NEURD: automated proofreading and feature extraction for connectomics

Brendan Celii[1,2,9,*], Stelios Papadopoulos[1,2], Zhuokun Ding[1,2], Paul G. Fahey[1,2], Eric Wang[1,2], Christos Papadopoulos[1,2], Alexander B. Kunin[1,2,8], Saumil Patel[1,2], J. Alexander Bae[4,5], Agnes L. Bodor[3], Derrick Brittain[3], JoAnn Buchanan[3], Daniel J. Bumbarger[3], Manuel A. Castro[4], Erick Cobos[1,2], Sven Dorkenwald[4,6], Leila Elabbady[3], Akhilesh Halageri[4], Zhen Jia[4,6], Chris Jordan[4], Dan Kapner[3], Nico Kemnitz[4], Sam Kinn[3], Kisuk Lee[4,7], Kai Li[6], Ran Lu[4], Thomas Macrina[4,6], Gayathri Mahalingam[3], Eric Mitchell[4], Shanka Subhra Mondal[4,5], Shang Mu[4], Barak Nehoran[4,6], Sergiy Popovych[4,6], Casey M. Schneider-Mizell[3], William Silversmith[4], Marc Takeno[3], Russel Torres[3], Nicholas L. Turner[4,6], William Wong[4], Jingpeng Wu[4], Szi-chieh Yu[4], Wenjing Yin[3], Daniel Xenes[15], Lindsey M. Kitchell[15], Patricia K. Rivlin[15], Victoria A. Rose[15], Caitlyn A. Bishop[15], Brock Wester[15], Emmanouil Froudarakis[1,2,12], Edgar Y. Walker[10,11], Fabian Sinz[1,2,13,14], H. Sebastian Seung[4], Forrest Collman[3], Nuno Maçarico da Costa[3], R. Clay Reid[3], Xaq Pitkow[1,2,9], Andreas S. Tolias[1,2,9], and Jacob Reimer[1,2,✉]

[1]Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine, Houston, USA
[2]Department of Neuroscience, Baylor College of Medicine, Houston, USA
[3]Allen Institute for Brain Science, Seattle, USA
[4]Princeton Neuroscience Institute, Princeton University, Princeton, USA
[5]Electrical and Computer Engineering Department, Princeton University, Princeton, USA
[6]Computer Science Department, Princeton University, Princeton, USA
[7]Brain & Cognitive Sciences Department, Massachusetts Institute of Technology, Cambridge, USA
[8]Department of Mathematics, Creighton University, Omaha, USA
[9]Department of Electrical and Computer Engineering, Rice University, Houston, USA
[10]Department of Physiology and Biophysics, University of Washington, Seattle, USA
[11]UW Computational Neuroscience Center, University of Washington, Seattle, USA
[12]Institute of Molecular Biology and Biotechnology, Foundation for Research and Technology Hellas, Heraklion, Greece
[13]Institute for Bioinformatics and Medical Informatics, University Tübingen, Tübingen, Germany
[14]Institute for Computer Science, University Göttingen, Göttingen, Germany
[15]Research Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, Baltimore, USA
[*]first author

We are now in the era of millimeter-scale electron microscopy (EM) volumes collected at nanometer resolution (Shapson-Coe et al., 2021; Consortium et al., 2021). Dense reconstruction of cellular compartments in these EM volumes has been enabled by recent advances in Machine Learning (ML) (Lee et al., 2017; Wu et al., 2021; Lu et al., 2021; Macrina et al., 2021). Automated segmentation methods can now yield exceptionally accurate reconstructions of cells, but despite this accuracy, laborious post-hoc proofreading is still required to generate large connectomes free of merge and split errors. The elaborate 3-D meshes of neurons produced by these segmentations contain detailed morphological information, from the diameter, shape, and branching patterns of axons and dendrites, down to the fine-scale structure of dendritic spines. However, extracting information about these features can require substantial effort to piece together existing tools into custom workflows. Building on existing open-source software for mesh manipulation, here we present "NEURD", a software package that decomposes each meshed neuron into a compact and extensively-annotated graph representation. With these feature-rich graphs, we implement workflows to automate a variety of tasks that would otherwise require extensive manual effort, such as state of the art automated post-hoc proofreading of merge errors, cell classification, spine detection, axon-dendritic proximities, and computation of other features. These features enable many downstream analyses of neural morphology and connectivity, making these new massive and complex datasets more accessible to neuroscience researchers focused on a variety of scientific questions.

EM connectomics | Neural Morphology | Automated Proofreading | Neural Annotation
Correspondence: ***reimer@bcm.edu***

Introduction

To understand the morphological features of individual neurons and the principles governing their connectivity, the use of large-scale electron microscopy and reconstruction of entire neural circuits is becoming increasingly routine. For example, in the past year, the MICrONS Consortium published a millimeter-scale open-source dataset of mouse visual cortex (Consortium et al., 2021) (approximately 80,000 neurons and 500 million synapses; "MICrONS dataset"), and a team at Harvard published a similar reconstructed volume of human temporal lobe (Shapson-Coe et al., 2021) (approximately 15,000 neurons, 130 million synapses; "H01 dataset"). These reconstructions offer opportunities for analysis of neural morphology and synaptic connectivity at a scale that was previously inaccessible. However, effectively using these massive and complex datasets for scientific discovery requires a new ecosystem of software tools.

Here, we describe NEURD — short for "NEURal Decomposition" — a Python software package that extracts useful information using only the 3-D mesh and synapse information of a neuron (along with some background information about the coordinate system of the volume), and implements workflows for a variety of downstream applications including automated proofreading, morphological analysis and connectomic studies. NEURD decomposes the 3-D meshes of neurons from EM reconstructions into a richly-annotated graph representation with many pre-computed features. These graphs characterize the

neuron at the level of non-branching segments (no protrusions other than spines and boutons) in the axonal and dendritic arbor, a relatively coarse level of representation that enables a wide variety of queries about the statistics and geometry of neural wiring.

We begin by demonstrating the utility of this framework in an automated proofreading pipeline that is highly effective at correcting merge errors using heuristic rules. We focus on merge errors because of their catastrophic effects on connectivity analyses. If all merge errors are resolved, then every synaptic connection on a neuronal segment can be trusted and used for downstream analysis, even if many connections are still missing. Thus, our proofreading workflow is conservative; it is intentionally biased towards pruning away mesh fragments (and their associated synapses) that may be due to merge errors, so that a very low percentage of remaining connections are invalid (low false positives), even at the expense of filtering away some potentially valid connections (false negatives). We evaluate the performance of this pipeline against human proofreading both at the level of cleaning merge errors from entire axonal and dendritic arbors, and at the level of generating split points for each error that can be executed in a hierarchical segmentation graph (Dorkenwald et al., 2022a,b) to batch-correct large numbers of errors. Because proofreading is performed by a set of interpretable heuristic rules, we show how the feedback from this validation process can be used to tune specific parameters in order to achieve even higher accuracy (true predictions / total predictions) on certain kinds of edits.

We then show how the pre-computed features extracted by NEURD can enable us to recapitulate and extend a variety of previous observations about neural morphology and geometry, taking advantage of the thousands of reconstructed neurons spanning all layers in these volumes. These diverse features include everything from spine annotations to the branching geometry of axonal and dendritic arbors, enabling powerful queries across multiple spatial scales. Using combinations of these features, we show that it is possible to distinguish excitatory and inhibitory neurons in both the mouse and human datasets with high accuracy, and to classify neurons into excitatory and inhibitory sub-classes (basket, bipolar, etc.), even when using isolated fragments of dendrites.

