SNAP: NEURON SEGMENTATION FROM SUPERVOXELS AND PAIRWISE AFFINITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

The ability to accurately reconstruct the architecture of neurons is a prerequisite for understanding the function of natural neural circuits. Existing neuron segmentation methods often rely on complex post-processing pipelines that require careful hyperparameter tuning. We introduce SNAP, a neuron segmentation method that jointly predicts supervoxels and their mutual affinities, reducing inference to a simple affinity-driven agglomeration procedure. On the established AC3/AC4 and FIB-25 benchmarks, our method achieves state-of-the-art results with inference time comparable to conventional affinity-based approaches. Experiments on these datasets demonstrate the robustness of our method across imaging modalities and dataset scales. Beyond segmentation performance, we revisit commonly used evaluation protocols and clarify inconsistencies in dataset splits and evaluation settings that complicate comparison between existing methods. Our results show that state-of-the-art neuron segmentation can be achieved without complex post-processing while providing a clearer and more reproducible basis for future comparisons. Code is available at this Drive link.

1 INTRODUCTION

Context Connectomics studies the structure and connectivity of neural circuits. By reconstructing these circuits at single-neuron resolution, researchers can build detailed wiring maps known as connectomes. These maps help scientists investigate how the brain is organized and how it processes information (Morgan & Lichtman, 2013; Galili et al., 2022; Bates et al., 2026). Neuron reconstruction is challenging for two main reasons: neurons are densely packed, requiring very high resolution to distinguish them (typically with Electron Microscopy (EM)), and they extend over long distances, requiring the analysis of volumes containing up to petavoxels of data Shapson-Coe et al. (2024). With data-generation technologies expanding at an unprecedented pace, developing accurate and efficient segmentation methods has become increasingly critical.

Neuron reconstruction is an instance segmentation task that requires assigning a unique identifier to each neuron in very large volumes. Manual reconstruction is impractical because it is prohibitively time-consuming, motivating the development of automated reconstruction methods. Deep learning has been extensively used to solve connectomics problems at scale (Dorkenwald et al., 2024; Shapson-Coe et al., 2024; Bae et al., 2025; Nern et al., 2025). However, current neuron segmentation methods still suffer from a significant error rate, making it necessary to correct the segmentation maps produced by the models. This task, called proofreading, is one of the main bottlenecks for obtaining a full mouse brain connectome (Jefferis et al., 2023). This limitation is addressed through two complementary directions: (1) improving neuron segmentation methods (Januszewski et al., 2018; Funke et al., 2019; Sheridan et al., 2023), and (2) developing automated proofreading techniques (Troidl et al., 2025; Januszewski et al., 2025). In this work, we focus on the former direction.

Flood-Filling Networks (FFN) (Januszewski et al., 2018) is one of the most widely used methods for neuron reconstruction at scale (Scheffer et al., 2020; Svara et al., 2022; Nern et al., 2025). However, its computational cost makes it difficult for researchers with more limited resources to consider it. The other leading approach is based on affinity prediction between neighboring voxels (Lee et al., 2017; Funke et al., 2019; Sheridan et al., 2023). This approach has the benefit of being two orders of magnitude faster than FFN (Sheridan et al., 2023), but has shown less consistent performance across

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

benchmarks and scales (Sheridan et al., 2023). Both approaches rely heavily on post-processing steps, making their performance sensitive to the choice and tuning of post-processing parameters.

Contributions In this article, we propose an approach that achieves state-of-the-art performance while maintaining a low computational cost. Our key idea is to jointly learn the supervoxel partition and the affinity function used to merge supervoxels (see Section 3). Specifically, a 3D U-Net predicts a supervoxel assignment probability map from which we derive the supervoxel partition. The second part of our framework predicts affinities between supervoxels, employing a lightweight Transformer to capture long-range interactions among them. To the best of our knowledge, this is the first end-to-end differentiable framework to jointly learn supervoxels and their affinities for neuron segmentation. At inference time, the learned affinities reduce post-processing to a simple agglomeration step (see Section 3.3). Through experiments on the established benchmark datasets AC3/AC4 (Kasthuri et al., 2015) and FIB-25 (Takemura et al., 2015), we achieve state-of-the-art results while maintaining inference time comparable to conventional affinity-based approaches (see Section 4.5). Finally, we address inconsistencies in neuron segmentation evaluation by clarifying the dataset splits and protocols that prior work uses for the widely adopted AC3/AC4 and FIB-25 benchmarks (see Section 4.1). For FIB-25, we additionally propose a practical evaluation split that improves the reproducibility of scalability assessments.

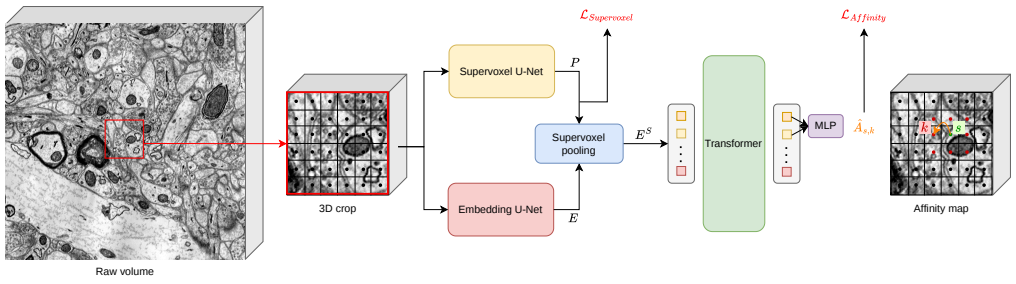


Figure 1: Overview of the method. A 3D crop is extracted from the raw volume and partitioned into a regular grid of candidate supervoxels. The crop is then processed by a Supervoxel U-Net that predicts $\boldsymbol{P}$, the probability for each voxel to be assigned to its neighboring candidate supervoxels. In parallel, an Embedding U-Net produces a dense embedding map $\boldsymbol{E}$ (one embedding per voxel). Supervoxel pooling uses the assignment map $\boldsymbol{P}$ to compute weighted averages of the dense embeddings, yielding supervoxel representations $\boldsymbol{E}^S$. The resulting supervoxel features are fed to a lightweight Transformer to capture long-range interactions, and pairs of transformed tokens are passed to an MLP to predict affinities $\hat{A}_{s,k}$ between neighboring supervoxels ((s, k) represented in green and red). Training uses the supervoxel prediction loss $\mathcal{L}_{Supervoxel}$ and affinity loss $\mathcal{L}_{Affinity}$.

2 Related works

Neuron segmentation in EM. Affinity segmentation is one of the main approaches for reconstructing neurons on a large scale (Dorkenwald et al., 2024; Bae et al., 2025). The first method proposed by Turaga et al. (2010) employed a CNN to predict the probability that adjacent voxels (front, upper, and left) belonged to the same neuron, yielding so-called "affinities". Several improvements to the original approach were proposed, including topology-aware losses (Briggman et al., 2009; Funke et al., 2019), or auxiliary losses (Lee et al., 2017; Sheridan et al., 2023). Further progress has been achieved through architectural enhancements of affinity prediction models (Sun et al., 2023; Luo et al., 2024; Li et al., 2026). The use of metric learning to derive a metric graph as affinities was also explored and delivered promising results (Lee et al., 2021). More recently, several alternative graph partitioning methods were proposed (Beier et al., 2017; Wolf et al., 2018; Bailoni et al., 2022). Notably, GASP (Bailoni et al., 2022) can perform agglomeration directly on pixel-level affinity predictions, although this approach is significantly slower.

Flood-filling network (Januszewski et al., 2018) is the other leading approach (Nern et al., 2025). It segments neurons one by one, with multiple passes in order to reduce merges as much as possible,

using heuristics to then merge objects together with oversegmentation consensus. This approach is approximatively two orders of magnitude slower than the affinity method (Sheridan et al., 2023).

Differently, cross-classification clustering (Meirovitch et al., 2019) encodes each neuron in every slice with a binary code. This code is then used to group neurons across slices. It requires multiple forward passes per slice and fails to improve upon long-range affinities (Lee et al., 2017).

Finally, direct instance segmentation on patches followed by stitching was recently proposed by Chen et al. (2025). They demonstrate interesting results with increased efficiency. However, evaluation was limited to small serial-section EM datasets, leaving scalability and applicability to other imaging modalities to be established.

Superpixel segmentation. Superpixel segmentation is the task of grouping pixels into perceptually meaningful and compact regions. Superpixels can serve as a starting point for other downstream tasks such as segmentation (Lucchi et al., 2012) or stereo matching (Yang et al., 2020). Superpixels improve downstream efficiency while maintaining dense precision.

In neuron segmentation, superpixels are already used within affinity-based pipelines to make the problem computationally tractable at connectome scale (Funke et al., 2019). A typical procedure for deriving a segmentation from affinities is as follows: 1) the predicted affinities are converted into a boundary map by averaging the three affinity offsets and binarizing the result using a predefined threshold; 2) a distance transform is computed from the resulting boundaries; 3) seeds are placed on the distance map, followed by a watershed-based oversegmentation to obtain superpixels; and 4) hierarchical agglomerative merging is applied to the resulting oversegmentation, guided by the affinity map. These post-processing steps can introduce potential difficulties regarding the tuning of their associated parameters. For instance, step 1) may create split errors for very thin objects (see Appendix A for an illustration). In addition, the inference procedure requires manual tuning of the binarization threshold in the first step, the minimum seed distance in step 3), and the agglomeration threshold, which may require additional effort in practice. Another example of method employing superpixels in EM is the use of the SLIC algorithm (Achanta et al., 2012) for mitochondria segmentation (Lucchi et al., 2010; 2012). This approach enabled efficient segmentation while being accurate, but relied on hand-crafted features to predict merge decisions. DEL (Liu et al., 2018) on the other hand, learns merge decisions with an embedding net that produces superpixel features. However, this approach was only tested on natural images.

Deep superpixel segmentation. Recently, differentiable superpixel algorithms showed improved accuracy and downstream performance while being more efficient than their non-differentiable counterparts (Jampani et al., 2018; Yang et al., 2020). When integrated with deep learning models, this approach enables the construction of fully end-to-end training pipelines (Jampani et al., 2018; Yang et al., 2020; Wang et al., 2020). One disadvantage compared to the other algorithms is the need for a post-processing step to enforce superpixel connectivity. Juybari et al. (2026) has recently proposed a differentiable loss term specifically designed to obviate this limitation. Most differentiable superpixel methods, however, were designed for 2D images and therefore require adaptation to handle volumetric data.

