nature methods

SUPPLEMENTARY INFORMATION

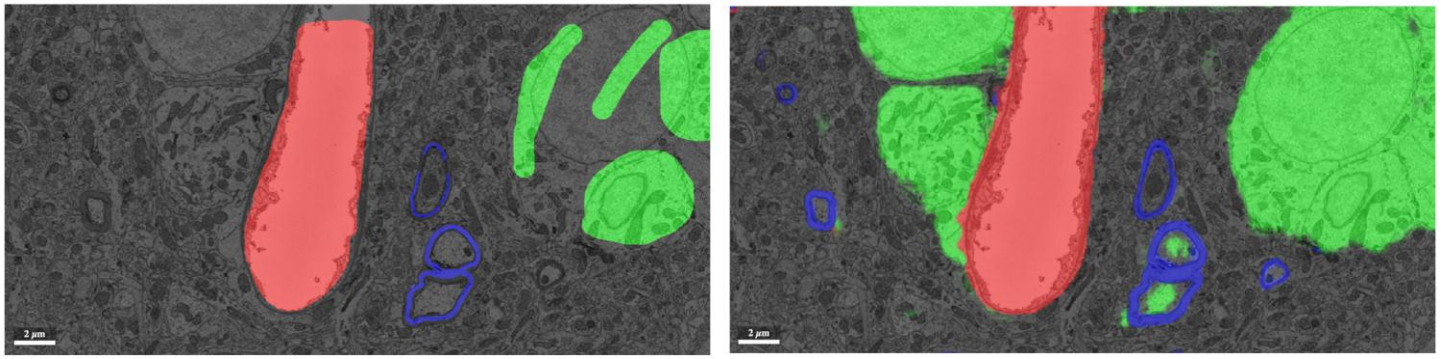
https://doi.org/10.1038/s41592-018-0049-4

In the format provided by the authors and unedited.

High-precision automated reconstruction of neurons with flood-filling networks

Michał Januszewski[1], Jörgen Kornfeld[2], Peter H. Li[3], Art Pope[3], Tim Blakely[4], Larry Lindsey[4], Jeremy Maitin-Shepard[3], Mike Tyka[4], Winfried Denk[2] and Viren Jain[3]*

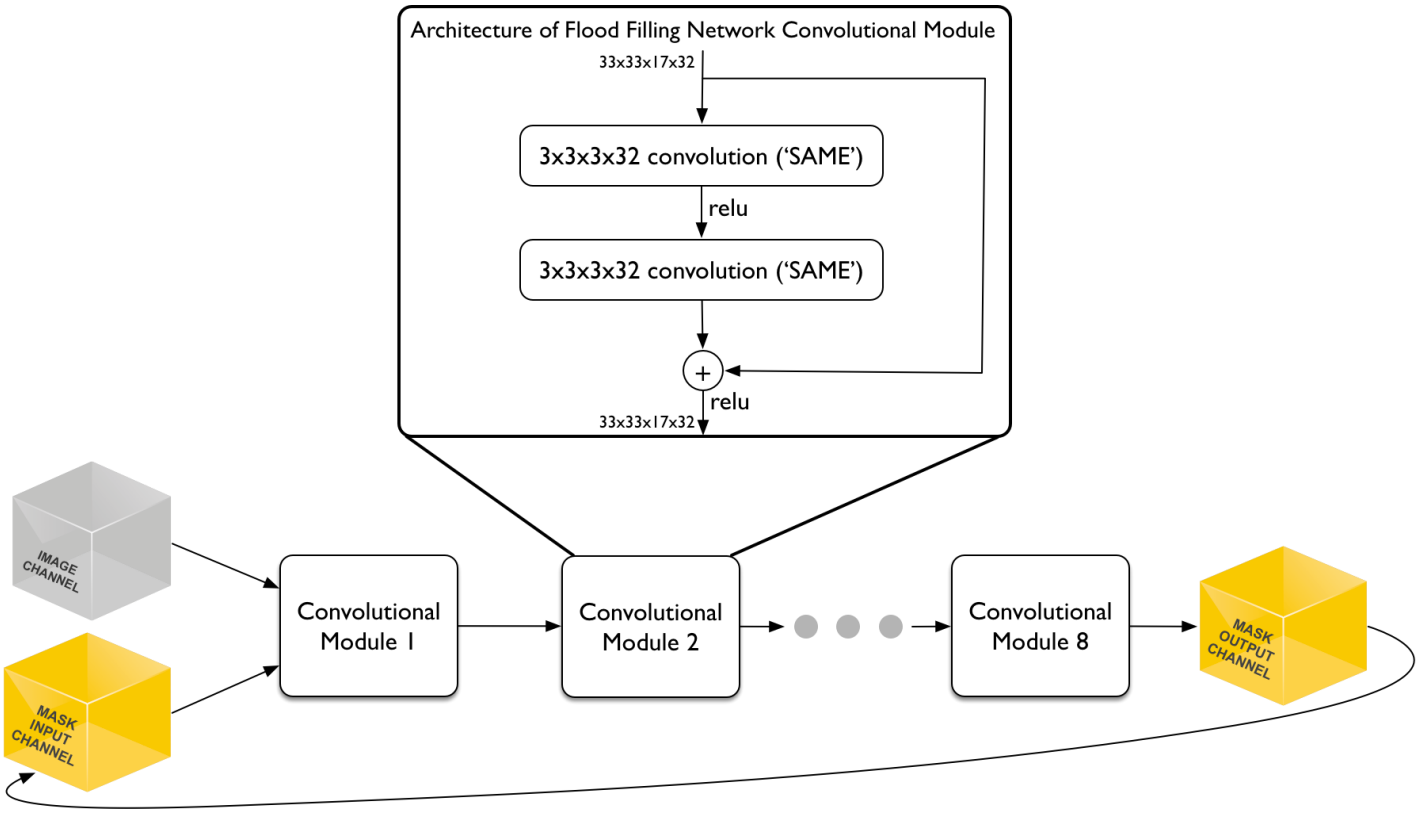
[1]Google AI, Zürich, Switzerland. [2]Max Planck Institute of Neurobiology, Planegg, Martinsried, Germany. [3]Google AI, Mountain View, CA, USA. [4]Google AI, Seattle, WA, USA. *e-mail: viren@google.com



Supplementary Figure 1

Tissue classification results.

Manual annotations (left) and convolutional network inference (right) of a subset of the labeled voxel classes: blood vessel (red), myelin (blue), and cell body (green). False positive identifications of cell body voxels are visible in the automated inference (inside the myelinated area). Scale bar is 2 microns.



Supplementary Figure 2

Architecture of the FFN.