Finally, we examine the potential of this highly-annotated, sparse but clean connectivity graph to yield novel scientific insights about neural circuit connectivity, including higher-order motifs where we observe at least three nodes and edges in a specific graph arrangement (e.g., 3 nodes with 3 edges connected in a triangle where all edges are pointing clockwise). NEURD includes a fast workflow to identify axonal-dendritic proximities (regions where the axon of one neuron passes within at least within 5 μm of the dendrite of another) that can serve as a null distribution for analyses focused on understanding the specificity of synapse formation between different cells, even if their axon and dendritic arbors are incomplete. We use this capability to show that the ratio of synapses to proximities (synaptic conversion rate) varies as expected across cell-type-specific connections, and we show that our automatic proofreading enables us to identify a set of rare multi-synaptic connections with four or more synapses between pairs of excitatory cells. We find that these pairs have more similar functional properties *in vivo*, as predicted by a principle of "like-to-like" connectivity in the mouse visual cortex (Lee et al., 2016; Ko et al., 2011; Ding et al., 2023).

The following tables are provided in the appendix for further reference: Table 1 - cell type subclass abbreviations and excitatory/inhibitory classification glossary, Table 2 - dataset sizes and relevant n's for all figures and statistics, and Table 3 - a comprehensive guide to the pre-computed features provided by NEURD.

With NEURD, we set out to achieve two goals: The first goal is to extract a large set of useful features that many downstream analyses have in common, and to represent them in a compact, intuitive, and manageable format. Having ready access to these extracted features will be a valuable resource and enable many analyses beyond the ones we consider here. The second goal is to use these features to perform automated proofreading of merge errors, removing the large majority of these confounding errors in the volume and producing a high-fidelity, sparsely-sampled connectome that can be paired with appropriate proximity controls to study principles of neural connectivity. A useful analogy may be drawn with software tools that have supported the widespread adoption of other complex data modalities such as population calcium imaging (CaImAn, Suite2P; Giovannucci et al. 2019; Pachitariu et al. 2017), neuropixel recordings (KiloSort, MountainSort; Pachitariu et al. 2023; Chung et al. 2017), label-free behavioral tracking (DeepLabCut, MoSeq, SLEAP; Mathis et al. 2018; Pereira et al. 2022; Markowitz et al. 2018), and spatial transcriptomics (Giotto, Squidpy; Dries et al. 2021; Palla et al. 2022). Like these packages, the goal of NEURD is to make "big neuroscience data" (in this case, large-scale EM reconstructions) accessible to a larger community, by providing a common pre-processing and feature extraction workflow that can support a wide variety of downstream analyses. As more large-scale EM reconstructions become available, tools like NEURD will become increasingly essential for exploring principles of neural organization across multiple species.

Results

Summary of large-scale dense EM reconstructions. Data collection for the MICrONS and H01 dataset has been described in previous publications (Consortium et al., 2021; Shapson-Coe et al., 2021). The tissue preparation, slicing procedure, and imaging resolution (4-8nm x 4-8nm x 30-40nm) was roughly similar in both cases: Both volumes were stained with heavy metals (Yin et al., 2020; Tapia et al., 2012), embedded in resin, and cut on an automatic tape-collecting ultramicrotome (ATUM) (Schalek et al., 2011; Own et al., 2015). However, the imaging and reconstruction workflows for the two datasets were very different. The MICrONS volume was collected with transmission electron microscopy (TEM) (Yin et al., 2020), while the

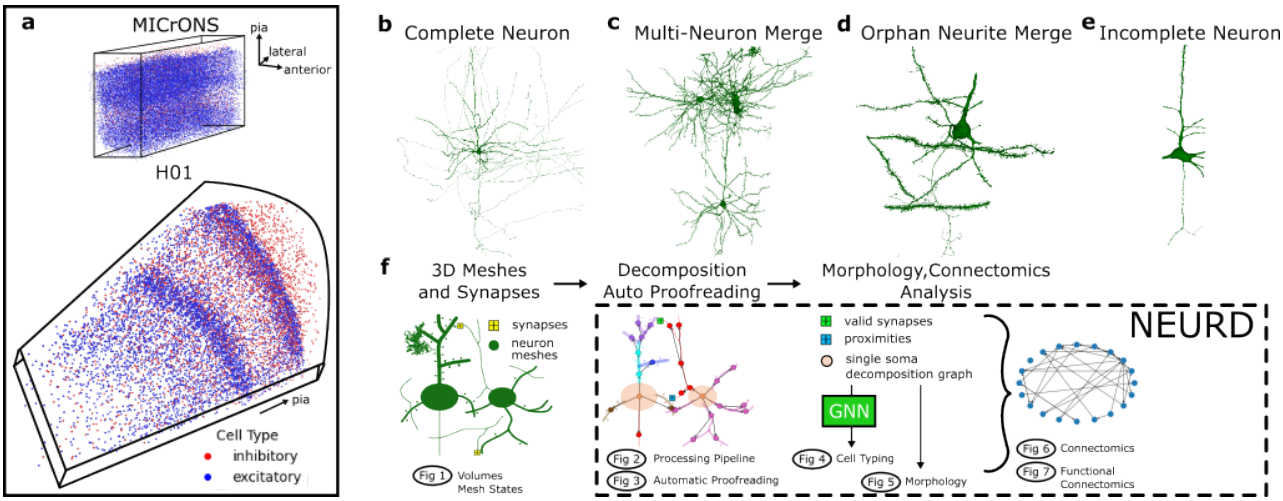


Fig. 1. Working with neuronal meshes from large-volume EM segmentations a The portion of the MICrONS volume that we analyzed (65% of the entire reconstruction) is an approximately $1300 \times 820 \times 520\,\mu\text{m}^3$ rectangular volume from mouse visual cortex, while H01 is a wedge-shaped volume with a longest dimension of 3mm, a width of 2mm, and a thickness of 150μm from human temporal cortex. Panels **b-e** illustrate the range of accuracy across neural reconstructions in both the MICrONS and H01 volumes. **b** Example of a nearly complete (manually-proofread) single neuron, and **c** a mesh containing two merged neurons from the MICrONS volume. **d** Example of an orphan merge error with a piece of dendrite incorrectly merged onto a neuron mesh, and **e** an incompletely-reconstructed neuron from the H01 volume. The reconstructed meshes are then pre-processed through a number of steps before analysis. Panel **f** provides an overview of the NEURD workflow with numbers representing the main figures in the paper.

H01 volume was collected with scanning electron microscopy (SEM) (Hayworth et al., 2014), and slice alignment and dense reconstruction for the two datasets were performed via different reconstruction pipelines.(Macrina et al., 2021; Consortium et al., 2021; Shapson-Coe et al., 2021). However, despite the differences in source tissue and significant differences in the acquisition workflows for MICrONS and H01, in the end the electron microscopy images are registered into a 3-D image dataset, voxels are automatically grouped into separate entities (the segmentation process), and finally the surface (ideally corresponding to the cell membrane) of these entities are wrapped with a triangular mesh. All volumetric reconstructions produce similar 3-D meshes as a common data product downstream of the segmentation process. The capabilities of NEURD are focused at the level of these mesh representations, which are much more lightweight than the original EM data, but still capture rich information about the microscale anatomy of neurons that can be useful for a variety of downstream analyses, including comparative analyses of neural circuitry across species, volumes, and reconstructions.