3 METHODS

Our method builds on regularized differentiable superpixel segmentation (Juybari et al., 2026) and learned superpixel-level similarity prediction (Liu et al., 2018; Wang et al., 2020), extending these ideas to 3D neuron instance segmentation. In our formulation, supervoxel representations are pooled directly through soft voxel-to-supervoxel assignments, allowing for a fully differentiable pipeline from supervoxels to merge decisions. We reformulate the underlying components of previous methods to operate on volumetric data and to address the additional challenges associated with separating individual neuron instances. We additionally propose a dedicated inference procedure for large-scale volumetric segmentation. An illustration of the whole framework is provided in Figure 1.

3.1 SUPERVOXEL SEGMENTATION MODEL

Consider a 3D volume of shape $H \times W \times D$, with height H, width W, and depth D, and let Ω denote its set of $N = HWD$ voxels. We adopt the local assignment formulation commonly used

in superpixel segmentation (Achanta et al., 2012; Jampani et al., 2018; Yang et al., 2020; Wang et al., 2021; Xu et al., 2024; Juybari et al., 2026) and extend it to 3D volumes. We first partition the volume into a regular 3D grid of stride S, whose $M = HWD/S^3$ cells define the set $\mathcal{S}$ of candidate supervoxel seeds. As in the 2D formulation, we restrict voxel-to-supervoxel assignments to spatially neighboring grid cells. In 3D, for each voxel v, we define its neighborhood $\mathcal{N}(v)$ as the $3 \times 3 \times 3$ block of grid cells centered on the cell containing v, yielding 27 candidate supervoxels. We employ a 3D U-Net (Supervoxel U-Net) to predict for each voxel v, the probability $p_s(v)$ of assigning this voxel to each candidate supervoxel s in $\mathcal{N}(v)$. We apply a softmax over the 27 candidates so that $\sum_{s\in\mathcal{N}(v)} p_s(v) = 1$. These assignment probabilities can be rewritten in a sparse matrix $\boldsymbol{P} \in \mathbb{R}^{M\times N}$, whose entry $P_{s,v}$ is the probability $p_s(v)$ when $s \in \mathcal{N}(v)$ and zero otherwise.

To train the Supervoxel U-Net, we follow (Jampani et al., 2018; Yang et al., 2020; Wang et al., 2021; Xu et al., 2024; Juybari et al., 2026) and use voxel-level supervision. We train the model to predict supervoxel assignments that preserve a target voxel-wise quantity. In our setting, this target is the ground-truth neuron instance segmentation. For convenience, we encode the instance labels as one-hot vectors in a matrix $\boldsymbol{Y} \in \{0,1\}^{N\times C}$, where each row $\boldsymbol{y}_v$ represents the instance label of voxel v, and C is the number of instances in a training crop, including background. To make the prediction comparable with the ground truth, we first aggregate the instance matrix $\boldsymbol{Y}$ into a supervoxel representation $\boldsymbol{Y}^S \in \mathbb{R}^{M\times C}$ and then project it back to voxel labels $\hat{\boldsymbol{Y}} \in \mathbb{R}^{N\times C}$. Mathematically, calling $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{P}\mathbf{1}_N)$ the expected size of each supervoxel, we first define $\boldsymbol{Y}^S$ as the *soft instance-label representation* of the supervoxels, obtained by aggregating the voxel-level instance labels according to the voxel-to-supervoxel assignment probabilities: $\boldsymbol{Y}^S = \boldsymbol{D}^{-1}\boldsymbol{P}\boldsymbol{Y}$. Each entry Y^S_{sc} represents the normalized assignment weight of supervoxel s to instance c:

$$Y^S_{sc} = \frac{\sum_{v=1}^{N} P_{sv} Y_{vc}}{\sum_{v=1}^{N} P_{sv}}. \tag{1}$$

Thus, each row of $\boldsymbol{Y}^S$ defines a probability distribution over the C instances for a given supervoxel. We then project the supervoxel representations back to the voxel level as $\hat{\boldsymbol{Y}} = \boldsymbol{P}^\top \boldsymbol{Y}^S$. Specifically, each entry is given by

$$\hat{Y}_{vc} = \sum_{s=1}^{M} P_{sv} Y^S_{sc}. \tag{2}$$

Because $\boldsymbol{P}$ is column-normalized, each row of $\hat{\boldsymbol{Y}}$ defines a probability distribution over the C instances.

We adopt the loss formulation of Juybari et al. (2026) for SCN (Yang et al., 2020), consisting of a weighted reconstruction loss $\mathcal{L}_{WR}$, a compactness loss $\mathcal{L}_C$, a minimum semantic distance loss $\mathcal{L}_{MSD}$, and a graph-Laplacian loss $\mathcal{L}_{LAP}$. The weighted reconstruction loss $\mathcal{L}_{WR}$ applies cross-entropy between the reconstructed voxel labels $\hat{\boldsymbol{Y}}$ and the ground-truth instance labels $\boldsymbol{Y}$, with larger weights assigned to grid cells containing multiple instances in order to emphasize boundary accuracy. The compactness loss $\mathcal{L}_C$ uses the original voxel coordinates $\boldsymbol{C} \in \mathbb{R}^{N\times 3}$ as training target and enforce spatially compact supervoxel by minizing the distance with reconstructed coordinates $\hat{\boldsymbol{C}} = \boldsymbol{P}^T\boldsymbol{D}^{-1}\boldsymbol{P}\boldsymbol{C}$. The minimum semantic distance loss $\mathcal{L}_{MSD}$ encourages the Supervoxel U-Net to predict voxel embeddings that remain separated by a prescribed margin across different neuron instances. We use the same margin value as Juybari et al. (2026) ($\delta = 1.5$). Finally, $\mathcal{L}_{LAP}$ promotes spatial connectivity within each predicted supervoxel by rewarding neighboring voxels with high assignment probabilities to the same supervoxel. Let $\Omega_s = \{v \in \Omega : s \in \mathcal{N}(v)\}$, $N_S = |\Omega_s|$, and let $\mathcal{A}(v)$ denote the 26-connected voxel neighborhood. We extend the graph-Laplacian loss of Juybari et al. (2026) to 3D with

$$\mathcal{L}_{LAP} = 1 - \frac{1}{M}\sum_{s\in\mathcal{S}} \frac{1}{26N_s} \sum_{v\in\Omega_s} \sum_{u\in\Omega_s\cap\mathcal{A}(V)} p_s(v)p_s(u).$$

This corresponds to their original formulation with the 8-connected pixel neighborhood and maximum degree 8 replaced by the 26-connected voxel neighborhood and maximum degree 26.

The complete objective is:

$$\mathcal{L}_{Supervoxel} = \lambda_C\mathcal{L}_C + \lambda_{WR}\mathcal{L}_{WR} + \lambda_{MSD}\mathcal{L}_{MSD} + \lambda_{LAP}\mathcal{L}_{LAP},$$

where λ_C, λ_{WR}, λ_{MSD}, and λ_{LAP} are user-defined hyperparameters. We adapt standard values from the literature to account for our specific crop sizes and 3D setting (Yang et al., 2020; Wang et al., 2021; Juybari et al., 2026). More details can be found in Appendix B. Only λ_{LAP}, which is a new hyper-parameter controlling the trade-off between supervoxel connectivity and boundary preservation, is empirically found using a grid search described in Appendix C.

3.2 Affinity model on supervoxels

We train the second part of our framework to predict affinities between supervoxels. To this end, we employ another 3D U-Net (Embedding U-Net), which predicts a dense voxel embedding $\boldsymbol{E}$. Following the region-level feature aggregation used by DEL (Liu et al., 2018) and Wang et al. (2020), we pool the voxel embeddings into supervoxel representations. In contrast to these approaches, we perform the pooling directly through the predicted soft voxel-to-supervoxel assignments, yielding $\boldsymbol{E}^S = \boldsymbol{D}^{-1}\boldsymbol{P}\boldsymbol{E}$. This formulation allows gradients from the affinity objective to propagate through $\boldsymbol{P}$ to the Supervoxel U-Net. Liu et al. (2018) and Wang et al. (2020) derive pairwise similarities directly from the resulting region-level features. Instead, we first contextualize the supervoxel embeddings using a lightweight Transformer. This is motivated by the difficulty of producing consistent embeddings for elongated neuron instances and by the limited spatial context available from the CNN receptive field (Lee et al., 2021). Multi-head self-attention allows us to integrate long-range interactions into the supervoxel features. We linearly project the supervoxel embeddings and add 3D sinusoidal positional encodings to form the input tokens of a lightweight Transformer encoder (Vaswani et al., 2017). We estimate the supervoxel positions $\boldsymbol{C}^S$ using the same pooling operation $\boldsymbol{C}^S = \boldsymbol{D}^{-1}\boldsymbol{P}\boldsymbol{C}$. We then explicitly predict affinities from pairs of contextualized supervoxel embeddings using an MLP. Specifically, for each supervoxel s, we denote by $\mathcal{V}(s)$ the set of its 26 neighbors in the regular grid and predict a merge probability $\hat{A}_{s,k}$ for each $k \in \mathcal{V}(s)$. To construct the training targets, we follow DEL and assign each supervoxel to a neuron when its pseudo-label probability in $\boldsymbol{Y}^S$ exceeds 0.7. The target affinity $A_{s,k}$ is then one if supervoxels s and k are assigned to the same neuron, and zero otherwise. While our affinity-prediction architecture differs from DEL (see Section 4.2), we retain the same loss formulation and train the affinity predictor using a balanced binary cross-entropy BCE_α loss:

$$\mathcal{L}_{\text{Affinity}} = \frac{1}{M} \sum_{s \in \mathcal{S}} \frac{1}{|\mathcal{V}(s)|} \sum_{k \in \mathcal{V}(s)} \mathrm{BCE}_\alpha \left(A_{s,k}, \hat{A}_{s,k} \right),$$

where α is the proportion of positive affinities. The total loss is the sum of all components:

$$\mathcal{L} = \lambda_{Supervoxel} \mathcal{L}_{Supervoxel} + \mathcal{L}_{Affinity},$$

with $\lambda_{Supervoxel}$ another hyperparameter empirically found through grid search (see Appendix C).

3.3 Overlapped inference

Model predictions. To segment the validation and test sets, we process each volume in a 3D patch-wise manner. The models operate on overlapping patches with a stride equal to half the patch size along each dimension. At the volume boundaries, we apply padding so that all voxels, including those near the borders, are covered by multiple overlapping patches. The padding strategy is dataset-dependent and is described in Section 4.3. For each patch, the model predicts supervoxel assignment probabilities and affinities between candidate supervoxels (see Figure 1). To combine predictions from overlapping patches, we weight each patch using a 3D Hann window. We then aggregate the weighted assignment predictions into a global assignment tensor in $[0, 1]^{H \times W \times D \times 27}$. Similarly, we aggregate the affinity predictions into a global sparse affinity matrix containing the predicted affinities between all neighboring supervoxel pairs in the volume.