Overall computational architecture of the Flood-filling Network. Each of the eight convolutional modules are identical and implement the operations shown in the top inset box. The predicted object map (POM) is shown as the yellow mask channel, and provides recurrent feedback to the FFN.

High-Precision Automated Reconstruction of Neurons with Flood-filling Networks

Supplementary Notes

Elastic alignment of the EM stack:

The raw EM sections were first translationally aligned by computing a globally optimal shift correction for every section based on cross-correlation of neighboring sections.

We then elastically aligned the dataset using a procedure based on Saalfeld et al [38], but with several modifications. First, block matching locations were decoupled from the elastic mesh vertices (block matches were searched for on a 200-pixel grid using normalized cross-correlation and a mesh with 500-pixel edges was used). Second, for each block match, a spiral of offsets around each grid location was analyzed until a match was found where the normalized cross correlation surface (a) attained a local maximum exceeding 0.3, (b) exceeded by 10% any other maximum within its 111x111 pixel neighborhood (avoiding ambiguous correlation peaks), and (c) had a ratio of principal curvatures of no more than 10 (avoiding elongated peaks with uncertain locations), which we found to improve results in regions with poor or ambiguous texture. Finally, a conjugate gradient solver was used to relax the mesh, which we found to be less sensitive to integration step size and spring stiffness than Euler's method, and which resulted in overall faster convergence.

Elastic alignment used 12 million patch match correspondences (i.e., tiepoints) between adjacent sections, and an additional 12 million between pairs of sections that were separated by an intervening section.

In the translationally aligned volume provided as input to elastic alignment, the tiepoints between adjacent sections had a mean residual error of 1.8 pixels; the 95th percentile error was 3.0 pixels. For the 12 million tiepoints across non-adjacent sections, the means and 95th %ile errors were 2.8 and 6.0 pixels.

After elastic alignment, these stats were 1.6 pixels (mean) and 2.0 pixels (95th %ile) between adjacent sections, and 2.6 and 5.0 pixels between non-adjacent sections. The mean of the magnitude of the displacement of the 3.3 million nodes of the elastic meshes used to model each section was 0.6 pixels.

Precision/recall estimation:

To estimate recall and precision of a single human annotator and the automated reconstruction (FFN-c) we used manually generated and proofread skeletons (see "Ground truth verification" below for details) and dendritic spine head/base location annotations. The skeletons were manually separated (split) and classified as axons or dendritic branches.

Dendritic spine recall was measured by comparing the segmentation at two manually placed locations, one at the base, the other at the head of a spine. If the segmentation labels at base and head were different, a false negative was counted. A spine was counted as found by a human-generated skeleton only if there were two disjoint sets of skeleton nodes that each contained at least one skeleton node and all of the nodes in the first set were within 250 nm of the base and correspondingly for the second set and the head location.

For dendrites and axons, single-annotator skeleton nodes were matched to the nodes of the ground truth skeletons within a radius of 800 nm. The matched path length was counted as true positive, the unmatched path length in the ground truth skeleton as false negative, and the unmatched path length in the single-annotator skeleton as false positive. For a segmentation, we computed precision and recall for every segment overlapping the ground truth skeletons, with the path length of the fragment of the ground truth skeleton overlapping the segment as true positive, and the path length of the automatically generated skeleton of a segment detected as merged and not matched with the ground truth skeleton as false positive. We computed recall and precision as a weighted sum of the per-segment recall and precision, with a weight of (path length of the fragment of the ground truth skeleton overlapping the segment) / (total path length of the ground truth skeleton). This estimates the expected value of precision/recall provided that segmentation is started from a random location on the ground truth skeleton.

Skeleton edge accuracy classification:

Segmentation quality is often evaluated with respect to ground truth pixel-wise labels or object masks [25], but creating such ground truth for large-scale EM datasets that span billions or trillions of voxels is highly laborious. A more efficient way to generate ground truth representations of large-scale neuron topology is to “skeletonize” neurons into a collection of points and their connections, which typically constitute an undirected tree [9].

We used a set of metrics to evaluate a segmentation on the basis of such skeletons. Similar to previous approaches, we classified individual edges in skeletons as *correctly* or *incorrectly* reconstructed based on the presence of mergers or splits that affect nodes attached to an edge ([12,40]). We assume:

- a ground-truth skeleton S_i consists of edges $\{e_1, e_2, \ldots, e_{|S_i|}\}$,
- an edge e is defined by two 3-d node coordinates $A(e)$ and $B(e)$,
- $S(e)$ denotes the ID of the ground-truth skeleton containing edge e,
- R denotes a predicted segmentation to be evaluated, and $R(p)$ returns the value (object ID) at point p. $R(e)$ denotes either $R(A(e))$ or $R(B(e))$ where this is unambiguous.

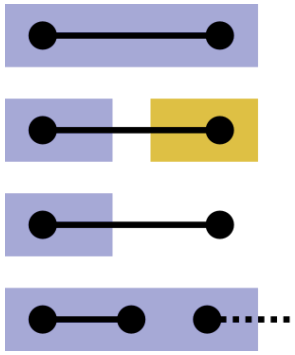
An edge e is defined as *correctly reconstructed* if both of its nodes belong to the same object in the reconstruction and if that object does not contain any nodes from different skeletons. More formally, we classified every edge e into one of four categories (see Supp. Note Fig. 1):

- *omitted* if $R(e) = 0$
- *split* if $R(A(e)) \neq R(B(e))$,

- *merged* if there exists an edge e_m such that $R(e) = R(e_m)$ but $S(e) \neq S(e_m)$,
- *correct* if none of the above is true.

The *edge accuracy* is the percentage of correctly reconstructed edges over all the ground truth skeletons, and incorrect edges can be further subdivided into the percentage of edges which have a merge, split, or omitted errors.

The definition of a merged node assumes that there is a skeleton for every object in the volume of interest. Some mergers could remain undetected when this assumption is violated, which is the case for large volumes where it is impractical to skeletonize every object manually. To mitigate this, we applied an additional merge-detection heuristic, which considers a segment *T* merged if there exists a point *p* where *R(p) = T* and *p* is more than *2.2* µm away from any skeleton node lying within *T*. The distance threshold was chosen based on the size of the neurites in the J0126 dataset and edge lengths in the ground truth skeletons. The merge detection heuristic was not applied in the vicinity of the cell body associated with the ground truth skeleton (when present in the dataset).



Supplementary Note Figure 1. Edge classes for skeleton accuracy computation. Colors correspond to segment IDs. From top to bottom: correct edge (both nodes have the same ID), split edge (nodes assigned to different segments), omitted edge (one or two nodes do not have an associated ID), merged edge (node assigned to a segment that covers more than one skeleton).