Preprocessing of Neuronal Meshes. EM reconstructions yield neural meshes with varying levels of completeness, and with different kinds of merge errors (Fig. 1b-e). Merge errors include multiple whole neurons connected together (Fig. 1c), and disconnected pieces of neurite ("orphan neurites") merged onto different neural compartments (Fig. 1d). Merge errors may also include glia or pieces of blood vessels merged onto neurons (Fig. 1f). We take advantage of existing tools for mesh processing (Fabri and Rineau, 2023; Cacciola et al., 2023; Gao et al., 2023; Dorkenwald, 2022) and apply them in an initial workflow that is agnostic to the identity of the mesh object. The downsampled meshes enter a neuron-specific mesh preprocessing pipeline (Fig. 2b, see Methods): In this pipeline, merged glia and nuclei submeshes are identified and filtered away, the locations of all somas are detected, and the mesh representation of axonal and dendritic processes are skeletonized into a series of connected line segments. In a final series of steps, the correspondence between each skeletal segment and its overlying mesh is reestablished, and the mesh segments composing neuronal spines are automatically detected, taking advantage of the fact that it is easier to identify protrusions on individual non-branching segments rather than the entire dendritic arbor. Systematic inspection by manual proofreaders confirmed the high accuracy of the soma, axon, dendrite, glia, and spine annotations generated during the mesh processing workflow (Fig. 8), as well as the labeling of basal and apical dendritic compartments (Fig. 9).

Graph Decomposition. We decompose skeletons of axonal and dendritic processes into a directed tree graph (NetworkX object in Python Hagberg et al. 2008) where the root node is the soma, the other nodes are individual non-branching segments and edges project downstream in the direction away from the soma; therefore, subgraphs downstream of the soma are a stem. There may initially be multiple soma nodes in the graph, but these are split apart if more than one soma is detected, and any cycles in the graph are broken during the decomposition process (see Methods, Fig.2d). Previous work has emphasized the utility of this kind of graph representation of each neuron, which facilitates flexible queries and analyses of features and annotations at different scales: individual branches, subgraphs of stems, or all of the stems graphs at once (entire neuron) (Pastor et al., 2021; Schneider-Mizell et al., 2016).

NEURD computes a variety of features of decomposed neurons at the branch (node), stem (subgraph), or whole neuron (graph) level (Fig. 2c,d). Node features include mesh features of each branch (number of spines, average width, etc.), skeletal

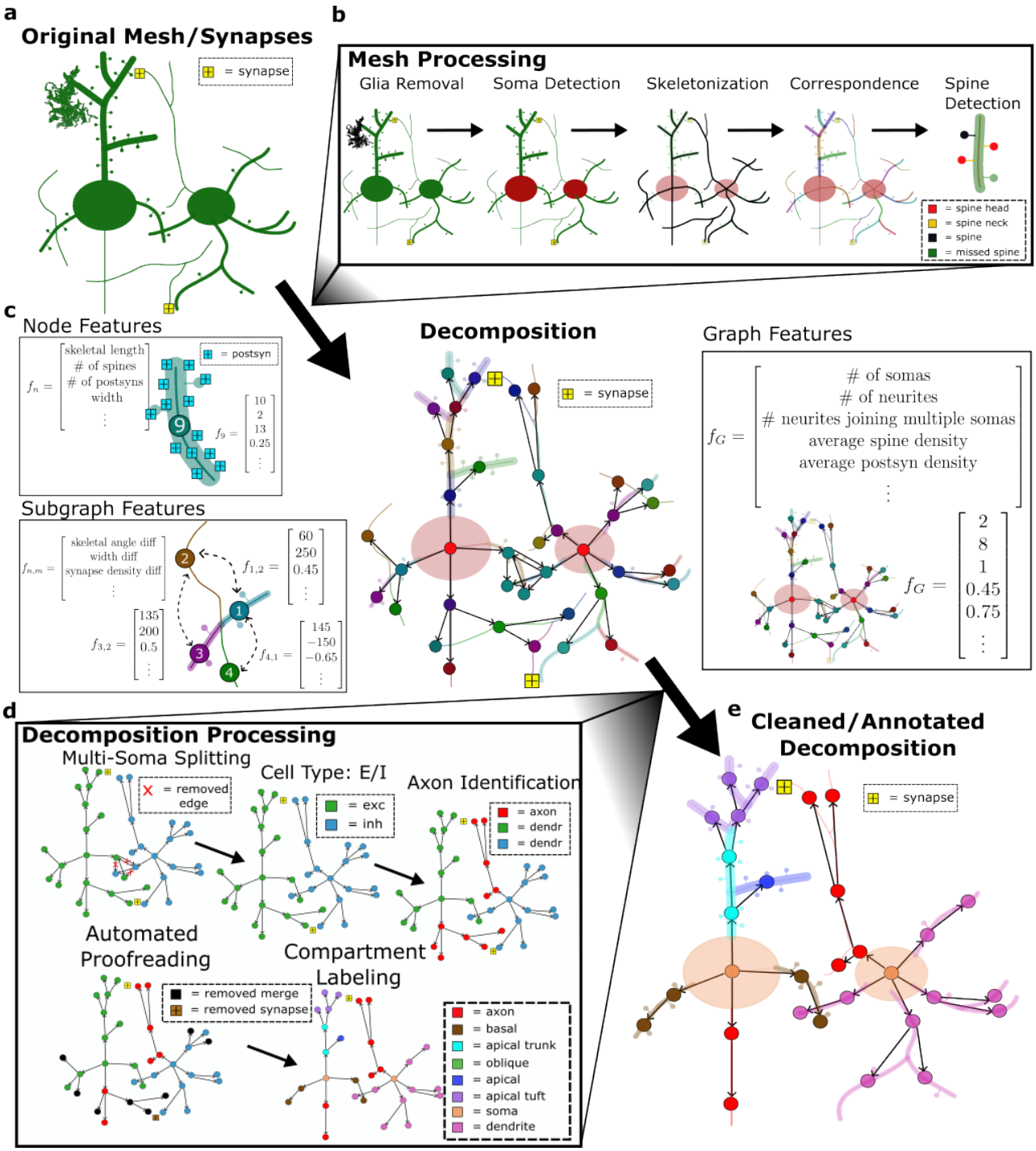


Fig. 2. Decomposition, feature extraction, and graph annotation a The input data (mesh and synapses) required for the decomposition, automatic proofreading and annotation downstream stages. **b** The reconstructed meshes are pre-processed through a number of steps including decimation, glia and nuclei removal, soma detection, and skeletonization. Finally, mesh features are projected onto branches from the skeleton and spines are detected. **c** Decomposition graph object composed of two neurons merged together. The decomposition compresses the skeleton, mesh and synapses (and any derived features) of a non branching segment into a node in a graph, with directed edges to the downstream segments after a branch point. The soma is a singular node connected to the downstream nodes bordering the soma. NEURD automates computation of features at multiple levels. Node (non-branching segment)-level features include basic mesh characteristics such as the diameter of the neural process or the number of synapses per skeletal length. Subgraph features capture relationships between adjacent nodes like branching angle or width differences. Graph features capture characteristics of the entire neuron and are computed by weighted average or sum of node features, or by counting subgraph motifs. **d** NEURD supports a variety of operations and manipulations on the decomposition objects. Multi-soma splitting is performed if a decomposition graph contains multiple soma nodes, splitting the graph into separate single soma subgraphs based on heuristic rules. The rest of the processing steps are performed on each single soma subgraph separately. The entire decomposition graph is classified as excitatory or inhibitory using a logistic regression model. A heuristic algorithm identifies one subgraph as the axon while labeling all other nodes as dendritic. Automated proofreading is performed by a series of heuristic graph rules identifying and removing probable merge errors (see Fig. 3). Finally, a set of heuristic rules are implemented to find subgraphs representing specific neural compartments (apical trunk, apical tuft, etc.). **e** The final product of this workflow is a cleaned and annotated decomposition object with a single soma that can be fed into a variety of downstream analysis pipelines.

