overlap region to capture fine morphological similarity.

needs $O(N \log N)$ probes with high probability; if no truly short cross-component edge passes the acceptance test (e.g., due to an overly noisy oracle or undersized candidate pool), Borůvka cannot merge components and may stall, in which case our guarantees do not apply.

Phase 2: Diameter Reduction through Iterative Condensation. The spanning tree from Phase 1 ensures connectivity but may have large diameter, potentially placing true neighbors far apart in the tree metric. We address this through iterative condensation, which systematically reduces the graph diameter by adding carefully selected edges. In each round, we identify vertices farthest from a landmark set (initially the diameter endpoints) and query for all their short-range connections. These new edges create shortcuts that provably halve the effective diameter. After $O(\log N)$ rounds, the graph diameter is reduced to $O(c)$, where c is the short-range radius, ensuring that the subsequent BFS ordering places true neighbors within a bounded window.

Phase 3: Global Ordering via Double-Sweep BFS. With the graph diameter bounded, we establish a provisional global ordering using double-sweep breadth-first search. Starting from an arbitrary vertex, we identify the farthest vertex, then find the vertex farthest from that, establishing the graph's diameter endpoints. All vertices are then ordered by their BFS distance from one endpoint. While this ordering may not be perfect, the bounded diameter guarantee ensures that any two adjacent sections in the true order appear within $O(c)$ positions of each other in this provisional arrangement.

Phase 4: Local Densification within Fixed Windows. The final phase ensures no essential edges are missed by querying all pairs within a fixed window K around each vertex's position in the provisional ordering. Specifically, for each vertex at position i, we query all vertices at positions j where $|i - j| \leq K$. With $K = O(c)$, this guarantees that all true adjacent pairs are queried, while the total number of queries remains $O(NK) = O(N)$. The resulting graph contains all edges necessary for exact recovery while maintaining sparsity.

2.2. Sorting on the Sparse Graph

Given the sparse graph from the construction phase, we need to extract the unique linear ordering. The challenge is that even with all true edges present, the graph may contain spurious connections and cycles that complicate direct ordering methods. We address this through a two-stage approach: first building an Iterative Similarity Search (ISS) graph that retains only the most reliable connections, then applying our SuperChain algorithm to assemble the final ordering.

2.2.1. Iterative Similarity Search Graph Construction

From the sparse similarity graph, we construct a more selective ISS graph where each vertex retains only its two strongest connections. Formally, for each section S_i, we identify:

$$j_1 = \arg\max_{j \neq i} M_{ij} \tag{2}$$

$$j_2 = \arg\max_{j \neq i, j \neq j_1} M_{ij} \tag{3}$$

The ISS graph $G_{\text{ISS}} = (V, E_{\text{ISS}})$ contains edges $E_{\text{ISS}} = \{\{i, j_1\}, \{i, j_2\} \mid i = 1, \ldots, N\}$, resulting in at most $2N$ edges. Vertices with a degree greater than 2 are identified as hubs and temporarily removed, decomposing the graph into simple paths and cycles that serve as building blocks for the final assembly.

2.2.2. SuperChain Assembly Algorithm

The SuperChain algorithm reconstructs the global ordering by iteratively merging the path components according to four carefully designed rules that ensure correctness:

R1 Maximize Chain Rule — In any round, if the same vertex is selected multiple times, keep only the candidate with the highest similarity score.