Expected Run Length:

In order to evaluate automated segmentation results, a metric is needed that compares volumetric 3d components to ground truth skeletons and computes a single score or run length. We used the "expected run length" (ERL), which is defined as follows.

For a given skeleton S, let *CE(S)* denote the set of correct edges (as defined above). We would like to partition this set into "correctly reconstructed components" (CRCs) -- subsets of edges corresponding to valid (without a merger) segments in *R*. By definition the set $\{R(e): e \in CE(S)\}$ contains only such segments.

We therefore partition *CE(S)* by the segment label *L* and define a correct component as:

CRC(S, L) = {e: e ∈ CE(S) and *R(e) = L}*.

The expected run length (ERL) is the expected size of the correctly reconstructed component, assuming tracing starts from a random point on the skeleton:

$$ERL(S) = \sum_{L} ||CRC(S, L)|| \cdot \frac{||CRC(S, L)||}{||S||}$$

where the skeleton size is $||S|| = \sum_{e \in S} ||e||$.

Note in particular that under this definition starting from a point which belongs to an incorrect edge (i.e. omitted, split, or merged) does not allow us to trace any correct path length and therefore does not contribute to the ERL. The ERL corresponds to the average segment length if the average is taken with the number of skeleton nodes in each segment as its weight.

The ERL for a set of skeletons $\{S_k\}$ is defined as:

$$ERL(\{S_k\}) = \sum_{k} w_k \cdot ERL(S_k)$$

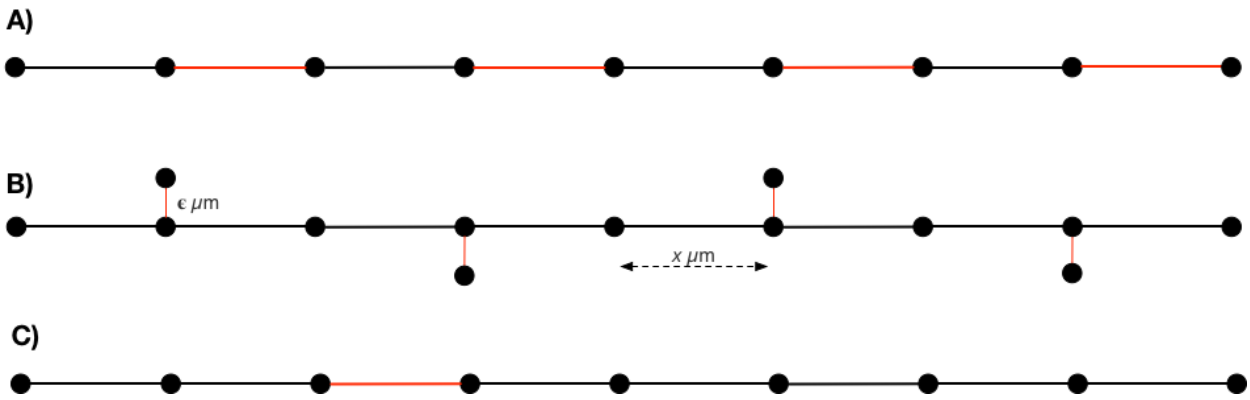
$$w_k = ||S_k|| \,/\, \sum_{i} ||S_i|| \,.$$

Expected Run Length compared to other metrics:

In contrast to prior approaches, the ERL takes into account the spatial distribution of errors. Previously proposed metrics, such as the total error-free path length (TEFPL) [12,40] and inter-error distance (IED) [12] are defined as simple averages and are thus insensitive to the distribution of lengths of the correctly reconstructed fragments (see Supp. Note Fig. 2 for an illustration).

Berning et al. define the number of splits and mergers in a predicted dense segmentation with respect to a set of ground truth skeletons as follows [12]:

1. An individual skeleton is said to *correspond to* a predicted segment if at least *k* nodes within the skeleton are contained within the predicted segment, for *k* = 1 or 2. Larger values of *k* provide robustness to the imprecise placement of skeleton nodes.
2. For each skeleton, the number of splits is determined as max(0, number of corresponding predicted segments - 1).
3. For each predicted segment, the number of mergers is determined as max(0, number of corresponding skeletons - 1).



Supplementary Note Figure 2. A) An idealized skeleton fragment representing *8x* μm length of neuropil where alternating edges (red) have been identified as incorrectly reconstructed in some candidate segmentation. Total error free path length (TEFPL) proposed by Pallotto et al computes an accuracy of 50% by dividing correct edge path length by total path length (4 μm / 8 μm), whereas the expected run length (ERL) for this fragment is *x* μm. The inter-error distance (IED) of Berning et al. is also x μm. B) An idealized skeleton fragment with "spines" of length ϵ; as $\epsilon \to 0$, the ERL and TEFPL converge to *8x*, while the IED converges to 2 μm (8 μm / 4) C) Single split dividing the process into two segments of unequal length. TEFPL is 7 μm, IED is 4 μm and ERL is 3.625 μm.

Berning et al's definitions result in a metric with different properties from the one we propose:

- In their metric, erroneously connecting two segments can **decrease** the total number of mergers; in the limit case of a single predicted component encompassing the entire volume, the number of mergers is (number of skeletons - 1).
- In their metric, the spatial distribution of splits and mergers is not taken into account: splitting off a single synapse is a single error, as is splitting a neuron in half.

In addition to being sensitive to the spatial distribution of errors, the ERL penalizes mergers very heavily since *all* edges covering the merged segments are considered erroneous, and edges marked incorrect do not contribute to expected run length. We argue that this a desired property of the metric, reflecting the significantly higher effort required to fix such errors manually during proofreading of the automated results.

Ground truth verification:

We found that the initial ground truth skeletons used for segmentation evaluation contained a substantial number of errors, despite being created by at least twofold redundant skeletonization ([9–11]) with manual resolution of discrepancies. To mitigate this, two of the authors (M.J., J.K.) inspected every discrepancy in each of the segmentations listed in Fig. 3, and when both agreed that a mistake was made in the skeletonization rather than by the FFN segmentation, the skeletons were updated. The procedure was repeated until no more error cases were detected. See Sup. Note Table 1 for a summary of the changes made.

Note that because of the merge detection heuristic we use, splits in the ground truth skeletons (e.g. missing neurite branches) are treated as mergers in the segmentation, and segments detected as merged do not contribute to the ERL. The quality of the ground truth data becomes increasingly more important as the path length of correctly reconstructed components improves. As an edge case, consider a complete correctly reconstructed cell with all its branches, and a single spine missing in the corresponding ground truth skeleton. This would cause the whole reconstructed object to be considered merged, and the ERL would be 0 µm.