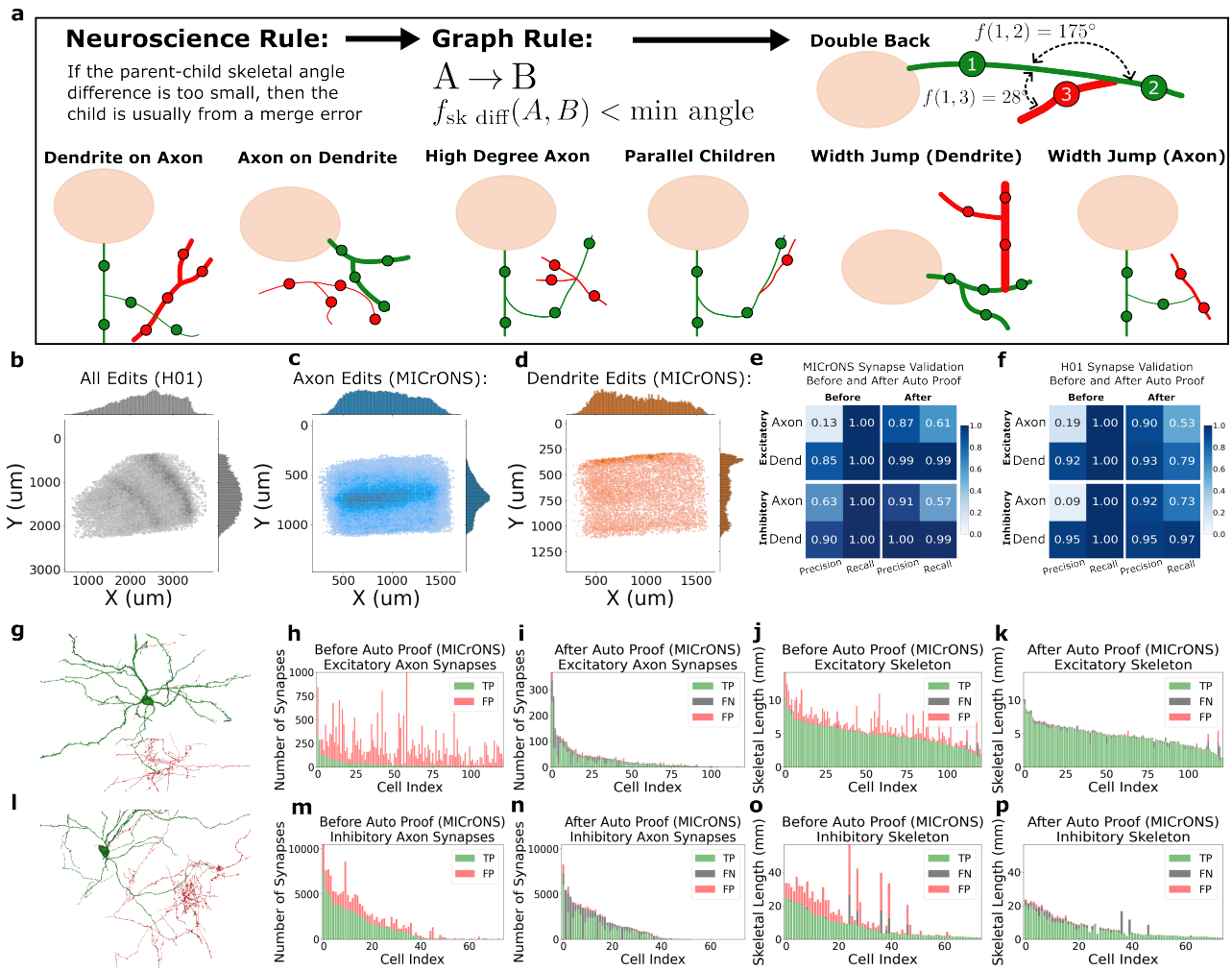


Fig. 3. NEURD graph decomposition enables automated proofreading a Implementing domain knowledge as subgraph rules to automatically identify and remove merge errors. Most of the same rules can be applied across excitatory and inhibitory cells in the MICrONS and H01 volume as-is, or with small changes in parameters. **b** Laminar distribution of all graph rule edits in the H01 dataset illustrating the inhomogeneity of errors across different layers, likely due to differences in neuropil density. Pial surface is to the right and slightly up. **c** In the MICrONS dataset, an increased frequency of axon edits is observed in layer 5 of cortex. Pial surface is up. **d** Dendritic errors in the MICrONS dataset are increased near the top layers of the volume, where fine excitatory apical tufts lead to more frequent merges. **e,f** Validation of MICrONS and H01 neurons quantified by synapse precision and recall compared to manual proofreading (regarded here as ground truth). "Before" indicates the state of the raw segmentation prior to any proofreading. Note the high precision of dendrites in both volumes, even prior to automated proofreading. The substantial increases in precision "After" automated proofreading indicate that the cleaned neurons have good fidelity nearly on par with manual cleaning. The reduction in "After" recall indicates that we are losing some valid synapses in the automatic proofreading process (mostly on the axons) but still retaining the majority of correct synapses. For the MICrONS dataset the validation is further separated into neurons with a multi-soma merge and those without in Fig. 15, showing the very high (99%) dendritic recall for single soma segments due to the high quality of initial dendritic reconstruction. **g**. Example excitatory neuron from the MICrONS dataset in the 75th percentile of merge error skeletal length; merge errors identified by automatic proofreading are shown in red. **h,i** Number of true positive (TP; green) and false positive (FP; red) axonal synapses from individual excitatory neurons in the validation set before and after automated proofreading, illustrating the large number of false positive synapses in the raw segmentation that are removed using the automated proofreading **j,k** Skeletal length of true and false excitatory axon skeletons before and after automated proofreading. **l-p** Same as **g-k** but for inhibitory neurons. Note that false-negative axonal skeleton in **j,o** indicates that some axon segments are lost in the mesh or decomposition preprocessing pipeline prior to automatic proofreading.

features (length, directed angle of projection, etc.) and synapse features (number of postsyns/presyns, number of spine head synapses, etc.). Subgraph features are computed from the interactions of node features in a local neighborhood of the decomposition graph, such as angle between skeleton projections and difference in average width. Graph features can be computed by averaging node features in the graph, such as the average number of spines per μm of skeletal length or the average number of postsynaptic connections per μm of skeletal length. Alternatively, graph features can be calculated by counting subgraph motifs, such as the number of subgraphs bordering at least one soma node or the number of subgraphs connected to multiple soma nodes.