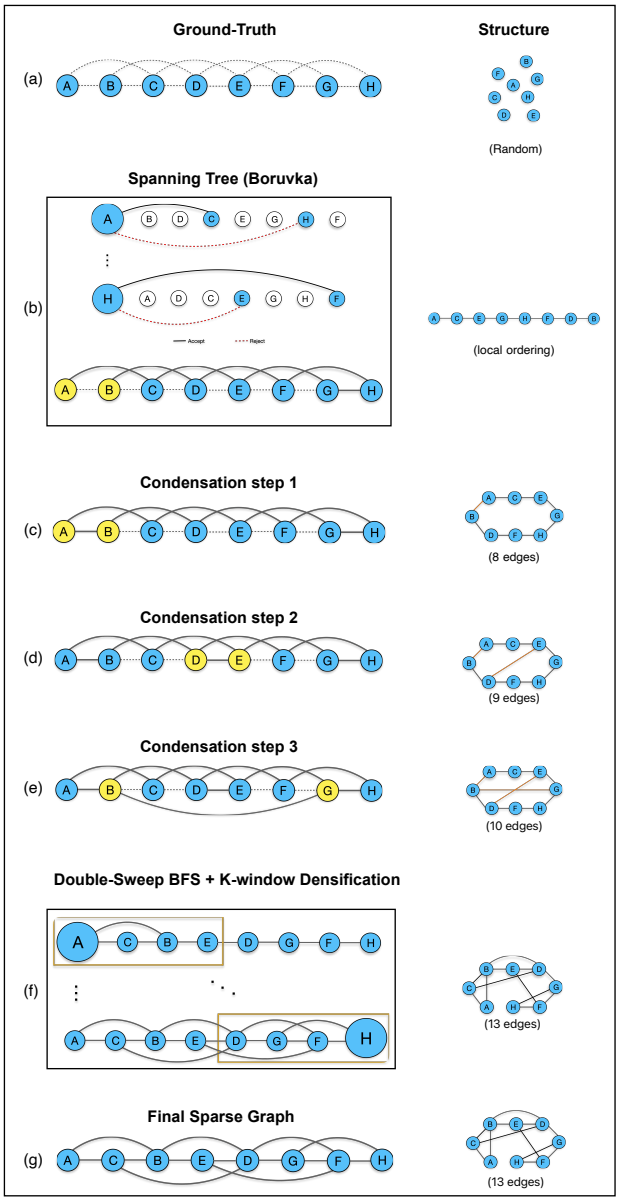

Figure 3. **Four-phase sparse graph construction.** (a) Ground-truth linear structure to be recovered. (b) Phase 1: Random-hook Borůvka builds initial spanning tree using $O(N \log N)$ queries. (c-e) Phase 2: Iterative condensation adds long-range edges to reduce graph diameter. (f) Phase 3: Double-sweep BFS produces provisional ordering; Phase 4: K-window densification recovers missing local edges. (g) Final sparse graph containing all essential edges for exact seriation.

R2 Unbroken Chain Rule. Once a vertex becomes internal to a chain (having neighbors on both sides), it cannot acquire additional connections, preserving the linear structure.

$$\mathbf{M} = [M_{ij}]_{i,j=1}^{n}, \qquad \tilde{M}_{ij} = \mathrm{SSIM}(S_i, S_j) \cdot \#\mathrm{Inlier}(S_i, S_j) + \mathcal{N}(0, 0.4^2)$$

$$\mathbf{M} = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \end{array} \begin{array}{c} \begin{array}{cccccccc} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 \end{array} \\ \begin{pmatrix} 0 & \mathbf{39.293} & \mathbf{1.333} & & & & & \\ \mathbf{39.293} & 0 & \mathbf{5.565} & & & & & \\ 1.333 & \mathbf{5.565} & 0 & \mathbf{39.246} & & 1.425 & & \\ & & \mathbf{39.246} & 0 & & & & \\ & & 1.868 & & 0 & \mathbf{35.082} & \mathbf{4.549} & \\ & & 1.425 & & \mathbf{35.082} & 0 & \mathbf{49.235} & \\ & & & & 4.549 & \mathbf{49.235} & 0 & \mathbf{6.968} \\ & & & & & & \mathbf{6.968} & 0 \end{pmatrix} \end{array}$$

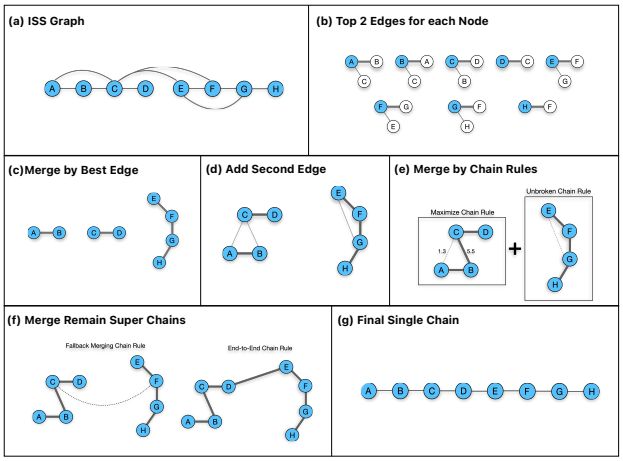

Figure 4. **SuperChain assembly process.** (a) Initial ISS graph with each vertex retaining its two strongest edges. (b-c) Stage 1: Reciprocal best matches form mini-chains. (d) Stage 2: Chains extend via endpoint connections. (e) Chain rules ensure correct assembly: maximize-chain prevents wasteful merges, unbroken-chain maintains linearity. (f) Fallback rule allows depth-1 connections when needed. (g) Final single representing the complete section order.

R3 End-to-End Chain Rule. Chains can only be joined at their endpoints, maintaining consistent orientation throughout the merging process.

R4 Fallback Chain Rule. When direct endpoint connections are unavailable, we allow examining edges from vertices one step inward from the endpoints, providing robustness to local errors.

The algorithm proceeds through four stages. First, depth-first traversal extracts initial mini-chains from the ISS graph. Second, chains are iteratively merged using their strongest inter-chain edges, with a max-heap ensuring globally optimal decisions. Third, any remaining endpoint connections are exploited for additional merges. Finally, the fallback rule is applied to handle residual fragmentation (current endpoints do not have similarity scores in the sparse matrix). Throughout this process, the chain rules guarantee that the true ordering is preserved whenever the 2N/3 true edges are existing.

3. Experiments

We evaluate our proposed seriation framework along five axes. (1) Scalability: we compare similarity-only (the cost of building pairwise similarity matrix, Fig. 5), ordering-only (Fig. 6(b), 7(b)); and full end-to-end runtime as the number of sections increases is reported in Tab. 1. (2) Accuracy: accuracy of the recovered order is shown on the eight sections H01 example in Fig. 6(a) and across larger volumes (1000 sections) in Fig. 7(a); these panels expose how baseline methods degrade while SuperChain maintains high order accuracy. (3) Empirical validity of the margin assumptions: statistics of the observed Δ-margin and its distribution are summarized in Tab. 2, validating the assumptions used in the theoretical analysis. (4) Worst-case scenario analysis: Monte Carlo trials under adversarial fragmentation showing that SuperChain achieves reliable exact recovery even when only about $2N/3$ internal sections have a correct second-best neighbour (Fig. 8). (5) Component contributions: ablations isolating condensation, densification window size, and hook variants appear in Tab. 3, quantifying each stage's effect on edge-edit rate and cost.