Because of the distance threshold used in the merge heuristic (2.2 µm), a number of smaller spines missing in the skeletons remain uncorrected. The number of missing spines in Sup. Table 1 should therefore be treated as a lower bound.

Fragment type	**Splits (untraced fragments)**		**Mergers (mistraced fragments)**	
	Number of errors	**Affected skeletons**	**Number of errors**	**Affected skeletons**
dendritic spine	59	13	0	0
axon	4	2	5	4
dendrite	2	1	1	1
cilium	1	1	0	0
glia	n/a	n/a	2	2

Supplementary Note Table 1. Errors identified in the ground truth verification process. In total, 665 µm of skeleton path length was added (to fix splits), and 166 µm was removed (to fix mergers). Additionally, 5 skeleton nodes were moved due to their initial placement outside of the neurite represented by the skeleton.

Segmentation parameters:

The neural network used in the baseline CNN method had the following architecture: 2d convolution (VALID mode, 5x5 filter size, 64 output features), 3d convolution (VALID mode, 5x5x5 filter size, 64 output features), 2d pooling (SAME mode, 2x2 stride, 2x2 filter size), 3d convolution (VALID mode, 5x5x5 filter size, 64 output features), 2d pooling (SAME mode, 2x2 stride, 2x2 filter size), 2d convolution (VALID mode, 5x5 filter size, 512 output features), pointwise convolution (147 output features). The output features of the last layer were treated as a 7x7x3 long-range affinity graph. The FOV of the CNN was 35x35x9.

To see if a larger FOV could improve results, we also evaluated a recursive CNN [39] that was trained to predict a boundary map. The approach used two convolution-pooling networks: bar_b with a 111x111x13 FOV, and bar_{br} with a 91x91x9 FOV (see Supplementary Table 3 of [39] for the detailed architecture of both networks). The first network (bar_b) took the image as input and predicted a boundary map. The second network (bar_{br}) took the image and the predictions of the first network, and predicted an updated boundary map. We used the output of the bar_{br} network, which had an effective FOV of 201x201x21, and performed a grid search for watershed parameters, optimizing for ERL computed within the densely skeletonized subvolume. The best ERL found this way was 0.7 μm, which was less than the baseline CNN. We therefore decided to exclude this network from further experiments.

SegEM, as well as all FFN segmentations, used the elastically aligned volume. CNN and CNN+GALA used the original volume which was only translationally aligned, as these methods applied to the elastically aligned volume showed worse performance as measured by the set of 12 skeletons used for hyperparameter tuning. Myelin, OOB, and blood vessels were masked out in all segmentations -- voxels classified into one of these three categories were set to 0 (background).

For SegEM, the best performing set of parameters was found to be: r=0 (no morphological filtering) and h=0.045. For CNN, the optimal parameters found were T_h=0.945, T_l=0.945, T_e=0.5, T_s=1000). For CNN+GALA, the agglomeration threshold was set at 0.9.

SegEM metrics:

In Supplementary Note Table 2 we present SegEM metrics for the segmentations discussed in Fig. 3. For the SegEM segmentation method, we found the inter-error distances to be consistent with those previously reported (optimal IED = 5.5 μm in the present work, and 7.9 μm and 4.9 μm for the two datasets discussed in the original paper[12]).

Segmentation	IED (split) [μm]	IED (merge) [μm]	IED ("optimal") [μm]
SegEM	7	36	5.5
CNN	2	497	2.0
CNN+GALA	10	171	9.5
FFN-a	30	594	28.5
FFN-b	14	48,670	13.5
FFN-c	33	23,335	33.0

Supplementary Note Table 2. Full-volume evaluation of segmentation quality with SegEM metrics and 50 ground truth skeletons. The merge detection heuristic discussed in "Skeleton

Edge Accuracy Classification" has been applied to detect mergers. Segments detected as merged were counted as a single merge error. Half of the harmonic mean of IED split and IED merge was used to compute the optimal IED.

Breakdown and comparison of computational cost:
On average, every voxel of the volume was processed by the FFN 59 times in a segmentation run. Ensembling segmentations from multiple seed points further multiplies FFN inference cost by 2.38x, and agglomeration introduces another factor of 1.63x. In total, the FFN pipeline required 14.4 x 2.38 x 1.63 = 56x greater computation compared to the baseline CNN.

FFN inference run	**FFN inference calls [x 10^9]**	**EFLOPs**	**Wall time with 1000 Tesla P100 GPUs [h]**
9x9x20 nm, forward	2.22	41.05	1.79
9x9x20 nm, backward	1.98	36.69	1.60
18x18x20 nm, forward	0.55	10.18	0.44
18x18x20 nm, backward	0.48	8.89	0.39
36x36x40 nm	0.05	1.00	0.04
agglomeration	3.33	61.50	2.69
total	8.62	159.32	6.964

Supplementary Note Table 3. Computational cost of FFN inference. No pre- and post-processing of the data is taken into account in the calculations. The wall-clock time is an empirical measurement based on average inference speed with single precision floating point numbers, on a single NVIDIA Tesla P100 GPU with TensorFlow, and using CuDNN v6 as the computational backend.

For comparison of segmentation cost between the FFN pipeline and the baseline CNN, we assumed that CNN inference was done in a distributed setting with overlapping subvolumes of size 182x182x158 (the size of the subvolume was limited by the need to store the intermediate feature maps in GPU memory).

Local versus Global Evaluations:
In our experiments, we have repeatedly found that local evaluations using small (order of hundreds of μm^3) subvolumes of data underestimate error rates. Similar observations were

made in the context of synapse prediction in [39]. In Supp. Note Table 3 we provide evaluations of the segmentations from Fig. 3 over the 5 µm x 5 µm x 5 µm densely skeletonized subvolume (with connected components of the segmentation recomputed after restricting it to the subvolume). The total skeletonized path length is 1 mm (20% of which is glial), and the maximum possible ERL is 13.4 µm.

Note that if two objects were merged outside of the subvolume, but were directly adjacent to each other within the subvolume, recalculation of connected components would not allow them to be split. The number of merges is therefore overestimated compared to what it would be if the segmentation procedure was restricted to the subvolume from the beginning (cf. FFN-c in Supp. Note Table 4 with and "recurrent single object (FFN)" in Supp. Note Table 6).

Segmentation	ERL [µm]	Edge accuracy	Merged edge fraction[1]	Split edge fraction	Omitted edge fraction
SegEM	4.1	79.8%	15.2%	4.8%	0.2%
CNN	3.3	87.4%	2.7%	9.5%	0.5%
CNN+GALA	4.2	88.7%	7.3%	3.5%	0.5%
FFN-a	5.0	84.1%	14.3%	0.5%	1.1%
FFN-b	8.9	97.9%	0.0%	0.9%	1.2%
FFN-c	10.9	98.3%	0.0%	0.6%	1.2%

Supplementary Note Table 4. Evaluation of segmentation quality on the densely skeletonized $[5\ \mu m]^3$ subvolume.