These multi-scale features make it straightforward to translate neuroscience domain knowledge into neuron or compartment-level operations and queries. The most important context for this translation is automatic proofreading, where we demonstrate that these features can be used for accurate cleaning of merge errors including multi-soma merges. NEURD provides a complete workflow for multi-soma splitting and can also remove self-synapses if desired. In principle, when the agglomeration step can

be re-run, multi-soma splitting could also be performed by using NEURD's automatically-detected soma centers and then re-running segment agglomeration with a single-soma constraint, as in Januszewski et al. (2018). NEURD also provides workflows for common tasks such as annotating cellular compartments and classifying cells based on morphology. In addition to heuristic approaches for these tasks, the suite of graph features computed during decomposition can be used as inputs to statistical analyses or as training sets for machine learning. In what follows, we demonstrate the use of these annotated graphs in the context of automated proofreading (Fig. 3), cell-type classification (Fig. 4), morphological analysis (Fig. 5), and connectivity analysis (Fig. 6, Fig. 7). Note, these decomposition graphs can easily be converted to SWC files in case the user desires to utilize any other neuron processing pipelines requiring that format. We provide step-by-step online tutorials on how to export SWC files and interface with other SWC-based analysis software.

Automated Proofreading. Node, graph, and subgraph features computed during the decomposition step can be queried to identify potential sites of merge errors in the reconstruction, enabling rule-based automatic proofreading where all nodes downstream of the error are stripped from the mesh, returned them to the "sea of unconnected fragments", and excluding them from any subsequent analysis. While this approach does not enable automatic extension (i.e. fixing split errors), it can facilitate many subsequent analyses of individual cell morphology and circuit connectivity.

Automated proofreading involves defining graph configurations of nodes and attributes (called graph filters) that typically indicate merge errors, finding matching subgraphs in each neuron graph, and then labeling either all or some of the nodes in that match as errors. For example, a graph filter to locate dendritic branches with three or more downstream nodes and with certain width jumps or skeletal angle differences is useful for identifying errors in the H01 dataset because there are more dendritic merge errors compared to MICrONS. The graph filters are mostly directed one-hop or zero-hop configurations where one-hop configurations only consider itself and adjacent nodes, and zero-hop configurations consider only individual node features. However, proofreading rules can also consider larger-scale features of the graph (such as enforcing the existence of only one axonal arbor for each cell). Graph filter heuristics and parameters can be tuned for axon and dendrite subgraphs, excitatory or inhibitory cell types, and specific data sources (H01 vs MICrONS). In a workflow based on this approach, the user defines which graph filters to apply in which order, and as the errors are identified, the meta-data for each correction is stored (to be used for subsequent semi-supervised proofreading or training data for machine learning approaches). Once the algorithm processes all filters, the mesh, graph nodes, and synapses associated with all error branches are removed.

To illustrate this approach, we provide a small set of heuristic proofreading rules implemented as graph filters (Fig. 3a; see Methods) that yielded good performance on merge error correction in both volumes, but especially in the MICrONS dataset. An example rule indicating a merge error is "Low Degree Axon" (the parent class of the "Parallel Children" rule), where an upstream branch node with exactly two downstream nodes has its local graph configuration checked for properties more typically associated with merge errors (an approximate 0 degree or 90 degree angle between the skeleton projection of downstream nodes, a synapse at the branching point, very thin mesh segmentation at branching point, etc.).

Manual validation of these rules was performed in the context of standard proofreading and multi-soma splitting using the NeuVue Proofreading Platform (Xenes et al., 2022). We provided the proofreading team at John Hopkins University Applied Physics Laboratory (APL) with suggested error locations in the MICrONS volume, and experienced proofreaders evaluated each proposed split for accuracy (Fig. 10a-c). This validation included identifying split points for multi-soma splitting, axon-on-axon, and axon-on-dendrite merge errors, and enabled us to measure both the accuracy of these proofreading rules and the speed benefits of a semi-supervised approach compared to a fully-manual effort. We were also able to optimize these rules based on proofreader feedback, and this process enabled us to identify specific rules and parameter thresholds that we could apply with high confidence to correct merge errors without human intervention. Specifically, we used feedback from proofreaders to pick parameters that selected a high confidence subset of axon-on-axon merges and axon-on-dendrite merges. Validation of this subset then revealed a 99% and 95% agreement between the algorithm's split operations and those performed by a human proofreader (Fig. 10e). We applied nearly 150,000 of these high-confidence automatic edits back into the current MICrONS segmentation (spread throughout the volume and not explicitly confined to any one sub region). Furthermore, using NEURD suggestions in a semi-supervised manner to guide the challenging process of splitting multi-soma segments increased the speed of this process more than three-fold compared to other augmented methods (Fig. 10d, see Methods).

It is worth noting the scale of the merge errors removed by automated proofreading. In the MICrONS dataset, we identified hundreds of thousands of merge errors corresponding to dozens of meters of incorrectly-merged axons and dendrites onto the neurons within the $1mm^3$ volume (Fig. 11). Corrections in the H01 dataset were an order of magnitude smaller due to fewer cells and the less complete initial reconstructions in that volume, but were still substantial (Fig. 11). These errors reflect the scale of neural wiring that exists in a tiny volume of cortex, despite the high performance of both automated segmentations. Merge errors were identified throughout both volumes, and were more prevalent for some regions due to sectioning artifacts (Fig. 12g-l, Fig. 13h-n), or due to intrinsic differences in the morphology of neurons across layers (Fig. 3b-d, Fig. 12a-f, Fig. 13a-g,) For example high- and low-degree axon edits in the MICrONS dataset were frequently made in upper layer 5, potentially due to higher quantities of inhibitory neuropil, while dendritic double-back and width jump errors were more frequently located near the top layers of the volume, due to merges between fine distal apical tufts of excitatory cells (Fig. 12, Fig. 13).

We compared the outcome of automatic proofreading (all edits, not just the high confidence subset) to manual proofreading (regarded as ground truth) on a test set of cells in the MICrONS (n=122 excitatory and n=88 inhibitory) and H01 (n=49 excitatory and n=18 inhibitory) volumes. The manually proofread cells in the MICrONS dataset have soma centers concentrated in a 100×100 µm wide column spanning pia to white matter, away from the border of the volume (Schneider-Mizell et al., 2023) as well as some cells in higher visual areas (Fig. 31).The H01 dataset used a random subset for their manually proofread set, focused on cell bodies in the center of the "slab" (Shapson-Coe et al., 2021) (Fig. 31). The precision of the synaptic data, i.e. the number of actually-true synapses that were labeled as true divided by the total number of synapses labeled as true was substantially higher after proofreading (e.g., 0.87 after compared to 0.13 before for MICrONS excitatory axons, a greater than sixfold increase). For the same axons, this increase in precision was achieved with only a 40% reduction in recall (number of correctly-identified synapses divided by the number of true synapses). Precision and recall are summarized in Fig. 3e,f, Fig. 14, Fig. 15, and Fig. 16). The precision and recall of our automated method is captured visually in the plots in Fig. 3g-p. A perfect automatic proofreading algorithm would remove all the red false positives in these histograms, leaving only the green true positives, and would not cut into the green to produce any false negatives (gray). From the difference in "before" and "after" histograms, it is possible to see the accuracy of NEURD proofreading: Red false positives are almost entirely removed, at the cost of a thin gray margin of false negatives cutting into the green true positives. For example, for the axons of the the 122 excitatory neurons in the MICrONS dataset, NEURD correctly removed 24,430 false synapses and incorrectly removed 1,420 true synapses, leaving 2,216 true synapses and 324 false synapses. Full numbers are available in Appendix Table 2.