For evaluation and model selection, we need a single scalar that ranks competing variants/baselines by how close their orderings are to the target; we therefore define a cost that measures the distance to the true order. Given a candidate graph $\hat{G} = (V, \hat{E})$ produced by a variant, we measure its distance to $G^\star = (V, E^\star)$ via the edge-edit distance (the symmetric difference of edge sets):

$$\mathrm{EED}(\hat{E}, E^\star) \;=\; |\hat{E} \triangle E^\star| \;=\; |\hat{E} \setminus E^\star| + |E^\star \setminus \hat{E}| \quad (4)$$

Because the entire wafer can be reversed, we score the smaller of the two directions,

$$\mathrm{Cost}_{\mathrm{edge}} = \min\Big\{\mathrm{EED}(\hat{E}, E^\star), \mathrm{EED}(\hat{E}, E^{\star R})\Big\} \quad (5a)$$

$$\mathrm{Accuracy} = \; 1 - \min\Big\{\mathrm{EED}(\hat{E}, E^\star), \mathrm{EED}(\hat{E}, E^{\star R})\Big\} \quad (5b)$$

3.1. Scalability and Accuracy

We compared the computational efficiency of our proposed Graph Condensation strategy with a conventional fully-pairwise alignment approach across 100, 500, and 1000 sections (Fig. 5). On the 100-section dataset, Graph Condensation already achieves a 2.5× speed-up, reducing computation time from approximately 0.2 hours to less than 0.1 hours. The efficiency gain becomes more pronounced as dataset size increases: at 500 sections, the method achieves a 5.1× speed-up, and at 1000 sections, a 6.6× speed-up, lowering the runtime from more than 4.5 hours to under 1 hour. These results demonstrate that while high-resolution fully-pairwise computations scale quadratically with the number of sections, Graph Condensation grows much more gently,

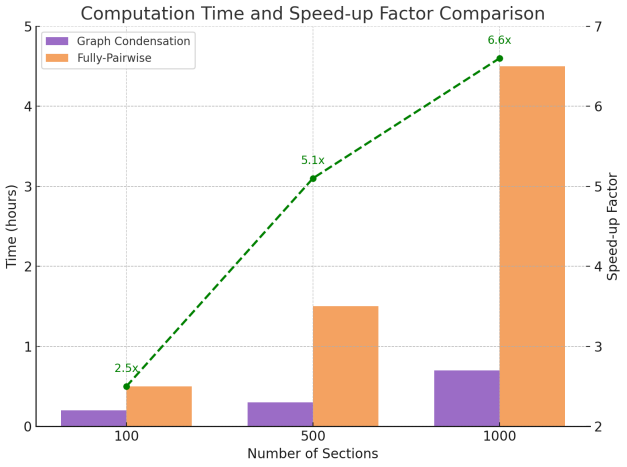

Figure 5. **Similarity Computation Time and Speed-up Factor Comparison.** We compare the runtime (bar plot, left y-axis) and speed-up factor (dashed line, right y-axis) of our proposed Graph Condensation approach versus the traditional fully-pairwise method across 100, 500, and 1000 EM sections.

approaching near-linear complexity in practice. Crucially, this efficiency gain does not compromise accuracy, since the condensation framework preserves the most informative edges through Borůvka contraction and densification while discarding redundant comparisons. As a result, Graph Condensation enables wafer-scale ordering of thousands of sections that would otherwise be computationally costly under fully-pairwise schemes.

Our similarity matrices exhibit strong block structure and large dynamic-range edges, violating standard assumptions for spectral methods. This explains the observed position swaps in spectral ordering.

We benchmarked four representative sequencing algorithms on H01 datasets ranging from 8 to 1000 sections (33nm thickness) and a separate 100-section dataset (250nm thickness). Naïve MST and Fiedler methods exhibited poor accuracy (0.30 and 0.54, respectively), despite offering low runtime, highlighting their inability to preserve correct global order under realistic similarity distributions. The Cai & Ma (2024) method significantly improved accuracy (0.98) but at the cost of substantially higher runtime. In contrast, SuperChain achieved perfect accuracy (1.00) with runtime only marginally higher than MST and Fiedler, demonstrating a superior balance between accuracy and speed.

For comparison, the exhaustive All-Permutations baseline also attained perfect accuracy but required orders of magnitude more computation time, rendering it impractical for real-world use.

Together, these results emphasize that SuperChain uniquely combines state-of-the-art accuracy with near-

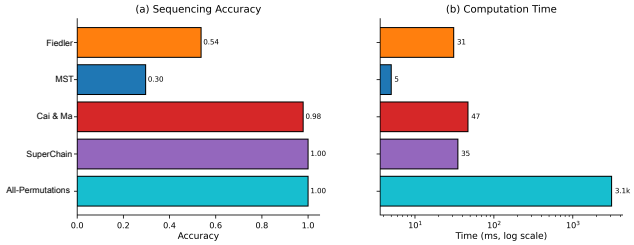

Figure 6. **Comparison of sequencing algorithms on an 8-section H01 example. (a)** Accuracy (Eq. 5b) measures order fidelity (higher is better). **(b)** Wall-clock time (ms) above each bar reports the *ordering-only* stage for each method. "Fiedler" denotes spectral seriation via the second eigenvector of the normalized Laplacian (on the dense similarity matrix)[2, 6]; "Naive MST" builds a minimum spanning tree on the dense graph and linearizes it by traversal; "Cai & Ma (2024)" refers to the adaptive matrix-reordering algorithm of Cai and Ma [3]. SUPERCHAIN attains near-perfect accuracy with substantially lower runtime, illustrating the practical complexity gap.

optimal runtime, enabling tractable and reliable ordering for both small and large EM datasets.