Segmentation	ERL [µm]	Edge accuracy	Merged edge fraction[1]	Split edge fraction	Omitted edge fraction
SegEM	42	67.7%	28.0%	3.7%	0.8%
CNN	26	76.3%	8.6%	14.0%	1.2%
CNN+GALA	112	50.1%	46.4%	2.5%	0.1%
FFN-a	208	84.3%	13.5%	1.3%	0.9%
FFN-b	88	96.0%	0.0%	2.8%	1.2%

[1] Fraction of edges belonging to segments that contain at least one merger.

FFN-c	1,097	94.5%	2.8%	1.6%	1.0%

Supplementary Note Table 5. Evaluation of segmentation quality on the test set of 50 skeletons, with a total path length of 97 mm and max ERL of 2.1 mm.

Comparing the data in Supp. Note Tables 4 and 5, we observe that small-scale evaluations can severely underestimate the merged path length and the relative quality of different segmentations. For instance, in Supp. Note Table 4, FFN-c looks like a small incremental improvement over FFN-b, but when evaluated over the whole volume, the former is significantly better in terms of reconstructed error-free path length.

There are two main reasons for this. First, some data quality issues are inherently local to different parts of the volume, and hard to capture in a single small subvolume. For instance, the densely skeletonized subvolume does not suffer from any misaligned slices, cutting artifacts, or weakly stained neurites. Second, as the rate of errors in the segmentation decreases, getting good estimates of the different error rates requires sampling larger path lengths, at some point exceeding those contained within a small subvolume.

Ablation Experiments:

To elucidate the impact of the recurrent and single-object nature of the FFNs, we trained four different network variants:

1. boundary prediction with no recurrent channel (standard approach),
2. boundary prediction with recurrent channel (multi-object prediction),
3. memory-less FFN (single object prediction, no recurrent channel),
4. full FFN (recurrent single object prediction).

Only the minimum required changes were made between the experiments, and all other parameters, such as the architecture of the network were kept fixed.

For experiments 1) and 2) with boundary prediction networks, the training data was formed by first applying morphological erosion with radius 1 to the ground truth data to ensure separation of nearby neurites, and then binarizing the resulting image. The soft labels of 0.05 for background and 0.95 for object interior were used, similarly to the original FFN. After inference, the boundary network predictions were thresholded at 0.5, and converted into a segmentation by computing the connected components of the regions labeled as object interior.

For experiments 1) and 3), the second channel of the network input was set to a uniform empty image at value 0.05, and fixed-step movement procedure was used with a step size of (8, 8, 4) voxels. The "disconnected voxel bias" was active, allowing the network to override prior predictions of "object interior" to "exterior", but not vice versa.

Network type	ERL [µm]	Merged	Split	Omitted
boundary	8.0	3.8%	0.5%	3.0%
recurrent boundary	8.8	0.0%	0.5%	2.5%
single object	5.8	1.9%	1.2%	4.5%
recurrent single object (FFN)	10.9	0.0%	0.5%	1.1%

Supplementary Note Table 6. Evaluation of segmentation quality of different network types.

The results of our experiments presented in Sup. Note Table 6 suggest that the recurrent nature of the network is the main factor responsible for its segmentation accuracy, driven by its ability to eliminate mergers. These small scale experiments do not show the single object nature of the FFN to have a significant impact on the results. We note however, that this property is crucial for other procedures used in our pipeline (consensus, agglomeration), which were not applied here, but which were necessary to obtain good segmentations at the scale of the whole volume.

SNEMI3d Experiments

Group Name	Rand Error	Trainable Parameters	Test-time Augmentations
PNI [18]	0.0249	1.5M	16x
FFN (GAIP) w/ Orphan Reconnection	0.0291	0.5M	1x (single-scale; no consensus)
FFN (GAIP)	0.0332	0.5M	1x (single-scale; no consensus)
PNI [18]	0.0333	1.5M	1x
Human	0.0600	-	-
DIVE	0.0602	18M x 3 models	16 variants X 3 models
IAL	0.0656	35M	20 variants

Supplementary Note Table 7. SNEMI3d benchmark results (http://brainiac2.mit.edu/SNEMI3D/leaders-board as of 3/9/2016). Smaller numbers are better.

The SNEMI3D challenge is to automatically segment a small (6.1 x 6.1 x 3 μm) block of mouse neocortex tissue acquired with ATUM-SEM. Due to the acquisition method, the image data is highly anisotropic, with a voxel size of 6x6x30 nm. Two 1024x1024x100-voxel volumes are provided: a manually annotated training volume, and a test volume for which the labels have been withheld. Challenge submissions are evaluated with the Adapted Rand error, defined as 1 - F1-score of the Rand index.

We trained an FFN network using the provided ground truth labels, and with the raw images pre-processed with CLAHE. The architecture and hyperparameters of the network were kept the same as that reported in the main text, except for the default POM value and movement threshold, which during both training and inference were set to 0.5 and 0.6, respectively (these settings resulted in slightly better convergence during training).

Given the small size of the subvolumes and lack of additional context data around the evaluation area, we applied mirror-padding to artificially extend the size of the volume by 64 voxels in the XY direction and 16 voxels in Z. Segmentation was performed on the 1152x1152x132-voxel extended volume. Inference results on the training volume had no merge errors, so we did not apply oversegmentation-consensus or multi-scale segmentation.

The 3 agglomeration hyper-parameters were optimized by grid search to minimize Rand error on the training volume, and set at $f_* >= 0.5$, $d_*/N_* < 0.04$, $J_{AB} >= 0.5$ where the subscript * stands for "either A or B". Candidate pairs were selected with the procedure described in "Candidate object pair generation". Similarly to the main text, local translation-only alignment was used, and subvolumes affected by a one-section irregularity detected as a lateral shift of equal to or exceeding 10 px were evaluated again with the bad section replaced with the preceding one.

For submission to the SNEMI3D challenge we post-processed the segmentation by performing a 2D watershed on the distance transform within the empty spaces, seeded with the FFN segments. This had the effect of completely filling the volume so that no background voxels remain. This submission was evaluated with a 0.0332 Rand error, well below the human annotator error of 0.0600 (see Sup. Note Table 7).

We then classified any fragment not touching one of the borders of the volume as an "orphan", analyzed all decisions points involving such fragments, and connected them to the highest scored partner by the Jaccard index J_{AB}, provided that score was >= 0.01. This further improved the results, decreasing the Rand error to 0.0291.