Agglomerative clustering. We first perform a voxel-wise argmax over the 27 regular-grid candidates to select one candidate for each voxel, whose grid location then determines its global supervoxel identity. We discard regular-grid candidates to which no voxel is assigned. The remaining supervoxels constitute the nodes of the agglomeration graph, with an edge connecting each neighboring supervoxel pair for which the model predicted an affinity. We then perform agglomerative clustering on this graph using UPGMA linkage (Sokal et al., 1958), restricting merge operations to supervoxels that share a boundary. We determine the clustering stopping threshold on the validation set (see Appendix C). Finally, we apply a minimal cleanup that removes only components consisting of a single supervoxel and assigns their voxels to the background.

4 EXPERIMENTS

4.1 DATASETS

AC3/AC4 and SNEMI3D. The volume created by Kasthuri et al. (2015) corresponds to mouse neocortex tissue imaged with serial electron microscopy at a resolution of $6 \times 6 \times 29$ nm. AC3/AC4 are two crops taken from this volume that have shapes of $1024 \times 1024 \times 256$ and $1024 \times 1024 \times 100$ voxels, respectively. SNEMI3D is a benchmark challenge based on the AC3/AC4 volumes. In this benchmark, the myelin sheath, which is annotated in the original volumes, is relabeled as background. The training set consists of AC4, while the test set comprises the last 100 stacks of AC3. Several established methods, including FFN (Januszewski et al., 2018) and Superhuman (Lee et al., 2017), have been evaluated on this benchmark.

More recent methods are evaluated on variants of AC3 / AC4 that do not correspond to the challenge. For example, Li et al. (2026); Chen et al. (2025); Luo et al. (2024); Liu et al. (2024) use the same splitting as SNEMI3D (AC4 for training and the last 100 stacks of AC3 for testing) but consider the myelin sheath as foreground. As the datasets are not exactly the same, authors reproduced previous works such as Superhuman (Lee et al., 2017) or MALIS (Funke et al., 2019) on this new split. To be able to compare with the most recent methods, we chose this last split.

The main limitation of this dataset is its size. Notably, most benchmark datasets remain substantially smaller than the scales typically encountered in connectomic studies. Because segmentation errors accumulate over the vast image volumes required for connectomics, even a low local error rate can translate into a substantial proofreading burden at the scale of a complete dataset. Consequently, segmentation methods must not only achieve high accuracy but also maintain this performance while processing increasingly large volumes (Januszewski et al., 2018; Funke et al., 2019; Sheridan et al., 2023). To this end, we complement our analysis with the substantially larger FIB-25 dataset.

FIB-25. Takemura et al. (2015) produced a volume of Drosophila medulla imaged with FIB SEM at a $8 \times 8 \times 8$ nm resolution. The reference methods outlined in the previous section employed this dataset to demonstrate their scalability (Januszewski et al., 2018; Funke et al., 2019; Sheridan et al., 2023). The training set is composed of two volumes of $520 \times 520 \times 520$ voxels and two others of $250 \times 250 \times 250$ voxels. The test set is a $8,090 \times 6,643 \times 6,446$-voxel volume of sparse annotations composed of proofread neurons.

Evaluation on FIB-25 is not well standardized. Januszewski et al. (2018) train on one $520 \times 520 \times 520$-voxel volume and use the other to monitor training. They tune post-processing hyperparameters on the first 5,006 stacks of the test volume and evaluate on the remaining stacks, a split also used by Funke et al. (2019). In contrast, Sheridan et al. (2023) train on all four training volumes and perform agglomeration over the larger neuropil region. However, the proofread neurons used for evaluation are restricted to the region covered by the last 3,084 stacks. They tune post-processing directly on these annotations and additionally define two subregions of interest (ROIs) within this region for evaluation at different scales.

We believe these protocols present several limitations. The split of Sheridan et al. (2023) tunes agglomeration threshold on the test set, which does not reflect a realistic evaluation setting. Conversely, the protocol of Funke et al. (2019); Januszewski et al. (2018) requires a large validation set, increasing computational requirements and providing substantially more validation data than is typically available for post-processing. Training data also differ across methods: notably, Sheridan et al. (2023) compare their method with FFN while training their method on more data. We therefore introduce a split designed to enable a fairer comparison. We retain the test set of Sheridan et al. (2023) and the two sub-ROIs they defined. For training, we use the same volume as their FFN and baseline and reserve the other $520 \times 520 \times 520$-voxel volume for validation. Our method thus uses no more training data than FFN results. This smaller validation set also reduces the computational cost of reproduction, although it may disadvantage our method compared to previously reported results. Further details are provided in Appendix D.

4.2 MODEL ARCHITECTURES

The Supervoxel U-Net and Embedding U-Net are both modified "Residual Symmetric U-Nets" from Lee et al. (2021). We adopt the same architecture, replacing Instance Normalization (Ulyanov et al.,

2017) with Batch Normalization (Ioffe & Szegedy, 2015) and using a different number of filters at each stage. More precisely, we reduce the number of filters to limit model capacity and mitigate overfitting, as our training set is half the size of that used by Lee et al. (2021). In addition, we train with a larger batch size than Lee et al. (2021), which motivates our use of Batch Normalization instead of Instance Normalization. The model field of view is $96 \times 96 \times 96$ for FIB-25 and $20 \times 160 \times 160$ for AC3/AC4. On AC3/AC4, we constrain the supervoxel model to predict 2D supervoxels because of the high anisotropy. We use a supervoxel size of $8 \times 8 \times 8$ for FIB-25 and $8 \times 8 \times 1$ for AC3/AC4. For the affinity model, we use two Transformer layers with 3 heads and an embedding dimension of 192 for FIB-25 and 96 for AC3/AC4. We selected the Transformer architecture based on preliminary experiments exploring different model depths and widths. The MLP consists of one linear layer with a rectified linear unit (ReLU) activation, followed by a second linear layer with an affinity head (Sigmoid activation). We provide additional details in Appendix E.

4.3 Data processing and training details

To constitute training batches, we extract randomly located crops matching the model field-of-view shape from the training volume. We apply random rotations, flips, elastic deformations, and grayscale augmentations to these crops. Since the two datasets correspond to different imaging modalities, we use modality-specific grayscale augmentations commonly employed for these types of data (Lee et al., 2021; Sheridan et al., 2023), chosen to reflect their respective imaging characteristics and acquisition artifacts. For AC3/AC4, we simulate out-of-focus sections, missing sections, slip-type and translation misalignment, and apply contrast augmentation. For the FIB-SEM data, we apply Gaussian noise and contrast augmentation. We then standardize the crops using statistics computed from the training volume and apply connected-components with 26 connectivity to the labels. Following Lee et al. (2021), we mask out from the affinity loss objects that were split by the connected component step. To account for the effect of padding during inference, we also pad the training volumes. For AC3/AC4, we apply a $5 \times 40 \times 40$ mirror padding on each side, whereas for FIB-25 we use a $24 \times 24 \times 24$ zero padding to match the zero-valued borders of the full volume.

We train the two models jointly using the Adam optimizer (Kingma & Ba, 2015). For FIB-25, we train the models for $17,760$ iterations with a batch size of 16 and an initial learning rate of $1e{-}3$. For AC3/AC4, we train them for $8,460$ iterations with a batch size of 20 and an initial learning rate of $1e-3$. For both datasets, we use a cosine learning rate decay schedule, with the minimum learning rate set to one tenth of the initial learning rate. We retain the best checkpoint on the validation set for evaluation. We provide more details on how we set the training hyperparameters in Appendix C.

4.4 Evaluation

We use Adapted Rand Error (Arand) (Rand, 1971) and Variation of Information (VOI) (Nunez-Iglesias et al., 2013) for evaluation, following previous works. VOI can be decomposed as the sum of two terms, $\mathrm{VOI}_{\mathrm{split}}$ and $\mathrm{VOI}_{\mathrm{merge}}$, the former representing the error of segmenting one neuron as multiple fragments and the latter representing the error of merging multiple neurons into one. For both metrics, a lower value means a better segmentation. On FIB-25, Arand is not reported, so we compared our method using VOI only.

4.5 Quantitative results

AC3/AC4. We compare SNAP with leading approaches on AC3/AC4, as shown in Table 1. Baseline scores are taken from previous works (Huang et al., 2022; Luo et al., 2024; Chen et al., 2025; Li et al., 2026). We do not report FFN results, as FFN has not been reproduced on this dataset in prior work, and reproducing it would require substantial engineering effort. Our method achieves the best performance on both VOI and Arand.

FIB-25. On FIB-25, we compare with the results reported by Sheridan et al. (2023). We obtain improvements in all regions of interest. The improvement again lies in a very important reduction in split errors. This reduction is consistent with our hypothesis that affinity-derived boundary maps can lead to inevitable splits, whereas directly partitioning the volume avoids this issue (see Section 2). Also note that we tuned our agglomeration threshold only on the $520 \times 520 \times 520$ validation cube,

Table 1: Comparison of segmentation methods on AC3/AC4 using different metrics. Results of methods marked with * are reported by the respective published articles.

Method	VOI	$\text{VOI}_{\text{split}}$	$\text{VOI}_{\text{merge}}$	Arand
Superhuman* (Lee et al., 2017)	1.408	1.145	0.263	0.122
PEA* (Huang et al., 2022)	1.084	0.852	0.232	0.094
LSD* (Sheridan et al., 2023)	1.677	1.448	0.263	0.134
FragViT* (Luo et al., 2024)	1.054	0.868	0.191	0.093
AGQ* (Chen et al., 2025)	0.967	0.677	0.290	0.095
CAD + KD* (Liu et al., 2024)	0.884	0.533	0.351	0.081
FGNet* (Li et al., 2026)	0.797	0.614	**0.183**	0.069
SNAP (ours)	**0.615**	**0.364**	0.251	**0.060**

which reinforces our result given the use of a larger validation set in previously reported results (Januszewski et al., 2018; Funke et al., 2019; Sheridan et al., 2023).

Runtime comparison. We compare the runtime of our method with the affinity-based method on the FIB-25 validation set using one A100 GPU and two AMD EPYC 7543 CPUs. SNAP requires approximately 135 s per volume, compared with 105 s for the affinity-based method, corresponding to a 28.57% increase in runtime. This moderate increase in runtime is offset by the improvement in accuracy achieved by our method. Additional details are provided in H.