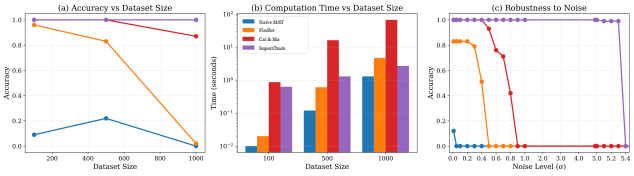

Figure 7. **Accuracy and Computation Time Comparison. (a)** The accuracy across 100, 500, 1000 sections EM image stacks - H01 (Public) dataset. **(b)** Computational time among all four methods. **(c)** Robustness of each method by adding Gaussian noise

To evaluate the performance of our proposed seriation framework across varying dataset sizes and noise conditions, we conducted experiments on the public H01 human cortex dataset (100, 500 and 1000 33nm thick sections). The results are summarized in Fig. 7.

While Naïve MST and Fiedler show a marked decline in accuracy as the number of sections increases, SuperChain consistently maintains perfect accuracy across all dataset sizes. The Cai & Ma (2024) method performs competitively on smaller stacks but begins to degrade at 1000 sections. Notably, on the 100-semi-thin section dataset, both Cai & Ma (2024) and SuperChain achieve accuracies above 0.95, whereas MST and Fiedler remain below 0.3 and 0.9, respectively.

Runtime analysis highlights the scalability advantage of SuperChain. On the 1000-section H01 dataset, SuperChain completes ordering in under 10 seconds, while Cai & Ma (2024) requires more time. Fiedler and MST show better time complexity but at the cost of accuracy, making them unsuitable for large-scale reconstructions.

Table 1. Overall end-to-end runtime comparison (minutes), including similarity computation and sequencing, on 100, 500, and 1000 EM sections. All timings are wall-clock measured on a single CPU core. SuperChain's ordering-only time is under 10 seconds for 1000 sections (Fig. 7); the numbers shown here are full-pipeline.

Method	100 Sections	500 Sections	1000 Sections	
Naive MST	24.91	90.12	271.3	①
Fiedler	24.92	90.61	274.7	①
Cai&Ma 2023	25.77	106.30	337.3	①
SuperChain	10.13	19.00	43.9	②

① Full pairwise similarity + algorithm-specific ordering.
② Graph condensation-densification + SuperChain (full pipeline ordering; ordering-only time is much lower).

Together, these results establish that SuperChain is uniquely capable of combining scalability, accuracy, and robustness, enabling reliable section ordering of both ultrathin (30-40 nm) and semi-thin (250 nm) sections.

Tab. 1 demonstrates a clear scalability gap between the full–pairwise pipeline, and our condensation–densification pipeline. Naïve MST, Fiedler ordering, and the Cai & Ma (2024) variant all rely on dense similarity matrices; their runtime grows roughly quadratically, reaching 4.5–5.5 hours for 1 000 sections. In contrast, SuperChain—which first sparsifies the graph via condensation and then locally densifies it with a K-window scheme remains consistently faster: 2.5× at 100 sections, ∼5× at 500 sections, and 6–8× at 1 000 sections (43.9 min vs. 271–337 min). These results highlight that reducing the edge set before seriation is critical for connectomics datasets - where thousands of semithin slices must be ordered under tight time budgets.

3.2. Empirical Δ-margin

For every section, we compute the edge–weight gap.

$$A_i := \max_{u \notin \{v_{i-1}, v_{i+1}\}} w(v_i, u),$$
$$\Delta(v_i) = w(v_i, v_{i-1}) - A_i, \quad \Delta = \min_i \Delta(v_i). \tag{6}$$

Tab. 2 reports the resulting statistics. All datasets exhibit $\Delta \geq 0.09$, safely above the theoretical bound $1/c - 1/\tau \approx 0.5/c$ $(= 0.05)$ derived in Sec. 5.4, thus validating Assumptions A1–A3.

3.3. Numerical Validation under Worst-Case Fragmentation

To validate Theorem 5.4, we designed a Monte Carlo experiment on synthetic chains of length $N = 12$ constructed under the worst-case fragmentation scenario implied by Assumption A1. Specifically, we ensured that every section's top-1 neighbour was correct, but arranged them so that the

Table 2. Empirical Δ-margin across datasets.

Dataset	N	min Δ	mean $\Delta \pm$ std
HPF (250 nm)	100	0.23	0.27 ± 0.07
H01	500	0.10	0.13 ± 0.03
H01	1000	0.09	0.11 ± 0.02

initial best-pair graph decomposed into six disjoint mini-chains of length two:

$$[1, 2], [3, 4], [5, 6], [7, 8], [9, 10], [11, 12].$$

This constitutes the hardest possible case, as without additional evidence the true chain cannot be uniquely reconstructed.