Note, because our automated proofreading procedure only removes data, recall is measured based on the synapses which were part of the automatic segmentation and which remained after manual proofreading was performed, but does not include synapses which were added in manual proofreading. Note also the high recall for dendrites, which was even higher (99% for MICrONS) when considering only single-soma neurons 15, reflecting the high performance of the initial segmentation. Overall recall was lower for axons (approximately 0.6 for both excitatory and inhibitory cells in the MICrONS dataset), indicating a larger number of incorrectly-removed axonal segments compared to dendrites. Performance on the H01 dataset was also reduced compared to MICrONS because the less-extensive reconstruction was associated with fewer merge errors overall. Initial reconstruction quality of dendrites in both volumes was high, and especially in the MICrONS volume, extensive, sometimes centimeter-scale arbors remained after removing merge errors (Fig. 3g,l, Fig. 17). Note that our proofreading heuristics and parameters were tuned to remove incorrect connections at the expense of sometimes removing correct connections, based on the rationale that incomplete, but largely-correct morphology and connectivity is more useful for downstream analyses than more complete neurons with falsely merged segments. In summary, both from the perspective of synapses (Fig. 3h,i,m,n) and skeletons (Fig. 3j,k,o,p), our automated proofreading approach can be applied at scale to yield neural reconstructions (Fig. 17) of similar quality to manually-cleaned cells without any extension.

Cell-type classification. Densely-reconstructed EM volumes hold great promise for understanding the connectivity between different neural subtypes (Schneider-Mizell et al., 2023, 2021; Dorkenwald et al., 2022a; Weis et al., 2022; Dorkenwald et al., 2022c; Peters and Feldman, 1976; Martin and Whitteridge, 1984). Because EM provides limited access to genetic markers, cell types must be identified by morphological features, although the relationship with molecularly-defined cell classes can be inferred from extensive previous work relating morphological features to transcriptomic classes (Scala et al., 2021; Peng et al., 2021; Gouwens et al., 2020), including from our consortium (Gamlin et al., 2023). Compared to light microscopy, the ultra-high resolution of EM reconstructions provides many additional features that can be used to classify cells by morphology. Previous studies have demonstrated that rich information enabling cell-type classification is available even in local nuclear and peri-somatic features (Elabbady et al., 2022; Al-Thelaya et al., 2021), small segments of neural processes (Dorkenwald et al., 2022c), different resolution views of single nuclei segments (Zinchenko et al., 2023), and the shape of postsynaptic regions (Seshamani et al., 2020). NEURD provides an additional rich and interpretable feature set at the level of non-branching segments that can be used for accurate cell-type classification via a number of different approaches, as we describe below.

As an initial demonstration, we performed Principle Components Analysis (PCA) on the averages of NEURD node features for neurons hand-labeled as inhibitory or excitatory (MICrONS n=3,985 excitatory and n=897 inhibitory; H01 n=5,800 excitatory and n=1,755 inhibitory). As expected, spine density (number of spines per μm of skeletal length) and shaft synapse density (number of synapses not on a spine per μm of skeletal length) were very informative features for distinguishing excitatory from inhibitory cells (Azouz et al., 1997). In fact, we found that a logistic regression model trained on just these two spine and synapse features separates excitatory and inhibitory cells with high accuracy, using the same parameters for classification across both the MICrONS and H01 dataset (Fig. 4a,b). We were able to perform this classification step prior to proofreading and axon/dendrite annotation, which enabled these rules to be specific to excitatory or inhibitory neurons.

To test whether NEURD graph objects could be used to distinguish even finer cell types, we turned to Graph Convolutional Networks (GCN) (Fig. 4c-f). We trained a simple GCN on the dendritic subgraph of a variety of hand-labeled cell types in the MICrONS volume (n=873 total cells), which represent a relatively complete set of cell type classes for this volume and are more thoroughly described in (Schneider-Mizell et al., 2023; Elabbady et al., 2022). We focused on the dendrites in this volume because of their high recall from the initial segmentation and the high precision after automated proofreading (Fig. 3f). This classifier produced an embedding space with a continuum of excitatory neurons progressing from the top layers down to

the bottom layers, while keeping inhibitory neurons and some morphologically-distinct layer 5 excitatory neurons (5P-NP, a subclass defined by the Allen Institute for Brain Science with connections to neighbouring regions (Schneider-Mizell et al., 2023), and 5P-ET, extra-telecenphalic neurons, which are sometimes referred to in literature as pyramidal tract (PT) neurons) clearly separated (Fig. 4c). Most of the embedding space was covered by the labeled dataset (Fig. 4c), and cells outside the labeled dataset had soma centroids at expected laminar depths (Fig. 4d); even though no coordinate features are used in the GCN classifier. Thus, the classifier was able to identify these layer-specific classes from distinct morphological features, not simply based on the location of cells in the training set (see Table 1 for the cell-type abbreviation glossary). Within the smaller volume with manually-labeled cells, we evaluated cell type classification performance on a held-out test set using a GCN with access to the entire dendritic graph (n=178 test neurons; mean class accuracy = 0.82; class accuracy: 23P (0.94), 4P (0.69), 5P-IT (0.60), 5P-NP (1.00), 5P-ET (0.86), 6P-CT (0.58), 6P-IT (0.62), BC (0.85), BPC (0.89), MC (1.00), NGC (1.00); Fig. 4e, also see actual counts for training, validation, and test in Fig. 19a-c). We also evaluated the classification performance using only disconnected dendritic stems (n=1023 test stems; mean class accuracy = 0.66; class accuracy: 23P (0.73), 4P (0.78), 5P-IT (0.33), 5P-NP (0.78), 5P-ET (0.78), 6P-CT (0.56), 6P-IT (0.49), BC (0.80), BPC (0.70), MC (0.79), NGC (0.50); Fig. 4f, counts for training, validation, and test in Fig. 19d-f). Consistent with previous literature classifying cells based on more local features, this demonstrates that the information present in disconnected individual dendritic stems (branching segments connected to the soma) is sufficient to perform fine cell-type classification nearly as well as graphs representing entire neurons. Additionally, because the classifier is a deep learning model, the output from the final softmax layer can be used as a confidence measurement, making it possible to restrict downstream analyses to neurons assigned to particular cell types with high confidence. (We used a confidence threshold of >= 0.7 for analyses downstream of GCN subclass predictions.)

Morphological Analysis. The features extracted by NEURD - including features of different compartments (Fig. 5a), the geometry of axonal and dendritic compartments (Fig. 5b), and spine features (Fig. 5c) - provide a rich substrate for morphological analysis (Fig. 20).