Table 2: Comparison of segmentation methods on the full region of interest and the two subregions of interest of FIB-25. Results of methods marked with * are reported by the cited articles. Split and Merge columns correspond to $\text{VOI}_{\text{split}}$ and $\text{VOI}_{\text{merge}}$.

	Full			ROI 1			ROI 2		
Method	VOI	Split	Merge	VOI	Split	Merge	VOI	Split	Merge
Baseline*	1.355	1.222	0.133	0.884	0.716	0.168	1.039	0.927	0.112
LR* (Lee et al., 2017)	1.867	1.603	0.257	1.357	1.089	0.268	1.489	1.207	0.282
MALIS* (Nguyen et al., 2023)	1.061	0.997	0.065	0.879	0.774	0.105	0.988	0.946	**0.042**
FFN* (Januszewski et al., 2018)	1.056	1.003	**0.053**	0.700	0.624	0.076	0.780	0.701	0.079
MTLSD* (Sheridan et al., 2023)	1.136	0.975	0.161	0.777	0.687	0.090	0.910	0.795	0.115
ACLSD* (Sheridan et al., 2023)	1.411	1.222	0.189	0.625	0.502	0.123	0.761	0.662	0.099
ACRLSD* (Sheridan et al., 2023)	1.413	1.138	0.275	0.699	0.568	0.131	0.877	0.691	0.186
SNAP (ours)	**0.700**	**0.631**	0.069	**0.444**	**0.398**	**0.046**	**0.455**	**0.398**	0.057

4.6 Qualitative results

We provide in Figure 2 a qualitative comparison between the ground truth segmentation, FFN, and our method on the first subregion of interest of FIB-25. Compared with FFN, our method produces reconstructions that more closely match the ground truth, with improved preservation of fine structures and connectivity. We also avoid important merge errors made by the FFN method. Please note that the ground truth is not perfect and, consequently, visual discrepancies between model predictions and the ground truth do not necessarily indicate that the model is incorrect. In practice, however, such discrepancies have only a limited impact on the overall evaluation due to the way the metric is computed (see Appendix F). A failure case of our method is provided in Appendix G.

4.7 Comparison with a DEL extension

We evaluate the role of the Transformer and the affinity formulation by comparing our learned affinity head with the DEL formulation (Liu et al., 2018), where affinities are computed directly from distances in feature space. Applying the DEL loss directly to the Embedding U-Net features leads to unstable training and inferior segmentation results, while applying it after the Transformer substantially improves performance. Our learned MLP affinity head on top of the Transformer achieves the best results.

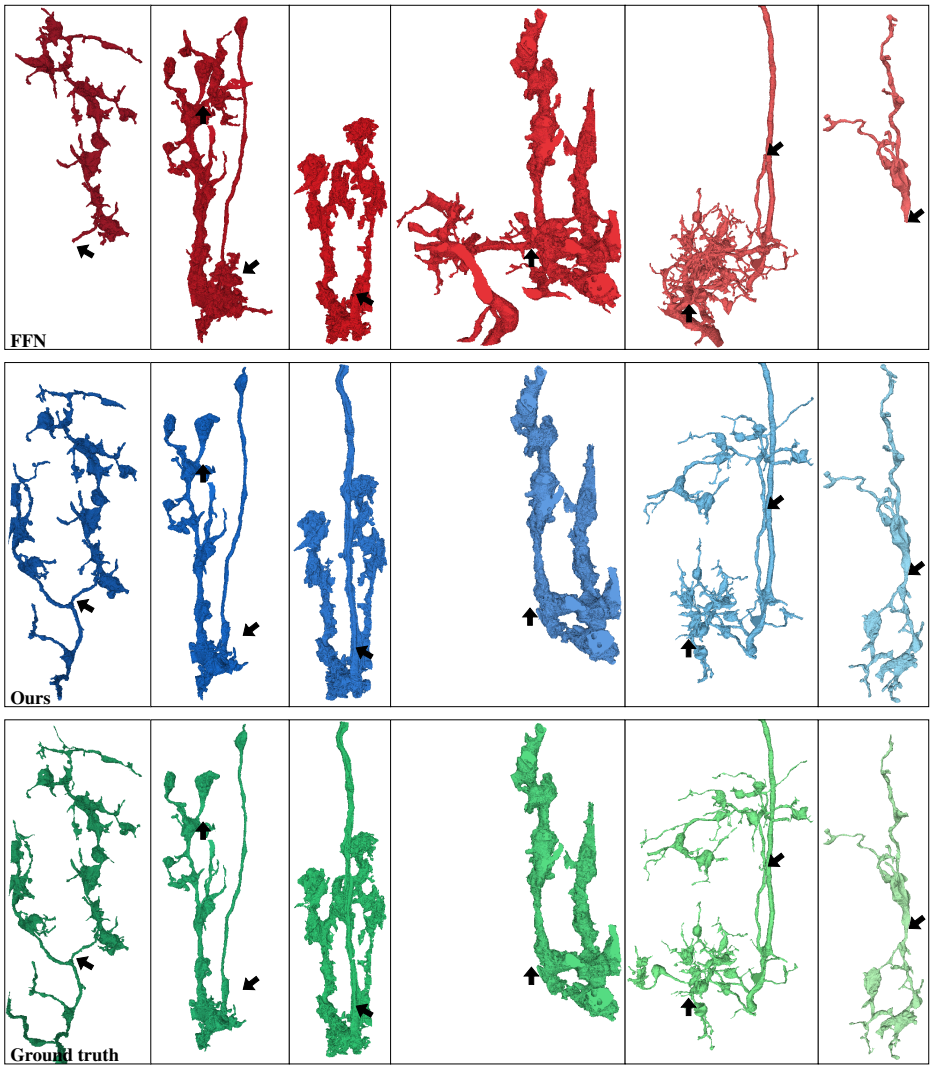


Figure 2: Illustration of segmentation differences on the first subregion of interest of FIB-25. Arrows highlight neuron reconstruction errors in FFN that are correctly recovered by our method.

Table 3: Ablation of the Transformer module and affinity prediction formulation.

Affinity formulation	VOI	$\text{VOI}_{\text{split}}$	$\text{VOI}_{\text{merge}}$
DEL affinity (U-Net embeddings)	1.734	1.186	0.549
DEL affinity (Transformer embeddings)	1.248	0.726	0.521
MLP affinity (Transformer embeddings)	**1.061**	0.561	0.501

5 Conclusion

In this work, we introduced SNAP, a novel approach to neuron segmentation based on deep supervoxel segmentation. SNAP achieves state-of-the-art performance on the AC3/AC4 and FIB-25 benchmark datasets, demonstrating both scalability and adaptability across different imaging modalities. By directly predicting supervoxels and their mutual affinities, our approach enables a simple inference procedure without the need for complex post-processing, while maintaining inference times comparable to conventional affinity-based methods. We expect this work to accelerate the process of connectome reconstruction through the reduction of the need for human proofreading.

AI USE STATEMENT

In this work, we used generative AI tools for implementing parts of the method. We did not use generative AI tools to generate research ideas, formulate hypotheses, design the methodology, interpret results, or derive scientific claims. Additionally, we used generative AI tools to improve the readability and phrasing of parts of the manuscript. We have reviewed all AI-assisted work. All LLM-generated code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the code and the weights of models trained on AC3/AC4 and FIB-25 to ensure reproducibility of our results at this Drive link.

REFERENCES

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, November 2012. ISSN 1939-3539. doi: 10.1109/TPAMI.2012.120. URL https://ieeexplore.ieee.org/document/6205760.

J. Alexander Bae, Mahaly Baptiste, Maya R. Baptiste, Caitlyn A. Bishop, Agnes L. Bodor, Derrick Brittain, Victoria Brooks, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Brendan Celii, Erick Cobos, Forrest Collman, Nuno Maçarico da Costa, Bethanny Danskin, Sven Dorkenwald, Leila Elabbady, Paul G. Fahey, Tim Fliss, Emmanouil Froudarakis, Jay Gager, Clare Gamlin, William Gray-Roncal, Akhilesh Halageri, James Hebditch, Zhen Jia, Emily Joyce, Justin Ellis-Joyce, Chris Jordan, Daniel Kapner, Nico Kemnitz, Sam Kinn, Lindsey M. Kitchell, Selden Koolman, Kai Kuehner, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Jordan Matelsky, Sarah McReynolds, Elanine Miranda, Eric Mitchell, Shanka Subhra Mondal, Merlin Moore, Shang Mu, Taliah Muhammad, Barak Nehoran, Erika Neace, Oluwaseun Ogedengbe, Christos Papadopoulos, Stelios Papadopoulos, Saumil Patel, Guadalupe Jovita Yasmin Perez Vega, Xaq Pitkow, Sergiy Popovych, Anthony Ramos, R. Clay Reid, Jacob Reimer, Patricia K. Rivlin, Victoria Rose, Zachary M. Sauter, Casey M. Schneider-Mizell, H. Sebastian Seung, Ben Silverman, William Silversmith, Amy Sterling, Fabian H. Sinz, Cameron L. Smith, Rachael Swanstrom, Shelby Suckow, Marc Takeno, Zheng H. Tan, Andreas S. Tolias, Russel Torres, Nicholas L. Turner, Edgar Y. Walker, Tianyu Wang, Adrian Wanner, Brock A. Wester, Grace Williams, Sarah Williams, Kyle Willie, Ryan Willie, William Wong, Jingpeng Wu, Chris Xu, Runzhe Yang, Dimitri Yatsenko, Fei Ye, Wenjing Yin, Rob Young, Szi-chieh Yu, Daniel Xenes, Chi Zhang, and The MICrONS Consortium. Functional connectomics spanning multiple areas of mouse visual cortex. *Nature*, 640(8058):435–447, April 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08790-w. URL https://www.nature.com/articles/s41586-025-08790-w.

Alberto Bailoni, Constantin Pape, Nathan Hutsch, Steffen Wolf, Thorsten Beier, Anna Kreshuk, and Fred A. Hamprecht. GASP, a generalized framework for agglomerative clustering of signed graphs and its application to Instance Segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11635–11645, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-66546-946-3. doi: 10.1109/CVPR52688.2022.01135. URL https://ieeexplore.ieee.org/document/9879425/.