FIB-25 Experiments

segmentation	**VOI_{split}**	**VOI_{merge}**	**VOI_{total}**
MALA [17]	1.1249	0.0221	1.1470

CELIS-MC [37]	1.0246	0.0962	1.1208
FFN (unagglomerated)	3.2032	0.0053	3.2085
FFN (agglomerated)	0.8837	0.0538	0.9375

Supplementary Table 8. Volumetric FIB-25 segmentation evaluation. Smaller numbers are better.

segmentation	**VOI_{split}**	**VOI_{merge}**	**VOI_{total}**
MALA [17]	2.3621	0.0101	2.3722
CELIS-MC [37]	2.6104	0.0568	2.6672
FFN (unagglomerated)	4.7394	0.0076	4.7470
FFN (agglomerated)	1.9968	0.0471	2.0439

Supplementary Note Table 9. Synaptic FIB-25 segmentation evaluation. Smaller numbers are better.

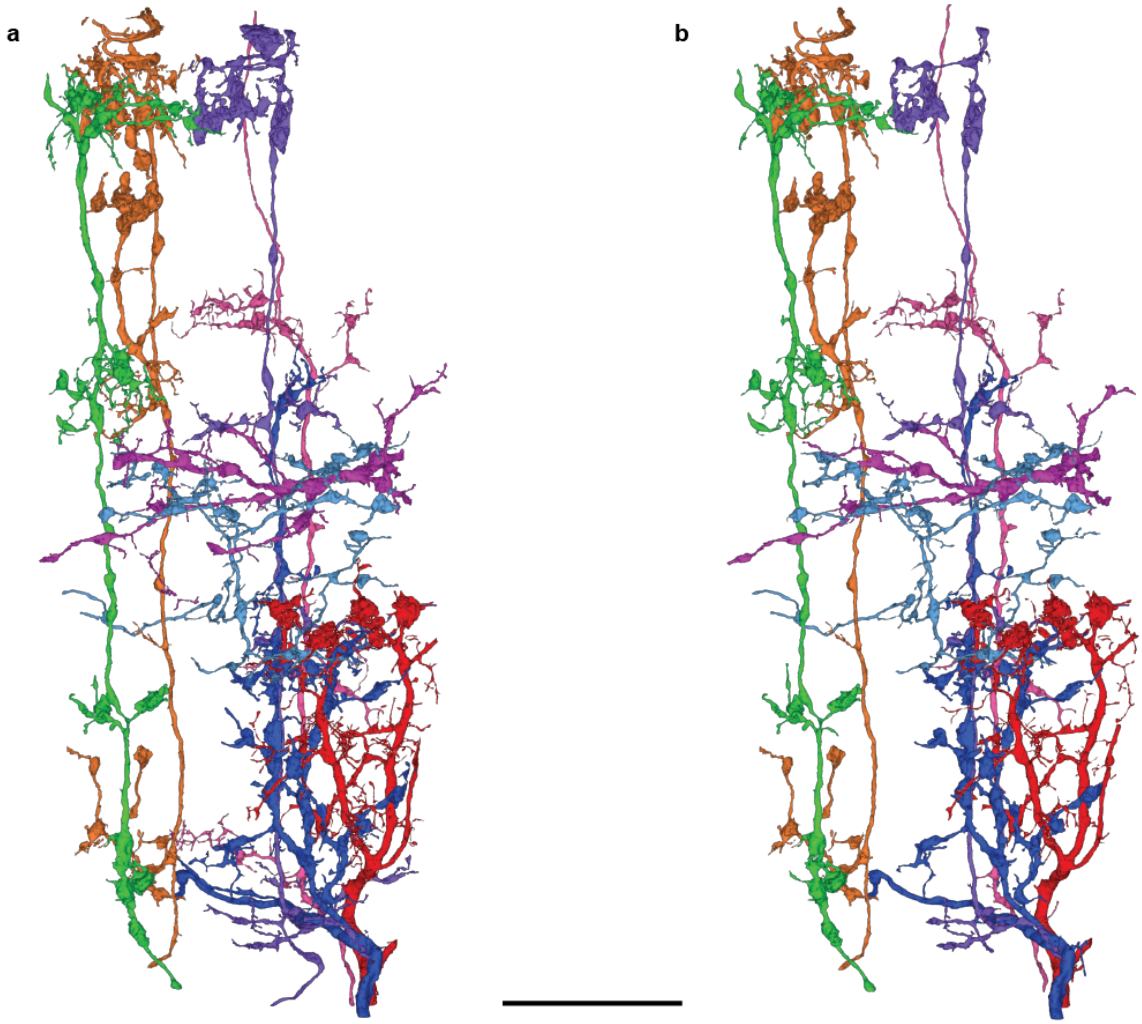
The FIB-25 dataset is a 52x53x65 μm volume of drosophila medulla imaged with FIB-SEM at 8x8x8 nm resolution. An irregularly-shaped 29.9 gigavoxel region of interest (ROI) containing 7 columns of the medulla was automatically segmented and a subset of objects in the ROI was proofread and corrected by human annotators. A list of synapse locations and the list of proofread objects within the ROI is provided along with two 520x520x520-voxel densely annotated subvolumes.

Prior work (MALA [17] and CELIS [37]) has divided the ROI into a "validation" ($z < 5006$) and "testing" ($z >= 5006$) region, as well as eroded the external boundaries of the ground truth volume with a radius of 50 voxels to remove heavily split areas. In our experiments reported below, we follow this convention. Based on recommendations from the authors of the FIB-25 dataset (the FlyEM project: https://www.janelia.org/project-team/flyem), we eroded the ground truth segmentation with a radius of 2 voxels and restricted evaluation to a “white-list" of fully-proofread objects (these two operations reduced the number of labeled voxels within the ROI to 7.5 billion). These procedures removed noise in the metrics caused by 1) the imprecise placement of the object boundaries in the ground truth (which were algorithmically generated) and 2) incompletely reconstructed objects in the ground truth (which were therefore excluded from the white-list). The authors of MALA and CELIS generously provided their results and we were able to re-evaluate these segmentations using this protocol.

We used one of the 520^3-voxel subvolumes to train the FFN (same subvolume used to train MALA), with a 33x33x33-voxel FOV, using the same procedure as for the SNEMI3D challenge. The second subvolume was used for network checkpoint selection. For inference, we performed

two segmentations of the selected network at full resolution (8x8x8 nm, forward and reverse seeds) and two segmentations at 2x reduced resolution (16x16x16 nm, forward and reverse seeds) and reconciled the segmentations with oversegmentation-consensus. We noticed that our standard seed selection procedure leaves a fraction of the finest branches branches untraced. To rectify this, we performed a computationally inexpensive FFN segmentation pass at full resolution with seeds at the local maxima of the distance transform within the remaining empty spaces. We also extended the segmentation with 3d seeded watershed. The 3 agglomeration hyper-parameters were optimized by grid search to maximize the synaptic variation of information (VOI) score on the validation region, and set at $f_* >= 0.3$, $d_*/N_* < 0.01$, $J_{AB} >= 0.75$.