Extensive work has linked spine morphology to synaptic strength and stability, making them important targets for understanding plasticity and connectivity in neural circuits. To enable these analyses, automating spine detection, annotation, and feature extraction is important, since spines can number in the thousands per neuron. A variety of methods have been developed to automate spine detection in 2-D or 3-D image data using fully-automatic (Xiao et al., 2018; Driscoll et al., 2019; Janoos et al., 2009; Shi et al., 2014; Basu et al., 2018) or semi-automatic (Benavides-Piccione et al., 2013) approaches. NEURD offers an accurate spine detection workflow that achieves high accuracy with a fully-automated mesh segmentation approach. Importantly, these spine features are then available for queries in combination with all the other morphology and connectivity information available in NEURD graphs. Precision and recall for spines with a skeletal length larger than 700 nm was 90% or higher in extensive validation by manual annotators on both volumes (Fig. 8). In addition, NEURD isolates individual spine meshes, then segments the spine head from the neck (when possible), and finally computes statistics about the overall spine mesh and also the individual head and neck submeshes, creating a large scale dataset for testing hypotheses about spine morphology conditioned on postsynaptic compartment type or the cell type of the pre- or postsynaptic cell. As expected, the spine head volume and synaptic density volume were the only strongly correlated spine features (Fig. 23; Harris and Stevens 1989; Arellano et al. 2007). One question is whether the MICrONS and H01 dataset contain spines with similar three-dimensional geometry. One way to examine this is by projecting metrics of spine shape (spine length, head volume, neck width etc) into a low-dimensional space using a technique like UMAP. The spatial distribution of UMAP embeddings (two-dimensional projection) for feature vectors of spines sampled from the MICrONS and H01 dataset showed a similar structure, with spines that share similar features embedded in similar locations and a somewhat consistent embedding pattern for inhibitory and excitatory spines in the two volumes. This similarity suggests that H01 and MICrONS spines sample from a similar landscape of diverse spine shapes. (Fig. 5d, Fig. 5e), consistent with previous work examining the distribution of non-parametric representations of post-synaptic shapes across diverse neural subtypes (Seshamani et al., 2020). NEURD builds on this work by providing interpretable features for these distributions.

We attempted to replicate and extend several other findings observed in previous studies of the MICrONS and H01 datasets regarding the sub-cellular targeting of synaptic inputs. First, we computed the number of synapses onto the axon initial segment (AIS) of neurons at different depths. Replicating a previous report, in the MICrONS volume, superficial L2/3 pyramidal cells received the largest number of AIS synapses, with up to 2-3 times the innervation of the lower cortical layers (Fig. 5f; Schneider-Mizell et al. 2021; Wang et al. 2019; Inan et al. 2013). However, in the H01 dataset, this laminar inhomogeneity in AIS synapses was much less prominent, with more similar numbers of AIS inputs observed across all depths (Fig. 5g). Additionally, like AIS synapses, we found a striking difference in the distribution of somatic synapses across depth between the MICrONS and H01 dataset. (Fig. 21g,h). Lastly, the overall frequency of somatic synapses were also distinct across the two volumes, consistent with previous literature describing fewer somatic synapses in the human compared to mouse (Fig. 21d; Wildenberg et al. 2021); however, we found the opposite trend for the AIS, with fewer AIS synapses in the mouse volume compared to the human volume (Fig. 21c).

In H01, deep layer pyramidal cells were previously observed to have a strong bias in the radial angle of their thickest basal dendrite (Shapson-Coe et al., 2021). The geometry of neural branching is a subgraph-level feature computed by NEURD, and

process diameter is a node feature, so we examined the combination of these features to replicate this finding. First we examined the MICrONS volume and did not observe a strong bias in thickest basal, even in deep layers (Fig. 5h). Then, looking at H01, we were able to replicate the pattern of thickest basal dendrite direction preferences in deeper layers (Fig. 5i). However, we also found that this pattern appeared to continue into more superficial layers. We wondered whether the previous emphasis on deep layers might be because the effect was more salient there. Indeed, making use of the pre-computed widths for all branches, we found that in deep layers the difference between the thickest and second-thickest dendrite was nearly doubled compared to more superficial layers. Thus, this dendritic orientation bias appears to be a general property of the H01 volume, albeit less salient in more superficial layers.

The diversity of pre-computed features offered by NEURD enabled us to identify several interesting morphological features that differ across cell types, including many that have been reported previously in other studies. For example, the spindly, non-branching basal dendrites of NP cells (Schneider-Mizell et al., 2023; Weis et al., 2022) are clearly distinct from extensively-branching basal dendrites of L2/3 pyramidal cells. This feature is clear in a histogram of total skeletal length across individual nodes (Fig. 5j), as is the fact that neurogliaform cells are the most highly-branched neurons with the largest number of leaf nodes (leaf nodes are terminal nodes with no further downstream branching; Fig. 22a). As an interesting sidenote, we found that across all neurons, dendritic stems with larger numbers of leaf nodes had a larger initial dendritic diameter at their connection to the soma (Fig. 22b-c), potentially reflecting developmental or metabolic constraints.

Comparing synapses onto the dendritic shaft with synapses onto dendritic spines provides a rough proxy of inhibitory versus excitatory inputs (Ribak et al., 1981; DeFelipe and Fariñas, 1992; Kwon et al., 2019). In a histogram of shaft to spine synapses, NP cells were again located at the higher end of the distribution, while L4 and L2/3 pyramidal cells had the lowest shaft-synapse to spine-synapse ratio (Fig. 5k), suggesting they receive a relatively larger fraction of excitatory (compared to inhibitory) input. Because NEURD also automatically segments both soma meshes and spine heads and necks, this enables comparison across cell types of features like soma volume and somatic synapses (Fig. 20b,f), spine neck length (Fig. 5l), spine density (Fig. 20a), and the relationship between spine synapse size and spine head volume as in Fig. 5m and Fig. 23.

Connectivity and Proximities. Next, we examined the connectivity graph in the MICrONS and H01 datasets after automatic proofreading. Proofreading substantially reduced the mean in-degree (number of incoming synapses onto a neuron) and out-degree (number of projecting synapses from a neuron) across both volumes due to the removal of merge errors, resulting in a sparsely-sampled but high-fidelity graph (Fig. 3e,f). While the recall (probability of retaining an originally correct synapse after classification) of the synaptic connectome is lower than ideal compared to manual proofreading, the term "high-fidelity" emphasizes the connectome's high precision (probability that a sampled synapse from the retained set is valid). Low precision can be catastrophic for downstream analyses, while low recall can support many analyses based on the assumption that the observed synapses are a representative sample of the full distribution. A variety of connectivity statistics including number of nodes and edges, mean in and out degrees, and mean shortest path between pairs of neurons along excitatory and inhibitory nodes is provided in Fig. 25. Note that "edges" here are synapses between neurons, and a subset of neurons that were not fully processed by the decomposition pipeline are excluded from all statistics comparing the "raw" graph to the graph after automated proofreading (neurons with manual proofreading are also excluded from this comparison). The shortest path statistics are generated by randomly sampling source and target neurons; sample pairs without a path between them are excluded from the calculation.