Alexander S. Bates, Jasper S. Phelps, Minsu Kim, Helen H. Yang, Arie Matsliah, Zaki Ajabi, Eric Perlman, Kevin M. Delgado, Mohammed Abdal Monium Osman, Christopher K. Salmon, Jay Gager, Benjamin Silverman, Sophia Renauld, Farzaan Salman, Janki Patel, Matthew F. Collie, Jingxuan Fan, Diego A. Pacheco, Yunzhi Zhao, Wenyi Zhang, Laia Serratosa Capdevila, Ruairí J. V. Roberts, Eva J. Munnelly, Nina Griggs, Helen Langley, Borja Moya-Llamas, Zuoyu Zhang, Ryan T. Maloney, Szi-chieh Yu, Amy R. Sterling, Marissa Sorek, Krzysztof Kruk, Nikitas Serafetinidis, Serene Dhawan, Finja Klemm, Paul Brooks, Ellen Lesser, Jessica M. Jones, Sara E. Pierce-Lundgren, Su-Yee Lee, Yichen Luo, Andrew P. Cook, Theresa H.

McKim, Dimitrios Stasi Giakoumas, Benjamin Gorko, Justin Ellis-Joyce, Jiayi Zhang, Emily C. Kophs, Tjalda Falt, Alexa M. Negron-Morales, Austin Burke, James Hebditch, Kyle P. Willie, Ryan Willie, Sergiy Popovych, Nico Kemnitz, Dodam Ih, Kisuk Lee, Ran Lu, Akhilesh Halageri, J. Alexander Bae, Ben Jourdan, Gregory Schwartzman, Damian D. Demarest, Emily Behnke, Doug Bland, Anne Kristiansen, Jaime Skelton, Tom Stocks, Dustin Garner, Anthony Hernandez, Sandeep Kumar, Kevin C. Daly, Sven Dorkenwald, Forrest Collman, Marie P. Suver, Lisa M. Fenk, Michael J. Pankratz, Zepeng Yao, Fei Wang, Stephen J. Huston, Tomke Stürner, Gregory S. X. E. Jefferis, Katharina Eichler, Andrew M. Seeds, Stefanie Hampel, Sweta Agrawal, Tatsuo S. Okubo, Meet Zandawala, Thomas Macrina, Diane-Yayra Adjavon, Jan Funke, John C. Tuthill, Anthony Azevedo, H. Sebastian Seung, Benjamin L. de Bivort, Mala Murthy, Jan Drugowitsch, Rachel I. Wilson, and Wei-Chung Allen Lee. Distributed control circuits across a brain-and-cord connectome. *Nature*, 656(8129):957–970, August 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10735-w. URL https://www.nature.com/articles/s41586-026-10735-w.

Thorsten Beier, Constantin Pape, Nasim Rahaman, Timo Prange, Stuart Berg, Davi D. Bock, Albert Cardona, Graham W. Knott, Stephen M. Plaza, Louis K. Scheffer, Ullrich Koethe, Anna Kreshuk, and Fred A. Hamprecht. Multicut brings automated neurite segmentation closer to human performance. *Nature Methods*, 14(2):101–102, February 2017. ISSN 1548-7105. doi: 10.1038/nmeth.4151. URL https://www.nature.com/articles/nmeth.4151.

Kevin Briggman, Winfried Denk, Sebastian Seung, Moritz Helmstaedter, and Srinivas C Turaga. Maximin affinity learning of image segmentation. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://papers.nips.cc/paper_files/paper/2009/hash/68d30a9594728bc39aa24be94b319d21-Abstract.html.

Hang Chen, Chufeng Tang, Xiao Li, and Xiaolin Hu. Efficient Neuron Segmentation in Electron Microscopy by Affinity-Guided Queries. *International Conference on Learning Representations*, 2025:54394–54414, May 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/86d70cc74581ba2d110f57526ebf2473-Abstract-Conference.html.

Sven Dorkenwald, Arie Matsliah, Amy R. Sterling, Philipp Schlegel, Szi-chieh Yu, Claire E. McKellar, Albert Lin, Marta Costa, Katharina Eichler, Yijie Yin, Will Silversmith, Casey Schneider-Mizell, Chris S. Jordan, Derrick Brittain, Akhilesh Halageri, Kai Kuehner, Oluwaseun Ogedengbe, Ryan Morey, Jay Gager, Krzysztof Kruk, Eric Perlman, Runzhe Yang, David Deutsch, Doug Bland, Marissa Sorek, Ran Lu, Thomas Macrina, Kisuk Lee, J. Alexander Bae, Shang Mu, Barak Nehoran, Eric Mitchell, Sergiy Popovych, Jingpeng Wu, Zhen Jia, Manuel A. Castro, Nico Kemnitz, Dodam Ih, Alexander Shakeel Bates, Nils Eckstein, Jan Funke, Forrest Collman, Davi D. Bock, Gregory S. X. E. Jefferis, H. Sebastian Seung, and Mala Murthy. Neuronal wiring diagram of an adult brain. *Nature*, 634(8032):124–138, October 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07558-y. URL https://www.nature.com/articles/s41586-024-07558-y.

Jan Funke, Fabian Tschopp, William Grisaitis, Arlo Sheridan, Chandan Singh, Stephan Saalfeld, and Srinivas C. Turaga. Large Scale Image Segmentation with Structured Loss Based Deep Learning for Connectome Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1669–1680, July 2019. ISSN 1939-3539. doi: 10.1109/TPAMI.2018.2835450. URL https://ieeexplore.ieee.org/document/8364622/?arnumber=8364622. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Dana S. Galili, Gregory Sxe Jefferis, and Marta Costa. Connectomics and the neural basis of behaviour. *Current Opinion in Insect Science*, 54:100968, December 2022. ISSN 2214-5753. doi: 10.1016/j.cois.2022.100968.

Wei Huang, Shiyu Deng, Chang Chen, Xueyang Fu, and Zhiwei Xiong. Learning to Model Pixel-Embedded Affinity for Homogeneous Instance Segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):1007–1015, June 2022. ISSN 2374-3468. doi: 10.1609/aaai.v36i1.19984. URL https://ojs.aaai.org/index.php/AAAI/article/view/19984.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, March 2015. URL http://arxiv.org/abs/1502.03167. arXiv:1502.03167 [cs.LG].

Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel Sampling Networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, volume 11211, pp. 363–380. Springer International Publishing, Cham, 2018. ISBN 978-3-030-01233-5 978-3-030-01234-2. doi: 10.1007/978-3-030-01234-2_22. URL https://link.springer.com/10.1007/978-3-030-01234-2_22. Series Title: Lecture Notes in Computer Science.

Michał Januszewski, Jörgen Kornfeld, Peter H. Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. High-precision automated reconstruction of neurons with flood-filling networks. *Nature Methods*, 15(8):605–610, August 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0049-4. URL https://www.nature.com/articles/s41592-018-0049-4.

Michał Januszewski, Thomas Templier, Kenneth Hayworth, David Peale, and Harald Hess. Accelerating Neuron Reconstruction with PATHFINDER, May 2025. URL https://www.biorxiv.org/content/10.1101/2025.05.16.654254v1. Pages: 2025.05.16.654254 Section: New Results.

Gregory Jefferis, Lucy Collinson, Carles Bosch, Marta Costa, and Philipp Schlegel. Scaling Up Connectomics: The Road to a Whole Mouse Brain Connectome. Technical report, Wellcome Trust, 2023. URL https://wellcome.org/insights/reports/scaling-connectomics.

Jeremy Juybari, Josh Hamilton, Shuvra Das, Chaofan Chen, Andre Khalil, and Yifeng Zhu. Differentiable Laplacian Matrix Guided Superpixel Segmentation. pp. 36168–36178, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Juybari_Differentiable_Laplacian_Matrix_Guided_Superpixel_Segmentation_CVPR_2026_paper.html.

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, Mike Roberts, Josh Lyskowski Morgan, Juan Carlos Tapia, H. Sebastian Seung, William Gray Roncal, Joshua Tzvi Vogelstein, Randal Burns, Daniel Lewis Sussman, Carey Eldin Priebe, Hanspeter Pfister, and Jeff William Lichtman. Saturated Reconstruction of a Volume of Neocortex. *Cell*, 162(3):648–661, July 2015. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2015.06.054. URL https://www.cell.com/cell/abstract/S0092-8674(15)00824-7.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, 2015. URL https://arxiv.org/abs/1412.6980.

Kisuk Lee, Jonathan Zung, Peter Li, Viren Jain, and H. Sebastian Seung. Superhuman Accuracy on the SNEMI3D Connectomics Challenge, May 2017. URL http://arxiv.org/abs/1706.00120. arXiv:1706.00120 [cs].

Kisuk Lee, Ran Lu, Kyle Luther, and H. Sebastian Seung. Learning and Segmenting Dense Voxel Embeddings for 3D Neuron Reconstruction. *IEEE transactions on medical imaging*, 40(12):3801–3811, December 2021. ISSN 0278-0062. doi: 10.1109/TMI.2021.3097826. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC8692755/.

Zhenghua Li, Hang Chen, Zihao Sun, Kai Li, and Xiaolin Hu. FGNet: leveraging feature-guided attention to refine SAM2 for 3D EM neuron segmentation. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, volume 40 of *AAAI'26/IAAI'26/EAAI'26*, pp. 6744–6752. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i8.37606. URL https://doi.org/10.1609/aaai.v40i8.37606.

Xiaoyu Liu, Miaomiao Cai, Yinda Chen, Yueyi Zhang, Te Shi, Ruobing Zhang, Xuejin Chen, and Zhiwei Xiong. Cross-dimension Affinity Distillation for 3D EM Neuron Segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11104–11113, Seattle, WA, USA, June 2024. IEEE. ISBN 979-8-3503-5300-6. doi: 10.1109/CVPR52733.2024.01056. URL https://ieeexplore.ieee.org/document/10656384/.

Yun Liu, Peng-Tao Jiang, Vahan Petrosyan, Shi-Jie Li, Jiawang Bian, Le Zhang, and Ming-Ming Cheng. DEL: Deep Embedding Learning for Efficient Image Segmentation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 864–870, Stockholm, Sweden, July 2018. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-2-7. doi: 10.24963/ijcai.2018/120. URL https://www.ijcai.org/proceedings/2018/120.

Ran Lu, Aleksandar Zlateski, and H. Sebastian Seung. Large-scale image segmentation based on distributed clustering algorithms. *CoRR*, abs/2106.10795, 2021. URL https://arxiv.org/abs/2106.10795. arXiv: 2106.10795.

Aurélien Lucchi, Kevin Smith, Radhakrishna Achanta, Vincent Lepetit, and Pascal Fua. A Fully Automated Approach to Segmentation of Irregularly Shaped Cellular Structures in EM Images. In Tianzi Jiang, Nassir Navab, Josien P. W. Pluim, and Max A. Viergever (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, pp. 463–471, Berlin, Heidelberg, 2010. Springer. ISBN 978-3-642-15745-5. doi: 10.1007/978-3-642-15745-5_57.