In Sup. Note Table 8 and 9 we compare our segmentation to the two previously published approaches using synaptic and volumetric VOI. The unagglomerated base segmentation is heavily split and has a very low rate of mergers, as expected; the agglomerated FFN segmentation achieved the highest overall accuracy (see Sup. Note Fig. 3).



Supplementary Note Figure 3. Randomly selected neurites from the FIB-25 dataset. (a) Ground truth data. (b) FFN segmentation. Scale bar represents 10 µm.

Coarse-to-Fine FFN Segmentation of FIB-25

segmentation	VOI_{split}	VOI_{merge}	VOI_{total}
base (unagglomerated)	1.8888	0.0093	1.8981
16 nm agglomeration	0.0099	1.4322	1.4420
16 nm & 8 nm agglomeration	0.7860	0.0989	0.8849

Supplementary Note Table 10. Volumetric FIB-25 coarse-to-fine FFN segmentation evaluation. Smaller numbers are better.

segmentation	VOI_{split}	VOI_{merge}	VOI_{total}
base (unagglomerated)	3.6590	0.0100	3.6690
16 nm agglomeration	3.3367	0.0106	3.3472
16 nm & 8 nm agglomeration	2.0890	0.0603	2.1493

Supplementary Note Table 11. Synaptic FIB-25 coarse-to-fine FFN segmentation evaluation. Smaller numbers are better.

The modularity of the FFN segmentation approach allowed us to construct a coarse-to-fine segmentation pipeline that reduced the total computational cost. In the original segmentation pipeline described in "FFN segmentation pipeline", the entire volume is processed multiple times and at multiple scales. Here, we segmented in a coarse-to-fine manner, starting from downsampled data and then proceeding to native resolution. In contrast to the original pipeline, areas of the volume segmented at a lower resolution are not processed at the higher resolution.

We trained two separate FFN models, each targeted to a specific scale. The architecture and hyperparameters of the network were kept constant. We segmented the data at the coarse resolution (2x downsampled) and performed oversegmentation-consensus. The resulting segmentation labeled objects that could be resolved at the reduced resolution, and contained empty space in other areas. We used this segmentation (after upsampling) as the initial state of the next segmentation performed at native resolution. The resulting base segmentation was of comparable quality to that created with the standard pipeline, while the cost of inference was reduced by approximately a factor of 5.

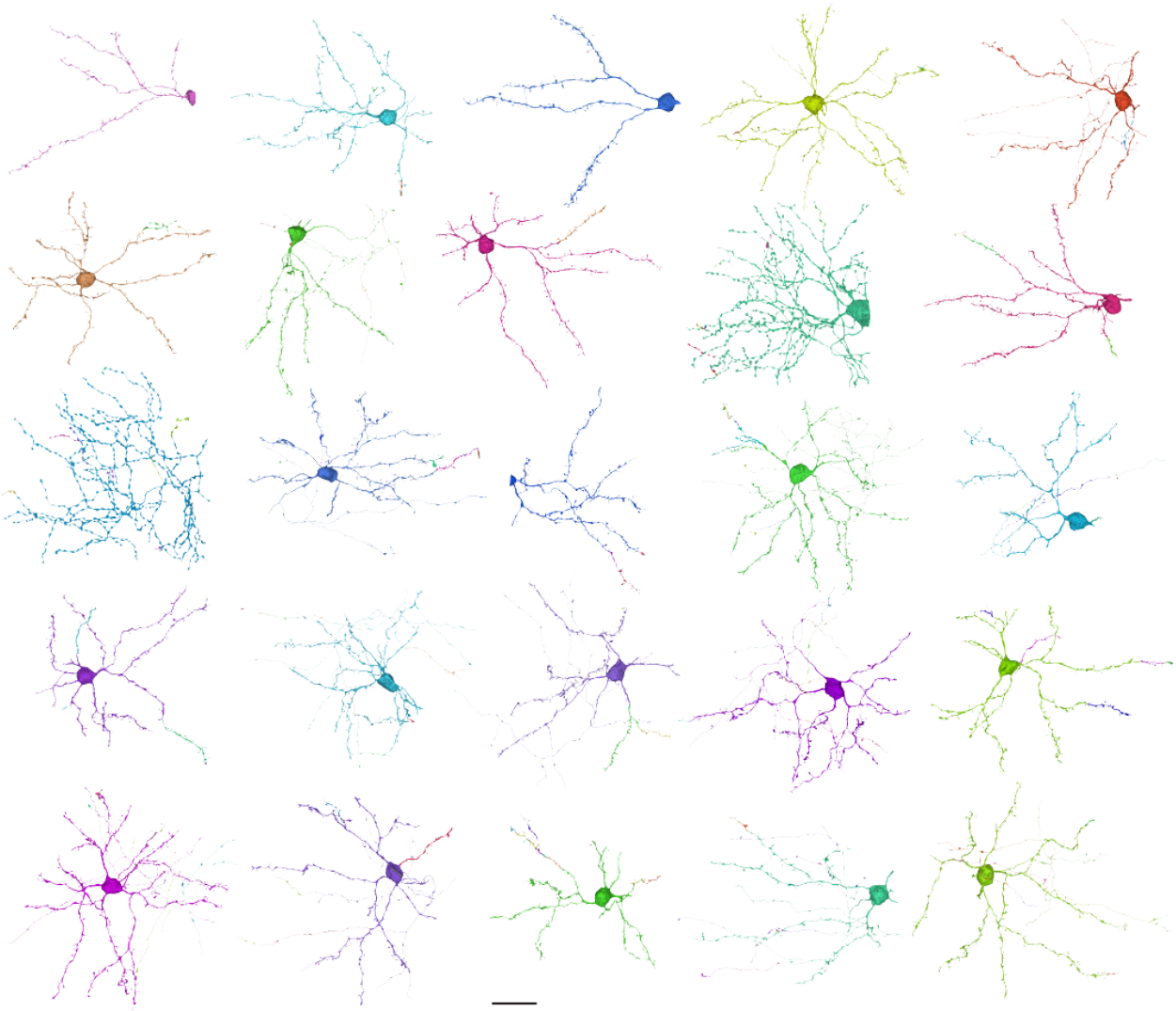
We evaluated the resulting segmentation using the protocol described in "FIB-25 Experiments." As documented in Sup. Note Table 10 and 11, the coarse-to-fine pipeline produced segmentations results that were comparable to the standard pipeline (Sup. Note Table 8 and 9).