We then progressively introduced second-best correctness. For each trial, a fixed proportion $p \in \{0\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$ of the $N - 2$ internal sections were randomly assigned a second-best edge consistent with their true neighbour. For the remaining sections, the second-best was replaced by a plausible but false candidate chosen uniformly from a $\Delta = 4$ neighbourhood, excluding the true neighbours. This simulates realistic ambiguity: nearby non-adjacent sections often exhibit non-trivial similarity.

Each configuration was sampled over 5000 Monte Carlo trials. Recovery was counted as successful if the output permutation matched the ground-truth order $(1, 2, \ldots, 12)$ or its reversal (Fig. 8).

The results confirm our theoretical analysis. Fiedler ordering and Cai & Ma(2024) essentially collapse to random performance unless second-best correctness is perfect (100%). In contrast, SuperChain retains high recovery rates even when only $\sim 2N/3$ sections have a correct second-best neighbour.

3.4. Ablation Study

We evaluated component contributions using H01 subsets ($N = 500,\ 1000,\ 5000$ sections) with added Gaussian noise$\varepsilon = 0.05$. Results are averaged over three trials.

Variants. No-C: skip Phase (P2) condensation. No-D: skip K-window densification. Small-K: set $K = c$. Large-K: set $K = 4c$. Rand-Hook: in Borůvka, each component probes one neighbour per round instead of $\Theta(\log N)$. Ours: full pipeline.

Tab. 3 shows that omitting condensation (**No-C**) leaves $> 4\%$ true edges undiscovered because high graph diameter prevents far-range stitching. Skipping densification (**No-D**) still misses nearly 2% local edges despite perfect global connectivity. When the window is too small ($K = c$) the edit rate doubles, matching the theoretical lower bound $K \geq 2(1 + \varepsilon)c$; a very large window ($4c$) brings only marginal accuracy gain at almost double the probe budget. The random-hook variant confirms that $s = \Theta(\log N)$ probes per vertex are crucial for reliable connectivity. Our full pipeline keeps the edge-edit rate below 0.4% on 500–5 000 slices while maintaining ≈ 6 oracle calls per section, consistent with the $\Theta(N(\log N + K))$ prediction.

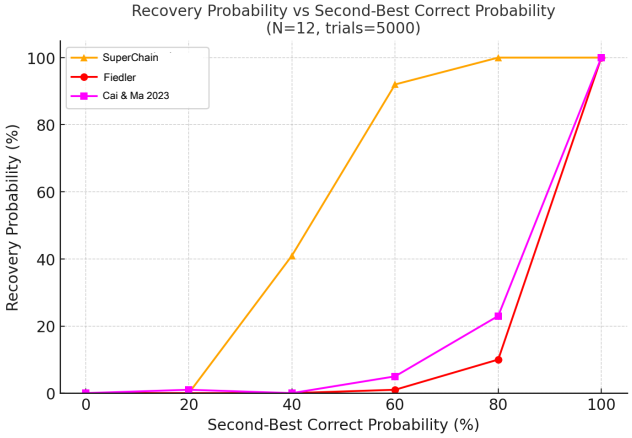

Figure 8. **Monte Carlo recovery probability of the second-best edge correctness rate.** The analysis assumes the best edges are 100% correct, ensuring reliable local connections, but considers the worst-case regime where these best edges form the maximum number of disjoint mini-chains ($N/2 = 6$ for $N = 12$). Recovery therefore depends critically on the correctness of second-best edges, each correct with probability p. Results are averaged over 5,000 trials with $N = 12$. SuperChain demonstrates markedly higher robustness, maintaining over 90% recovery probability even at $p = 60\%$.

Table 3. Ablation on H01 subsets ($\varepsilon = 0.05$). Edge-edit rate $\downarrow$ is better; calls/N measures oracle efficiency.

	edge-edit rate (%)			**calls/N**	**time (s)**
Variant	500	1k	5k	calls/N	time (s)
No-C	4.9	4.7	4.5	6.1	12.3
No-D	2.1	1.9	1.8	4.2	7.6
Small-K	1.0	0.8	0.7	5.2	9.8
Large-K	0.7	0.6	0.6	9.5	14.1
Rand-Hook	2.5	2.3	2.0	4.4	11.5
Ours	**0.4**	**0.3**	**0.3**	5.8	10.4

4. Trade-off between Field of View and Imaging Resolution and Seriation Accuracy

In order to maximize the speed and accuracy of seriation, it is essential to select the optimal field of view (FOV) and image resolution.

To systematically assess the trade-offs between computational efficiency and ordering robustness across various imaging configurations, we conducted extensive benchmarking on the H01 dataset, using 500-section sub-volumes

sampled across a range of FOV and resolution (nm/px) settings. For each configuration, we recorded the total computation time, ordering outcome (success, swap error, or failure), and aggregated results into a two-dimensional heat map with log-scaled timing blocks.