Aurélien Lucchi, Kevin Smith, Radhakrishna Achanta, Graham Knott, and Pascal Fua. Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks With Learned Shape Features. *IEEE Transactions on Medical Imaging*, 31(2):474–486, February 2012. ISSN 1558-254X. doi: 10.1109/TMI.2011.2171705. URL https://ieeexplore.ieee.org/document/6044718.

Naisong Luo, Rui Sun, Yuwen Pan, Tianzhu Zhang, and Feng Wu. Electron Microscopy Images as Set of Fragments for Mitochondrial Segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(4):3981–3989, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i4.28191. URL https://ojs.aaai.org/index.php/AAAI/article/view/28191.

Yaron Meirovitch, Lu Mi, Hayk Saribekyan, Alexander Matveev, David Rolnick, and Nir Shavit. Cross-Classification Clustering: An Efficient Multi-Object Tracking Technique for 3-D Instance Segmentation in Connectomics. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8417–8427, Long Beach, CA, USA, June 2019. IEEE. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00862. URL https://ieeexplore.ieee.org/document/8953569/.

Joshua L Morgan and Jeff W Lichtman. Why not connectomics? *Nature methods*, 10(6):494–500, June 2013. ISSN 1548-7091. doi: 10.1038/nmeth.2480. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC4184185/.

Aljoscha Nern, Frank Loesche, Shin-ya Takemura, Laura E. Burnett, Marisa Dreher, Eyal Gruntman, Judith Hoeller, Gary B. Huang, Michał Januszewski, Nathan C. Klapoetke, Sanna Koskela, Kit D. Longden, Zhiyuan Lu, Stephan Preibisch, Wei Qiu, Edward M. Rogers, Pavithraa Seenivasan, Arthur Zhao, John Bogovic, Brandon S. Canino, Jody Clements, Michael Cook, Samantha Finley-May, Miriam A. Flynn, Imran Hameed, Alexandra M. C. Fragniere, Kenneth J. Hayworth, Gary Patrick Hopkins, Philip M. Hubbard, William T. Katz, Julie Kovalyak, Shirley A. Lauchie, Meghan Leonard, Alanna Lohff, Charli A. Maldonado, Caroline Mooney, Nneoma Okeoma, Donald J. Olbris, Christopher Ordish, Tyler Paterson, Emily M. Phillips, Tobias Pietzsch, Jennifer Rivas Salinas, Patricia K. Rivlin, Philipp Schlegel, Ashley L. Scott, Louis A. Scuderi, Satoko Takemura, Iris Talebi, Alexander Thomson, Eric T. Trautman, Lowell Umayam, Claire Walsh, John J. Walsh, C. Shan Xu, Emily A. Yakal, Tansy Yang, Ting Zhao, Jan Funke, Reed George, Harald F. Hess, Gregory S. X. E. Jefferis, Christopher Knecht, Wyatt Korff, Stephen M. Plaza, Sandro Romani, Stephan Saalfeld, Louis K. Scheffer, Stuart Berg, Gerald M. Rubin, and Michael B. Reiser. Connectome-driven neural inventory of a complete visual system. *Nature*, 641(8065):1225–1237, May 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08746-0. URL https://www.nature.com/articles/s41586-025-08746-0.

Tri M. Nguyen, Logan A. Thomas, Jeff L. Rhoades, Ilaria Ricchi, Xintong Cindy Yuan, Arlo Sheridan, David G. C. Hildebrand, Jan Funke, Wade G. Regehr, and Wei-Chung Allen Lee. Structured cerebellar connectivity supports resilient pattern separation. *Nature*, 613(7944):543–549, January 2023. ISSN 1476-4687. doi: 10.1038/s41586-022-05471-w. URL https://www.nature.com/articles/s41586-022-05471-w.

Juan Nunez-Iglesias, Ryan Kennedy, Toufiq Parag, Jianbo Shi, and Dmitri B. Chklovskii. Machine Learning of Hierarchical Clustering to Segment 2D and 3D Images. *PLOS ONE*, 8(8):e71715, August 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0071715. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0071715.

William M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971. ISSN 0162-1459. doi: 10.2307/2284239. URL https://www.jstor.org/stable/2284239.

Louis K Scheffer, C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth J Hayworth, Gary B Huang, Kazunori Shinomiya, Jeremy Maitlin-Shepard, Stuart Berg, Jody Clements, Philip M Hubbard, William T Katz, Lowell Umayam, Ting Zhao, David Ackerman, Tim Blakely, John Bogovic, Tom Dolafi, Dagmar Kainmueller, Takashi Kawase, Khaled A Khairy, Laramie Leavitt, Peter H Li, Larry Lindsey, Nicole Neubarth, Donald J Olbris, Hideo Otsuna, Eric T Trautman, Masayoshi Ito, Alexander S Bates, Jens Goldammer, Tanya Wolff, Robert Svirskas, Philipp Schlegel, Erika Neace, Christopher J Knecht, Chelsea X Alvarado, Dennis A Bailey, Samantha Ballinger, Jolanta A Borycz, Brandon S Canino, Natasha Cheatham, Michael Cook, Marisa Dreher, Octave Duclos, Bryon Eubanks, Kelli Fairbanks, Samantha Finley, Nora Forknall, Audrey Francis, Gary Patrick Hopkins, Emily M Joyce, SungJin Kim, Nicole A Kirk, Julie Kovalyak, Shirley A Lauchie, Alanna Lohff, Charli Maldonado, Emily A Manley, Sari McLin, Caroline Mooney, Miatta Ndama, Omotara Ogundeyi, Nneoma Okeoma, Christopher Ordish, Nicholas Padilla, Christopher M Patrick, Tyler Paterson, Elliott E Phillips, Emily M Phillips, Neha Rampally, Caitlin Ribeiro, Madelaine K Robertson, Jon Thomson Rymer, Sean M Ryan, Megan Sammons, Anne K Scott, Ashley L Scott, Aya Shinomiya, Claire Smith, Kelsey Smith, Natalie L Smith, Margaret A Sobeski, Alia Suleiman, Jackie Swift, Satoko Takemura, Iris Talebi, Dorota Tarnogorska, Emily Tenshaw, Temour Tokhi, John J Walsh, Tansy Yang, Jane Anne Horne, Feng Li, Ruchi Parekh, Patricia K Rivlin, Vivek Jayaraman, Marta Costa, Gregory SXE Jefferis, Kei Ito, Stephan Saalfeld, Reed George, Ian A Meinertzhagen, Gerald M Rubin, Harald F Hess, Viren Jain, and Stephen M Plaza. A connectome and analysis of the adult Drosophila central brain. *eLife*, 9:e57443, September 2020. ISSN 2050-084X. doi: 10.7554/eLife.57443. URL https://doi.org/10.7554/eLife.57443.

Alexander Shapson-Coe, Michał Januszewski, Daniel R. Berger, Art Pope, Yuelong Wu, Tim Blakely, Richard L. Schalek, Peter H. Li, Shuohong Wang, Jeremy Maitin-Shepard, Neha Karlupia, Sven Dorkenwald, Evelina Sjostedt, Laramie Leavitt, Dongil Lee, Jakob Troidl, Forrest Collman, Luke Bailey, Angerica Fitzmaurice, Rohin Kar, Benjamin Field, Hank Wu, Julian Wagner-Carena, David Aley, Joanna Lau, Zudi Lin, Donglai Wei, Hanspeter Pfister, Adi Peleg, Viren Jain, and Jeff W. Lichtman. A petavoxel fragment of human cerebral cortex reconstructed at nanoscale resolution. *Science*, 384(6696):eadk4858, May 2024. doi: 10.1126/science.adk4858. URL https://www.science.org/doi/full/10.1126/science.adk4858.

Arlo Sheridan, Tri M. Nguyen, Diptodip Deb, Wei-Chung Allen Lee, Stephan Saalfeld, Srinivas C. Turaga, Uri Manor, and Jan Funke. Local shape descriptors for neuron segmentation. *Nature Methods*, 20(2):295–303, February 2023. ISSN 1548-7105. doi: 10.1038/s41592-022-01711-z. URL https://www.nature.com/articles/s41592-022-01711-z.

Robert R. Sokal, Robert R. Sokal, and Charles D. Michener. A Statistical Method for Evaluating Systematic Relationships. *The University of Kansas science bulletin*, 38(22):1409–1438, March 1958. URL https://www.biodiversitylibrary.org/part/385476.

Rui Sun, Naisong Luo, Yuwen Pan, Huayu Mai, Tianzhu Zhang, Zhiwei Xiong, and Feng Wu. Appearance Prompt Vision Transformer for Connectome Reconstruction. volume 2, pp. 1423–1431, August 2023. doi: 10.24963/ijcai.2023/158. URL https://www.ijcai.org/proceedings/2023/158.

Fabian Svara, Dominique Förster, Fumi Kubo, Michał Januszewski, Marco dal Maschio, Philipp J. Schubert, Jörgen Kornfeld, Adrian A. Wanner, Eva Laurell, Winfried Denk, and Herwig Baier. Automated synapse-level reconstruction of neural circuits in the larval zebrafish brain. *Nature Methods*, 19(11):1357–1366, November 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01621-0. URL https://www.nature.com/articles/s41592-022-01621-0.

Shin-ya Takemura, C. Shan Xu, Zhiyuan Lu, Patricia K. Rivlin, Toufiq Parag, Donald J. Olbris, Stephen Plaza, Ting Zhao, William T. Katz, Lowell Umayam, Charlotte Weaver, Harald F. Hess, Jane Anne Horne, Juan Nunez-Iglesias, Roxanne Aniceto, Lei-Ann Chang, Shirley Lauchie, Ashley Nasca, Omotara Ogundeyi, Christopher Sigmund, Satoko Takemura, Julie Tran, Carlie Langille, Kelsey Le Lacheur, Sari McLin, Aya Shinomiya, Dmitri B. Chklovskii, Ian A. Meinertzhagen, and Louis K. Scheffer. Synaptic circuits and their variations within different columns in the visual system of Drosophila. *Proceedings of the National Academy of Sciences*, 112(44):13711–13716, November 2015. doi: 10.1073/pnas.1509820112. URL https://www.pnas.org/doi/abs/10.1073/pnas.1509820112.