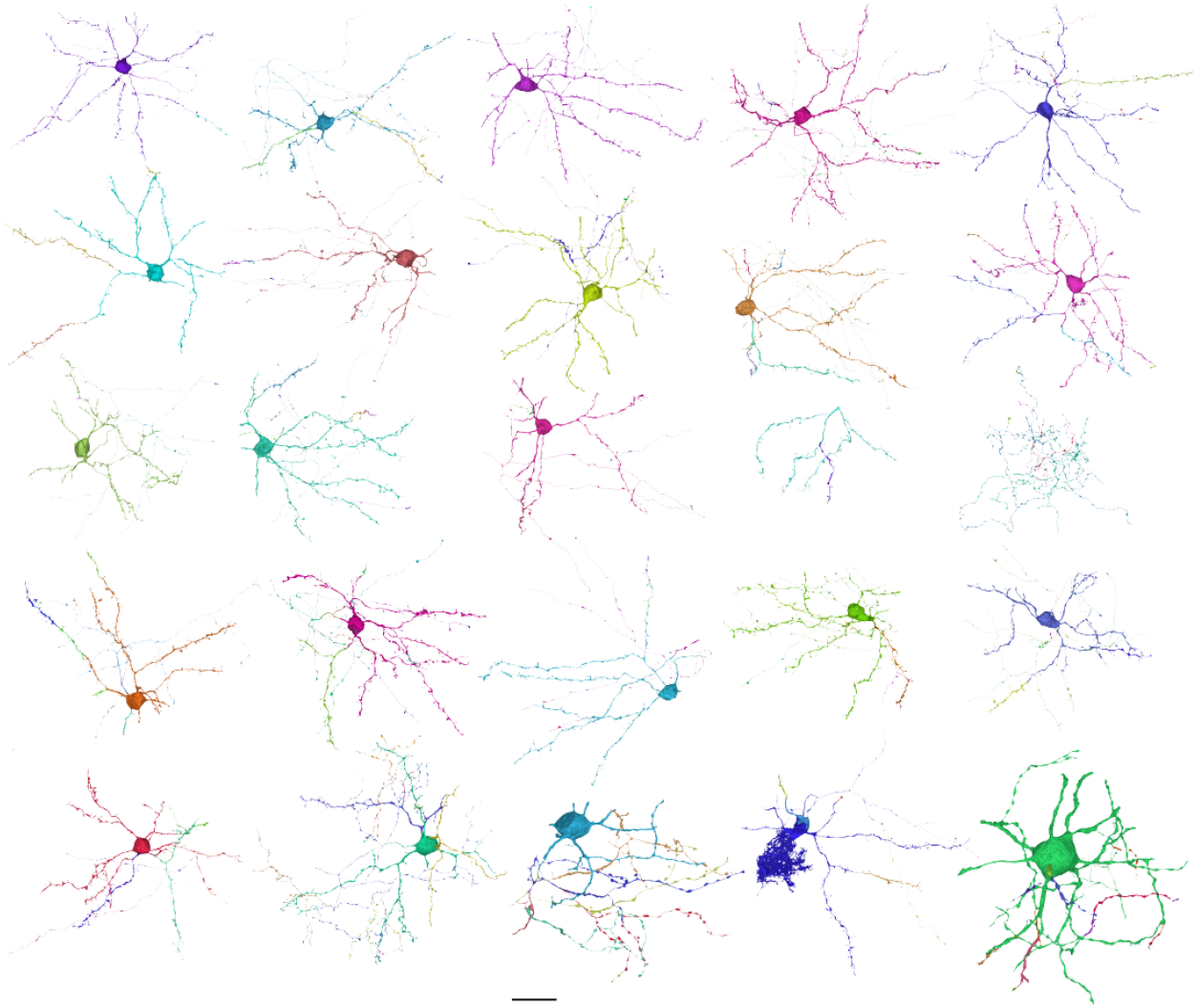
Hyperparameters used for J0126, SNEMI3d and FIB-25

Setting	Songbird (J0126)	SNEMI3d	FIB-25
Image preprocessing	none	CLAHE; mirror padding	CLAHE
FFN FOV size [voxels]	33x33x17	33x33x17	33x33x33
FFN FOV step size [voxels]	8x8x4	8x8x4	8x8x8
Number of residual modules in the FFN	9	9	12
Initial FOV fill value	0.05	0.5	0.5
Initial FOV seed value	0.95	0.95	0.95
FOV movement threshold	0.9	0.6	0.6
Image irregularity detection	tissue classification; patch-wise cross-correlation between neighboring sections	none	none
Consensus input segmentations	forward (36x36x40 nm), forward (18x18x20 nm), reverse (18x18x20 nm), forward (9x9x20 nm), reverse (9x9x20 nm)	forward (6x6x30 nm)	forward (16x16x16 nm), reverse (16x16x16 nm), forward (8x8x8 nm), reverse (8x8x8 nm)
Local alignment and bad section replacement during agglomeration	yes	yes	no
Agglomeration criteria (see "Agglomeration scoring" in Methods for symbol definitions; * means "both A and B")	$f_* >= 0.6 \wedge (d_A/N_A < 0.02 \vee d_B/N_B < 0.02) \wedge J_{AB} >= 0.8$	$((f_{A*} >= 0.5 \wedge d_A/N_A <= 0.04) \vee (f_{B*} >= 0.5 \wedge d_B/N_B <= 0.04)) \wedge J_{AB} >= 0.5$	$((f_{A*} >= 0.3 \wedge d_A/N_A < 0.01) \vee (f_{B*} >= 0.3 \wedge d_B/N_B < 0.01)) \wedge J_{AB} >= 0.75$

Supplementary Note Table 12. Summary of the hyperparameters used for segmentation of the zebra finch Area X (J0126), mouse cortex (SNEMI3d) and drosophila medulla (FIB-25) datasets.

J0126 Neuron Reconstructions in Skeleton Test Set





Supplementary Note Figure 4. Qualitative analysis of songbird data segmentation accuracy. Different colors indicate different segments. Neurons reconstructed with the full pipeline (FFN-c) ordered from largest (92%, top-left) to shortest (0%, bottom-right) fraction of maximum expected run lengths according to the skeleton ground truth test set. Scale bar represents 25 μm.

FFN Reconstruction of Single Neurite
FILE ATTACHED: supplementary_video_1_ffn.mov

Supplementary Note Video 1. FFN reconstruction of a single neurite (i.e., seeded from a single voxel) in J0126 volume.