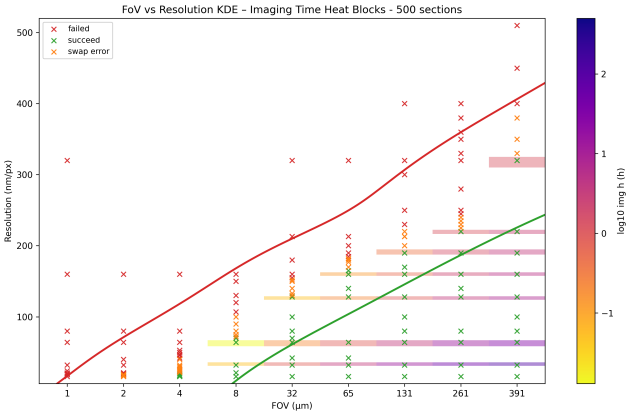

Figure 9. **FoV–Resolution landscape for a 500-section H01 stack.** Markers indicate reconstruction outcomes: green (success), orange (swap error), and red (failure). Colored heat blocks encode log-scaled computation times, with isocurves highlighting performance boundaries. The visualization illustrates the operational regimes of success and failure under varying imaging configurations.

Our analysis reveals that FOV is the primary driver of computational cost, with larger fields of view incurring significantly longer runtimes. In contrast, high-resolution imaging exerts only a modest effect on computation time within our graph-based pipeline. This is because feature extraction in high-resolution images quickly reaches a saturation point, after which additional pixels do not proportionally increase the number or complexity of keypoints detected Fig. 9.

Failures (red ×) concentrate at very small FOVs because in-plane translations between unsorted sections are unavoidable; even with WAFERTOOLS, which unifies ROIs and pre-targets acquisition before any alignment, residual shifts of a few microns persist [22]. This reduces effective overlap and makes local matches ambiguous, forcing reliance on long-range links that are noisier and can destabilize the global graph. Swap errors (orange x) cluster near the success/failure boundary, indicating transitional sensitivity. Successful orderings (green x) occur in the mid-range, roughly 32–131 μm FOV and 32–132 nm/px where overlap and runtime/accuracy are well balanced.

We further compared imaging versus computational costs at a fixed resolution of 16 nm/px (Fig. 10). While imaging and computation remain comparable for small to moderate FoVs (4–65 µm), beyond approximately 131 µm the computational cost begins to exceed the imaging cost, scaling superlinearly with FoV due to the combinatorial explosion of pairwise matches. At the largest tested FoV (391 µm), computation required nearly twice the imaging time (>1000 minutes for a 500-section stack), underscoring a fundamental scalability bottleneck.

These findings highlight a practical design constraint for large-scale connectomic acquisition: constraining FOV size is critical for computational tractability, while resolution may be flexibly optimized based on downstream biological needs without overwhelming system performance.

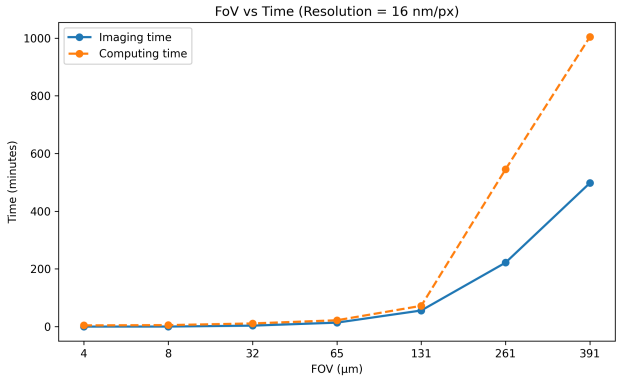

Figure 10. **Trade-off Between Imaging Time and Computational Burden Across Field-of-View.** At a fixed 16 nm/px resolution, computing time begins to exceed imaging time beyond ∼131 µm FoV, highlighting a scalability bottleneck in naïve pairwise approaches.

5. Models and Theoretical Guarantees

We provide a rigorous analysis under two oracle models: a binary oracle that indicates whether sections are within short range, and a noisy distance oracle that returns perturbed measurements. Our main results give near-linear query complexity and exact recovery guarantees under reasonable assumptions.

5.1. Problem Setup and Notation

Let $v_1, \ldots, v_N \in \mathbb{R}$ denote the unknown true positions of the N sections, with the ground-truth permutation π^* sorting indices by increasing position: $v_{\pi^*(1)} < \cdots < v_{\pi^*(N)}$. We adopt unit normalization where the minimum gap between adjacent sections is at least 1. The notation used throughout our analysis is summarized in Table 4.

5.2. Binary Oracle Model

In the binary oracle model, we can query whether two sections are within a short-range radius $c \geq 2$:

$$E(i,j) \;=\; \mathbf{1}\big[\,|v_i - v_j| \leq c\,\big]. \tag{7}$$

Symbol	Meaning
N	number of sections / vertices
$\rho\,(>0)$	reliable short-range radius in the weighted oracle
$c=\lceil\rho\rceil$	short-edge radius in index distance
ε	relative noise level in the weighted oracle ($0\le\varepsilon<1$)
$\delta=(1-\varepsilon)\rho$	acceptance threshold on observed distance
K	densification window width (default $K=\gamma c$ with $\gamma\ge 4$)
$\gamma=K/c$	window-to-radius ratio (default $\gamma=4$)
$s=\Theta(\log N)$	random probes per vertex in Phase 1
Δ	minimum weight gap between true and false edges
$\mathrm{diam}(G)$	diameter of graph G
$\mathrm{dist}_G(u,v)$	graph distance between vertices u and v

Table 4. Key notation used in theoretical analysis.