J. Troidl, J. Knittel, W. Li, F. Zhan, H. Pfister, and S. Turaga. Global Neuron Shape Reasoning with Point Affinity Transformers. *bioRxiv: The Preprint Server for Biology*, pp. 2024.11.24.625067, June 2025. ISSN 2692-8205. doi: 10.1101/2024.11.24.625067.

Srinivas C. Turaga, Joseph F. Murray, Viren Jain, Fabian Roth, Moritz Helmstaedter, Kevin Briggman, Winfried Denk, and H. Sebastian Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, 22(2):511–538, February 2010. ISSN 1530-888X. doi: 10.1162/neco.2009.10-08-881.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization, November 2017. URL http://arxiv.org/abs/1607.08022. arXiv:1607.08022 [cs.CV].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Kai Wang, Liang Li, and Jiawan Zhang. End-to-end trainable network for superpixel and image segmentation. *Pattern Recognition Letters*, 140:135–142, December 2020. ISSN 0167-8655. doi: 10.1016/j.patrec.2020.09.016. URL https://www.sciencedirect.com/science/article/pii/S0167865520303421.

Yaxiong Wang, Yunchao Wei, Xueming Qian, Li Zhu, and Yi Yang. AINet: Association Implantation for Superpixel Segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7058–7067, Montreal, QC, Canada, October 2021. IEEE. ISBN 978-1-6654-2812-5. doi: 10.1109/ICCV48922.2021.00699. URL https://ieeexplore.ieee.org/document/9710587/.

Steffen Wolf, Constantin Pape, Alberto Bailoni, Nasim Rahaman, Anna Kreshuk, Ullrich Köthe, and Fred A. Hamprecht. The Mutex Watershed: Efficient, Parameter-Free Image Partitioning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, volume 11208, pp. 571–587. Springer International Publishing, Cham, 2018. ISBN 978-3-030-01224-3 978-3-030-01225-0. doi: 10.1007/978-3-030-01225-0_34. URL https://link.springer.com/10.1007/978-3-030-01225-0_34. Series Title: Lecture Notes in Computer Science.

Sen Xu, Shikui Wei, Tao Ruan, and Lixin Liao. Learning Invariant Inter-pixel Correlations for Superpixel Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6):6351–6359, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i6.28454. URL https://ojs.aaai.org/index.php/AAAI/article/view/28454.

Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel Segmentation With Fully Convolutional Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*

(CVPR), pp. 13961–13970, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-7281-7168-5. doi: 10.1109/CVPR42600.2020.01398. URL https://ieeexplore.ieee.org/document/9156320/.

A FAILURE CASE OF AFFINITY BASED BOUNDARY MAP

We show in Figure 3 the process of converting a perfect affinity map derived from the ground truth into the binary boundary map, which serves as a support for the watershed oversegmentation step (Lee et al., 2017; Funke et al., 2019; Sheridan et al., 2023). We observe that this conversion systematically splits the thinnest red object into two pieces. Consequently, during the agglomeration step, the corresponding watershed fragments are never merged. Additionally, depending on the threshold we set for the binary boundary map, if affinity predictions are not perfect, it can lead to other errors.

In Figure 4, we show that direct superpixel partitioning preserves the connectivity between the superpixels, allowing them to be merged.

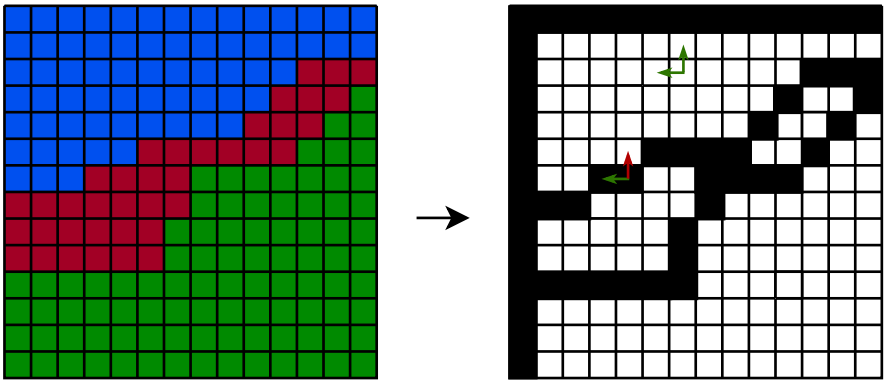


Figure 3: 2D example of the conversion of a perfect affinity map derived from the ground-truth into a binary boundary map. Left: Ground truth instance segmentation of three objects, shown in blue, red, and green. Each grid cell represents one pixel. Right: Binary boundary map obtained from affinities to the upper and left neighboring pixels. An affinity is positive when the two adjacent pixels have the same instance label; a pixel is classified as interior (white) only when the average of both affinities are superior to a predefined threshold, and is otherwise classified as boundary (black). In this example we set the threshold to 0.5: a pixel is black if one of the two affinities is zero. Arrows illustrate the evaluated neighbor directions (green: positive affinity; red: negative affinity).

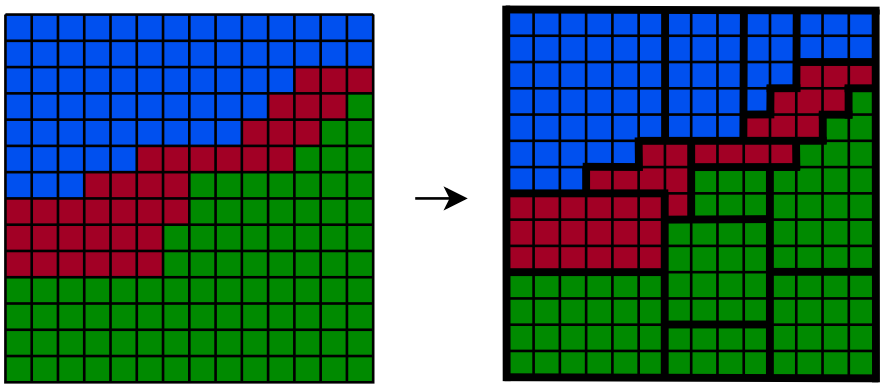


Figure 4: 2D example of a ground-truth superpixel partitioning. Left: Ground truth instance segmentation of three objects, shown in blue, red, and green. Right: A superpixel partitioning of the ground truth. Bold lines delineates the different superpixels.

B SUPERVOXEL LOSS HYPERPARAMETERS

We retain the loss calibration used in the superpixel literature and apply only deterministic normalizations required by the change from the reference 2D discretization to our 3D setting. We take the SCN configuration of Yang et al. (2020), subsequently used by Wang et al. (2021) and Juybari et al. (2026), as our reference.

Both reconstruction and compactness losses are unnormalized sums over all voxels in the volume. Consequently, to maintain the same relative weight of the different supervoxel losses, we normalize these two losses by the voxel count of the training crops we use. More precisely, let $\hat{\lambda}_{WR}$ and $\hat{\lambda}_C$ be the loss weights used by Yang et al. (2020). We define r_{N} as the ratio of the pixel count used in their work to our voxel count:

$$r_{\mathrm{N}} = \frac{208^2}{N}.$$

Additionally, the raw compactness loss is a spatial sum of unsquared Euclidean coordinate errors and therefore scales both with the number of voxels and with the characteristic linear extent of a supervoxel. In the superpixel literature, this dependence on spatial scale is accounted for by weighting the compactness term according to m/S (Yang et al., 2020; Wang et al., 2021; Juybari et al., 2026), with $m = 0.003$ and $S = 16$ in the reference 2D configuration. We extend this convention to our anisotropic 3D setting by defining the effective stride

$$S_{\mathrm{eff}} = \frac{\sqrt{s_z^2 + s_y^2 + s_x^2}}{\sqrt{2}},$$

where $\mathbf{s} = (s_z, s_y, s_x)$ denotes the supervoxel stride. This definition recovers $S_{\mathrm{eff}} = 16$ for the reference two-dimensional stride $(16, 16)$. Relative to this reference configuration, the corresponding geometric correction is therefore

$$r_{\mathrm{geom}} = \frac{16}{S_{\mathrm{eff}}} = \frac{16\sqrt{2}}{\sqrt{s_z^2 + s_y^2 + s_x^2}}.$$

This approximately preserves the contribution of the compactness term when changing the dimensionality or grid stride. The effective loss coefficients we use are then

$$\lambda_{WR} = r_{\mathrm{N}} \hat{\lambda}_{WR}, \qquad \lambda_C = r_{\mathrm{geom}}\, r_{\mathrm{N}} \hat{\lambda}_C.$$

For FIB-25, this gives $(r_{\mathrm{N}}, S_{\mathrm{eff}}) \approx (0.049, 9.80)$, while for AC3/AC4 we obtain $(r_{\mathrm{N}}, S_{\mathrm{eff}}) \approx (0.084, 8.03)$. The remaining two losses are already independent of crop size in their definition, so we directly use the same coefficients.

C GRID SEARCH

We perform a grid search to tune the training hyperparameters on both datasets. The learning rate, the Laplacian loss weight λ_{LAP}, the total weight of supervoxel loss $\lambda_{Supervoxel}$, and the hierarchical agglomeration threshold are varied during the search. Table 4 summarizes the explored parameter ranges and indicates the selected hyperparameter values in bold. All training runs are performed on two A100 GPUs, each with 80 GB of GPU memory.

Table 4: Grid search range for FIB-25 and AC3/AC4

Dataset	Learning rate	λ_{LAP}	$\lambda_{Supervoxel}$	Threshold
AC3/AC4	$\{1e-4, 2e-4, 5e-4, \mathbf{1e-3}, 2e-3, 5e-3\}$	$\{0, \mathbf{360}, 1000, 3600\}$	$\{0.1, 0.5, \mathbf{1.0}, 2.0, 10.0\}$	$\{\mathbf{0.3}, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
FIB-25	$\{1e-4, 2e-4, 5e-4, \mathbf{1e-3}, 2e-3, 5e-3\}$	$\{0, \mathbf{360}, 1000, 3600\}$	$\{0.1, 0.5, \mathbf{1.0}, 2.0, 10.0\}$	$\{0.3, 0.4, \mathbf{0.5}, 0.6, 0.7, 0.8, 0.9\}$

D FIB-25 DATASET EVALUATION PROTOCOL

FIB-25 full volume is $8,090 \times 6,643 \times 6,446$ voxels large with an irregular mask to exclude cell bodies from predictions (after masking, 10% of the full volume remains). To make the prediction on the full volume, we skip fully masked patches, and run the model normally on the rest of the volume. Aggregation is performed on the resulting graph, using the threshold found on the validation set. Following previous work (Januszewski et al., 2018), we dilate the final segmentation so that no background voxels remain. For subregions of interest, we crop the full-volume prediction and run connected-component relabelling. Please note that this is not equivalent to running agglomerative clustering on the cropped prediction. Because applying the full pipeline to an entire volume requires additional engineering effort, we also report results obtained by applying it directly to cropped regions of interest (Table 5). This allows others to compare future methods with ours without having to process the full volume.