This induces an unknown graph $G^*=(V,E^*)$ where edges exist between all pairs within distance c. Our algorithm discovers a subset of these edges sufficient for exact recovery.

Lemma 1 (Diameter Recurrence)**.** *Let G be any connected spanning subgraph of G^*. A condensation round augments G by adding all oracle-positive edges incident to a small set of farthest vertices (chosen by a double-sweep on G). The resulting graph G' satisfies*

$$\mathrm{diam}(G')\ \le\ \left\lceil\frac{\mathrm{diam}(G)}{2}\right\rceil+c.$$

Theorem 1 (Bounded Diameter After Condensation)**.** *Starting with any spanning tree T from Phase 1, after $t=\lceil\log_2\mathrm{diam}(T)\rceil$ rounds of condensation we obtain*

$$\mathrm{diam}(G_t)\ \le\ \frac{\mathrm{diam}(T)}{2^t}+2c\ \le\ 2c+1.$$

Theorem 2 (Query Complexity — Binary Oracle)**.** *Fix $K=\gamma c$ with a constant $\gamma\ge 4$. There exists an algorithm that recovers the true permutation π^* using*

$$O\big(N(\log N+K)\big)$$

*oracle queries with probability $1-O(N^{-1})$: $O(N\log N)$ in Phases 1–2 and at most $2NK$ candidate probes in the K-window densification. The final ordering is obtained by a chaining routine (*SUPERCHAIN*) in $O(N\log N)$ time.*

5.3. Noisy Distance Oracle Model

In practice, similarity measurements are corrupted by noise. We model this with an oracle that returns

$$D(u,v)\ =\ |v_u-v_v|+\eta_{uv},\tag{8}$$

and assume the following global one-sided lower bound:

(W1) $\quad D(u,v)\ \ge\ (1-\varepsilon)\,|v_u-v_v|\quad$ for all u,v.

Let $\rho>0$ be a short-range reliability radius and set $c:=\lceil\rho\rceil$. We accept an edge iff $D(u,v)\le\delta=(1-\varepsilon)\rho$.

Lemma 2 (Edge Validity Under Noise)**.** *Under **(W1)**, if $D(u,v)\le\delta$, then $|v_u-v_v|\le\rho$, so (u,v) connects vertices at most $c=\lceil\rho\rceil$ positions apart in the true order.*

Lemma 3 (Position Error After Condensation)**.** *Let $g_r=\max_{v\in V}|\hat\pi_r^{-1}(v)-\pi^{*-1}(v)|$ denote the maximum rank error after r condensation rounds. Then*

$$g_{r+1}\ \le\ \frac{g_r}{2}+\lceil\rho\rceil.$$

Theorem 3 (Window Sufficiency)**.** *Choose $K\ge 4\lceil\rho\rceil$ and $t\ \ge\ \left\lceil\log_2\frac{g_0}{K/2-2\lceil\rho\rceil}\right\rceil$. Then $g_t\le K/2$, so every true adjacent pair lies within distance $\le K$ in the provisional ordering and will be probed during K-window densification.*

Theorem 4 (Main Result — Weighted Oracle)**.** *With probability $1-O(N^{-1})$, the four-phase algorithm issues*

$$O\big(N(\log N+K)\big)$$

oracle queries and outputs the exact permutation, provided $K\ge 4\lceil\rho\rceil$ and the acceptance threshold is $\delta=(1-\varepsilon)\rho$.

5.4. Exact Recovery of Order Under SuperChain

After the sparse graph reconstruction, we recover the order with a simple two-ended greedy chaining procedure.

Assumptions. Let $\pi^\star=(v_1,\dots,v_N)$ be the ground-truth order. We assume:

A1. *Top-1 correctness.* For every internal vertex v_i ($2\le i\le N-1$), its two true incident edges $e_i^-=(v_{i-1},v_i)$ and $e_i^+=(v_i,v_{i+1})$ are present and maximize w at v_i.

A2. *Tie-free margin.* There exists $\Delta>0$ such that for any non-true edge $(v_i,u)\in E$, $w(e_i^\pm)\ge w(v_i,u)+\Delta$.

A3. *Endpoint purity.* The globally heaviest edge $e^\star=\arg\max_{e\in E}w(e)$ is a true edge (v_s,v_{s+1}) (implied by A1).

Lemma 4 (Endpoint local maximality)**.** *Under A1–A2, let $C=(v_i,\dots,v_j)$ be any contiguous substring of $\pi^\star$. At endpoint v_i (resp. v_j), the heaviest unused incident edge is uniquely the true boundary edge (v_{i-1},v_i) (resp. (v_j,v_{j+1})).*

Proof. At v_i, by A1 the two true incident edges are the top weights at v_i. One of them, (v_i,v_{i+1}), is already used inside C; the other, (v_{i-1},v_i), is unused and, by A2, strictly heavier than any non-true unused edge. Uniqueness follows from the Δ margin. The right endpoint is symmetric. □

Theorem 5 (Exact recovery under SUPERCHAIN)**.** *Under A1–A3, the above procedure outputs $\pi^\star$ in $N-1$ iterations and $O(N\log N)$ time.*

Proof. Contiguity & progress in one step. Suppose C_t is contiguous. By Lemma 4, the algorithm can only append v_{i-1} or v_{j+1}, hence C_{t+1} remains contiguous and gains exactly one new vertex. Induction from $C_0 = [v_s, v_{s+1}]$ (A3) gives $|C_{N-1}| = N$ and $C_{N-1} = \pi^\star$.