Table 5: Segmentation results on the two subregions of interest of FIB-25, with the full pipeline applied directly to the crops.

	ROI 1			ROI 2		
Method	VOI	$\text{VOI}_{\text{split}}$	$\text{VOI}_{\text{merge}}$	VOI	$\text{VOI}_{\text{split}}$	$\text{VOI}_{\text{merge}}$
SNAP (ours)	0.492	0.439	0.054	0.473	0.406	0.067

E MODEL ARCHITECTURES

The two U-Nets follow the architecture proposed by Lee et al. (2021). Each convolutional layer consists of a ReLU activation, batch normalization, and a $3 \times 3 \times 3$ convolution. Each block comprises four convolutional layers, with a residual connection that adds the output of the first convolutional layer to the output of the third convolutional layer. The initial and final convolutional layers of the U-Net use $5 \times 5 \times 5$ kernels. For AC3/AC4, we replace these kernels with $5 \times 5 \times 1$ kernels. The U-Nets consist of five stages, with the number of filters increasing by 16 at each stage. We employ 16 filters for the first layer of the supervoxel model and 36 for the embedding model. We linearly project the output of the embedding model to the Transformer embedding dimension and add a 3D sinusoidal positional encoding. We compute supervoxel positions using voxel coordinates (see Section 3). We rescale the voxel coordinates according to the volume's anisotropy, thereby accounting for unequal voxel spacing along the spatial axes and providing the Transformer with a more accurate representation of each supervoxel's position.

F DATASET DEFECTS

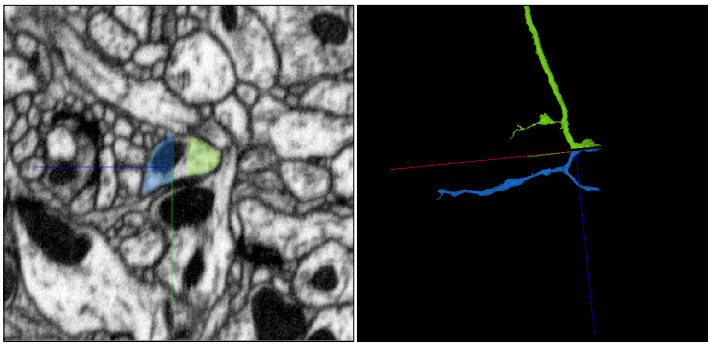


Figure 5: False split in the first subregion of interest of FIB-25. If models segment this neuron correctly, it is counted as a merge error. This type of error in the ground truth is rare.

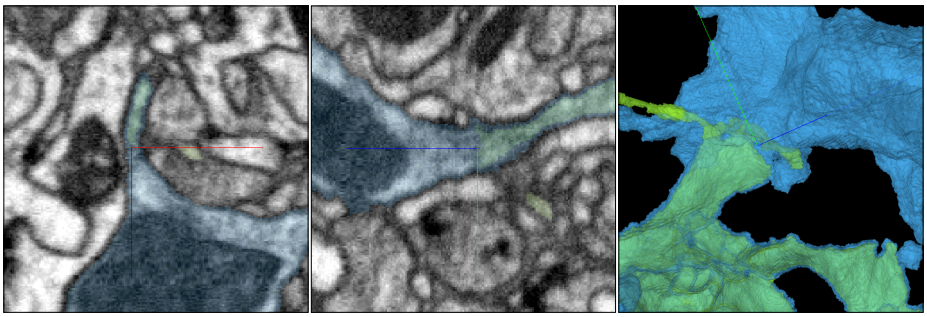


Figure 6: A neuron in the ground truth that is not completely reconstructed in the first subregion of interest of FIB-25. In green the ground truth, and in blue the prediction of SNAP. VOI excludes background voxels in the ground truth annotation, implying that a model that correctly segments this neuron will not be penalized.

G VISUAL INSPECTION ON FULL VOLUME

We provide a visualization of randomly selected neurons from the full volume segmentation. We observe that, in general, our method produces clean neuron reconstructions (see Figure 7). Our method can struggle with glial processes, mixing them with neurons due to thin and ambiguous boundaries. An example can be seen in Figure 8: a glial fragment merges two neurons together. We believe that incorporating classes into the training set could reduce these kinds of errors.

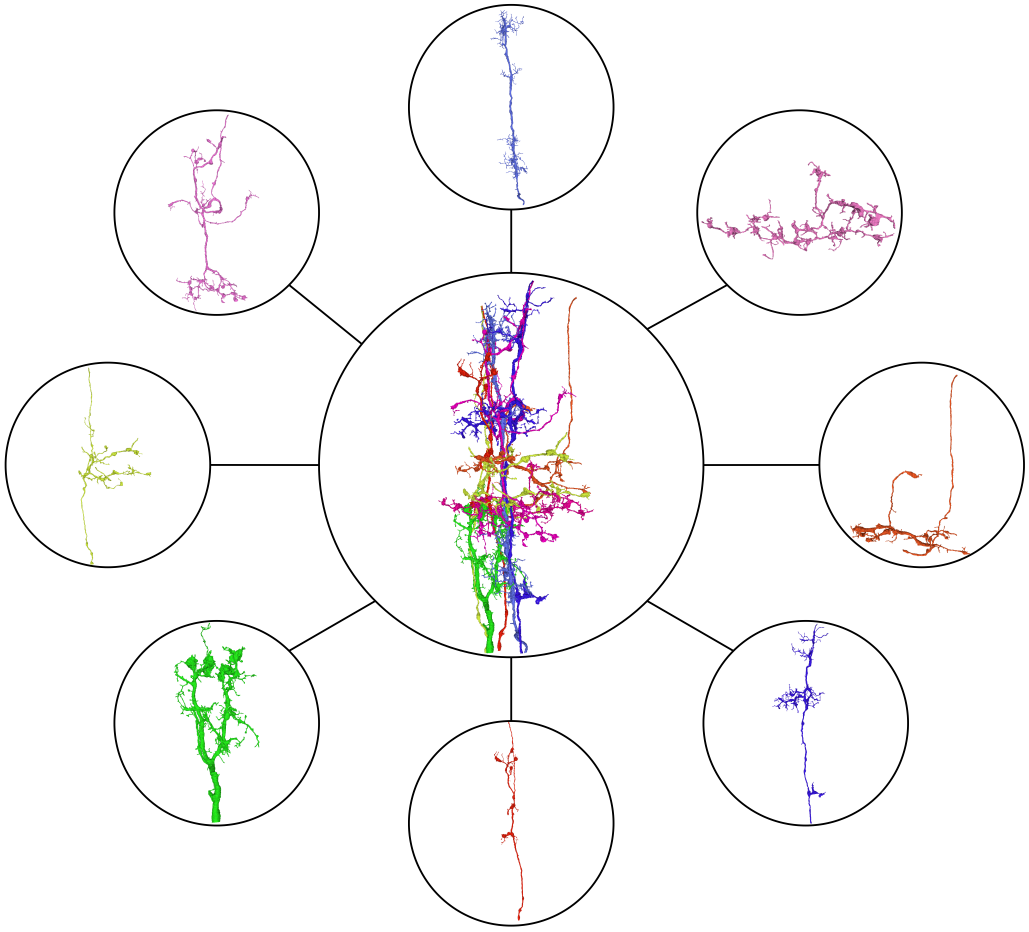


Figure 7: Qualitative inspection of SNAP segmentation on eight selected neurons from the full region of interest.

G.1 Failure mode

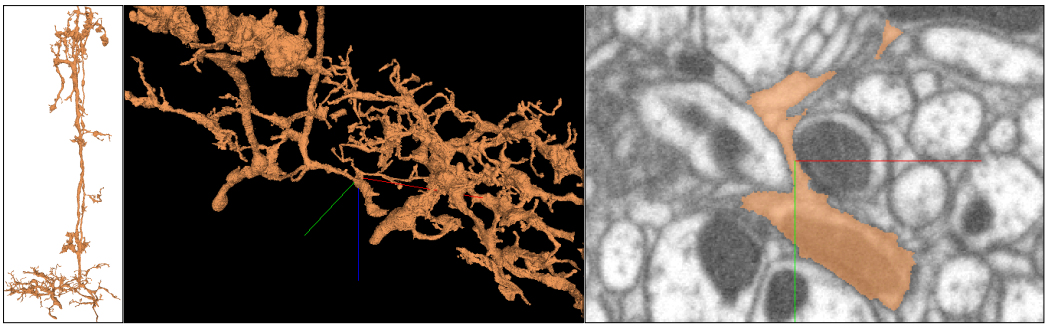


Figure 8: Failure mode of SNAP. A glial fragment merges two neurons together.

H Supplementary information on runtime experiment

For a fair runtime comparison, we implement the affinity-based baseline using the same U-Net architecture as our method and apply the same overlapping prediction strategy during inference. Thus, we ensure differences arise from the respective methods.

Table 6: Runtime breakdown on the FIB-25 validation set.

Method	Model + Finalization	Post-processing	Agglomeration	Total
Affinity-based	68.41 s	29.10 s	6.84 s	104.35 s
SNAP (ours)	127.39 s	–	8.50 s	135.89 s

In practice, the post-processing operations of affinity-based methods can be performed block-wise, allowing them to be efficiently parallelized. We follow this strategy and process the predicted affinities using blocks of size $128 \times 128 \times 128$ with a halo of $24 \times 24 \times 24$ voxels. Block-wise post-processing is parallelized over the 64 CPU cores available on our machine. The final agglomeration step is performed on the full volume. We use the `waterz` library for the affinity-based method and similarly perform our supervoxel agglomeration globally. Although agglomeration could in principle also be parallelized (Lu et al., 2021), doing so would require additional engineering for both methods; we therefore retain full-volume agglomeration to ensure a comparable implementation setting.

Under these conditions, on the FIB-25 validation set and using one NVIDIA A100 GPU and two AMD EPYC 7543 CPUs, SNAP requires approximately 135 s to segment a volume, compared with 105 s for the affinity-based method. This corresponds to a 28.57% increase in runtime, which we consider a moderate computational overhead given the improvement in segmentation performance. We additionally report a breakdown of the runtime in Table 6, measuring separately (i) model prediction and finalization (argmax computation on whole volume, Hann window normalization, etc.), (ii) post-processing, and (iii) agglomerative clustering.