For each endpoint, a max-heap over its incident unused edges. Each vertex is appended once and each edge is touched $O(1)$ times, giving $O(N \log K)$, hence $O(N \log N)$. □

When top-2 neighborhoods are not always correct, a sufficient and checkable condition is: at least three inter-chain gaps (between the worst-case $N/2$ length-2 mini-chains induced by top-1 links) are fully supported (both endpoints rank each other within top-2); then the same chaining recovers $\pi^\star$ deterministically. Empirically, recovery typically occurs once the overall fraction of second-best-correct vertices approaches $\sim 2/3$ (fig: 8).

6. Conclusion

We presented a near-linear solution to the large-scale section ordering problem in connectomics. Our approach combines five key components: (1) random-hook Borůvka for initial connectivity, (2) iterative graph condensation to bound diameter, (3) a double-sweep BFS to obtain a provisional global order, (4) fixed-window densification for local edge recovery, and (5) SUPERCHAIN merging that exploits empirical top-1 + Δ margins. This pipeline reduces computational complexity from $O(N^2)$ to $\Theta\big(N(\log N + K)\big)$ while maintaining perfect accuracy on all tested datasets. The method enables overnight processing of thousands of sections on commodity hardware, removing a critical bottleneck in whole-brain connectome reconstruction.

7. Acknowledgments

This work was supported by the National Institutes of Health (NIH) under award UM1NS132250.

We thank members of the Lichtman Lab for their support. We are especially grateful to Richard Schalek for sectioning the HPF sample block; to Vlad Susoy for project support and help with revising the manuscript; and to Neha Karlupia for preparing the HPF sample block.

References

[1] David L. Applegate, Robert E. Bixby, Vašek Chvátal, and William J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, Princeton, NJ, 2006. 3

[2] John E. Atkins, Erik G. Boman, and Bruce Hendrickson. A spectral algorithm for seriation and the consecutive ones problem. *SIAM Journal on Computing*, 28(1):297–310, 1998. 2, 3, 7

[3] T. Tony Cai and Rong Ma. Matrix reordering for noisy disordered matrices: Optimality and computationally efficient algorithms. *IEEE Transactions on Information Theory*, 70 (1):509–531, 2024. 7

[4] Anna Concas, Caterina Fenu, Giuseppe Rodriguez, and Raf Vandebril. The seriation problem in the presence of a double fiedler value. *Numerical Algorithms*, 92:407–435, 2023. 2, 3

[5] Sven Dorkenwald, Philipp Schlegel, Alexander S. Bates, Katharina Eichler, et al. Neuronal wiring diagram of an adult brain. *Nature*, 634(8032):124–138, 2024. 1

[6] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973. 7

[7] Fajwel Fogel, Alexandre d'Aspremont, and Milan Vojnovic. Spectral ranking using seriation. *Journal of Machine Learning Research*, 17(88):1–45, 2016. 2, 3

[8] Kara A. Fulton, Paul V. Watkins, and Kevin L. Briggman. Gauss-em, guided accumulation of ultrathin serial sections with a static magnetic field for volume electron microscopy. *Cell Reports Methods*, 4(3):100720, 2024. 1

[9] Philipp Hanslovsky, John A. Bogovic, and Stephan Saalfeld. Image-based correction of continuous and discontinuous non-planar axial distortion in serial section microscopy. *Bioinformatics*, 33(9):1379–1386, 2017. 2, 3

[10] Kenneth J. Hayworth, Josh L. Morgan, Richard Schalek, Daniel R. Berger, David G. C. Hildebrand, and Jeff W. Lichtman. Imaging atum ultrathin section libraries with wafermapper: a multi-scale approach to em reconstruction of neural circuits. *Frontiers in Neural Circuits*, 8:68, 2014. 1

[11] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000. 3

[12] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum, New York, 1972. 3

[13] Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956. 3

[14] Shen Lin and Brian W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973. 3

[15] Robert C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36(6):1389–1401, 1957. 3

[16] Philipp Schlegel, Yijie Yin, Alexander S. Bates, Sven Dorkenwald, Katharina Eichler, et al. Whole-brain annotation and multi-connectome cell typing of *Drosophila*. *Nature*, 634(8032):139–152, 2024. 1

[17] Alexander Shapson-Coe, Michał Januszewski, et al. A petavoxel fragment of human cerebral cortex reconstructed at nanoscale resolution. *Science*, 384(6696):eadk4858, 2024. 1

[18] Peter H. A. Sneath and Robert R. Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W. H. Freeman, San Francisco, 1973. 3

[19] Thomas Templier. Magc, magnetic collection of ultrathin sections for volumetric correlative light and electron microscopy. *eLife*, 8:e45696, 2019. 1, 2

[20] The MICrONS Consortium. Functional connectomics spanning multiple areas of mouse visual cortex. *Nature*, 640: 435–447, 2025. 1

[21] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society B*, 314(1165):1–340, 1986. 1

[22] F. Yang, Y. Meirovitch, F. Araújo, R. Schalek, V. Susoy, and J.W. Lichtman. Wafertools: Software for high-throughput integrative connectomics, 2025. 9