To facilitate the analysis of synapse specificity in sparse connectomes, we implemented a fast workflow for identifying axonal-dendrite proximities, regions where the axon of one neuron passes within at least within 5 μm of the dendrite of another neuron (See Methods, Fig. 6a). Note that these proximities would not be meaningful without first cleaning all skeletons of merge errors. The width measurement at every skeletal point can also be used to exclude myelinated axons from this proximity computation using a maximum width threshold. Previous studies have computed proximities from skeletons of simulated models (Udvary et al., 2022), or manually traced data (Mishchenko et al., 2010; Kasthuri et al., 2015), with a similar logic. Proximities are necessary but not sufficient for the formation of a synapse (Peters and Feldman, 1976; Brown and Hestrin, 2009; Mishchenko et al., 2010; Costa and Martin, 2011), and so the "proximity graph" can serve as a valuable null distribution for comparing potential connectivity with synaptic connectivity between neurons: Instead of looking at synapse counts between cells which are dependent on the geometry and completeness of the neuropil, proximities make it possible to calculate "conversion rates" - the fraction of proximities which resulted in actual synaptic connections. NEURD also provides functions to compute "presyn skeletal walk" - the distance from a synapse to the soma of the presynaptic neuron along the axon, and "postsyn skeletal walk" - the distance from synapse to soma along the postsynaptic dendrite. Combined with cell typing, compartment labeling and spine annotation, these features enable powerful analyses of neural connectivity conditioned on the cellular identity and subcellular location of synapses on both pre- and post-synaptic partners (Fig. 6b).

Conversion rates between neural subtypes in the MICrONS dataset replicated previous results from connectivity measured via slice multi-patching and EM reconstructions, especially the prolific connectivity of basket cells onto both excitatory and inhibitory somas (Fig. 6c; Jiang et al. 2015; Schneider-Mizell et al. 2023; Lee et al. 2013; Freund and Katona 2007) and inhibitory-inhibitory relationships including BC inhibiting other BC, MC avoiding inhibiting other MC, and BPC preferentially

inhibiting MC (Fig.27; Pfeffer et al. 2013; Jiang et al. 2015; Lee et al. 2013; Schneider-Mizell et al. 2023, see Table 1 for the cell-type abbreviation glossary).

The subcellular targeting of different inputs is apparent in plots of the "postsyn skeletal walk" distance to the soma for synapses arriving at the basal dendrite. As has been previously described (Ribak et al., 1981; Hwang et al., 2021; Megıas et al., 2001), inhibitory-onto-excitatory synapses tend to be found closer to the somatic compartment than excitatory-onto-excitatory synapses (Fig. 6d,e). At an even smaller scale, with the spine head, spine neck, or shaft classification propagated to synapses, we can study how excitatory and inhibitory inputs to spines display different scaling relationships between synapse size and spine head volume (Fig. 24). We also show, as expected, that excitatory and inhibitory cells differ in the number and relative sizes of synapses on their target spine heads (Fig. 24; Parnavelas et al. 1977; Megıas et al. 2001).

Conversion rates for excitatory-to-excitatory proximities were low in both H01 and MICrONS, consistent with previous findings of sparse pyramidal cell connectivity in the cortex (Fig. 6f,g; Campagnola et al. 2022; Jiang et al. 2015; Kasthuri et al. 2015; Mishchenko et al. 2010). However, conversation rates were substantially higher for excitatory-to-inhibitory proximities (Fig. 6f,g), especially in H01, and were substantially higher for proximity distances less than 2 microns (unlike excitatory synapses onto excitatory cells, where spines presumably reduce the dependence on distance). Combining the presynaptic (axonal) skeletal walk features and proximity analyses revealed an interesting similarity in excitatory-onto-inhibitory connectivity between the MICrONS and H01 datasets, with conversion rates peaking in the more proximal axon a few hundred microns from the soma (Fig. 6h,i; Bock et al. 2011; Bopp et al. 2014; Schmidt et al. 2017; Dorkenwald et al. 2022c). Conversion rates were also higher above (more superficial to) the presynaptic soma than below (deeper than) the presynaptic soma for excitatory-onto-inhibitory connections in both volumes (Fig. 26).

Large-volume EM connectomics offers tremendous potential opportunities to examine higher-order motifs on a large scale, beyond pairwise connectivity. By comparing the frequency of synaptic motifs with "proximity motifs" defined by neurons that had a chance to participate in a particular synaptic motif based on their axon/dendrite geometry, it is possible to begin to ask questions about higher-order connectivity, connections participating in higher-order motifs, even with incompletely-reconstructed neurons because they have been stripped of merge errors. With this approach we found that more densely-connected triangle motifs were enriched in the MICrONS volume compared to several controls, including a null graph with the same number of nodes and edges where synaptic degree distribution is held the same but edges are shuffled (configuration model), a control where the synaptic edges are shuffled only between neurons with an existing proximity edge, or one where synapses are random shuffled between neurons regardless of proximity (Fig. 6j). We also performed the same analysis restricted to either excitatory-only or inhibitory-only subgraphs. Within these subgraphs, the effects on motif expression can be analyzed for null conditions where the excitatory to excitatory or inhibitory to inhibitory conversion rate is maintained (Fig. 28). This is consistent with previous findings at smaller spatial scales suggesting that this higher-order organization is enriched in the cortex (Song et al., 2005; Perin et al., 2011; Milo et al., 2002; Udvary et al., 2022; Schneider-Mizell et al., 2021). A similar pattern was observed in the H01 dataset, consistent with previous modeling of connections and proximities there (Udvary et al., 2022). In the H01 volume several of the three-node motifs with larger numbers of connected edges were missing due to the less complete reconstruction (Fig. 28).

Functional Connectomics. A key advantage of the MICrONS dataset is the additional functional characterization of matched neurons in vivo prior to EM data collection. The relationship between function and synaptic connectivity is covered in detail in a separate paper (Ding et al., 2023), which includes analysis using NEURD's automatically-proofread data. Here, we wanted to provide an illustration of how automated proofreading can enable a specific functional connectomics analysis that would be otherwise very challenging. We identified pairs of excitatory neurons connected by one, two, three, or four or more synapses. Querying for these rare high-degree connections between pyramidal cells was only possible after automated proofreading to correct merge errors, especially since inhibitory axon fragments (for example from basket cells) are much more likely to form multi-synaptic connections and so merge errors onto excitatory cells are a substantial confound for this analysis. In fact, approximately 97% of the 10,000+ pairs with four or more connections were identified as merge errors during automatic proofreading. Connections were further restricted to synapses onto postsynaptic spines in order to guard against possible missed merge errors where an inhibitory axon segment might still be merged to an excitatory neuron. Examples of these multi-synaptic connections have been highlighted in the H01 data set (Fig. 7a), and rare examples can also be found in the MICrONS data set (Fig. 7b; see also Chicurel and Harris 1992). We were able to identify n=11 of these pairs in exclusively-automatically proofread neurons (no manual proofreading) where both neurons also had been characterized functionally (Fig. 7c). The average response correlation was calculated for each group of pairs (see Methods and Ding et al. 2023; Wang and Tolias 2023). We found that neurons with four or more synapses had significantly higher response correlations (0.259 +- 0.042 (n=11)) to visual stimuli than neurons with < 4 synapses (syn = 1: 0.118 +- 0.002 (n=5350), syn = 2: 0.140 +- 0.011 (n=280), syn = 3: 0.133 +- 0.030 (n=34)) connecting them (Fig. 7d), consistent with a Hebbian "fire together/wire together" rule governing high-degree connectivity in the cortex, and this same pattern was also observed for n=12 pairs of manually-proofread neurons. For autoproofread neurons, two Sample Kolmogorov-Smirnov test and t-test for comparing each multi-synaptic group's null likelihood of being drawn from the same distribution as the one-synapse group: 2 synapses KS test statistic = 0.068 and p = 0.17, t-test p<0.03, 3 synapses KS test statistic = 0.139 and p=0.49, t-test p = 0.60, 4 synapses KS test statistic = 0.508 and p <

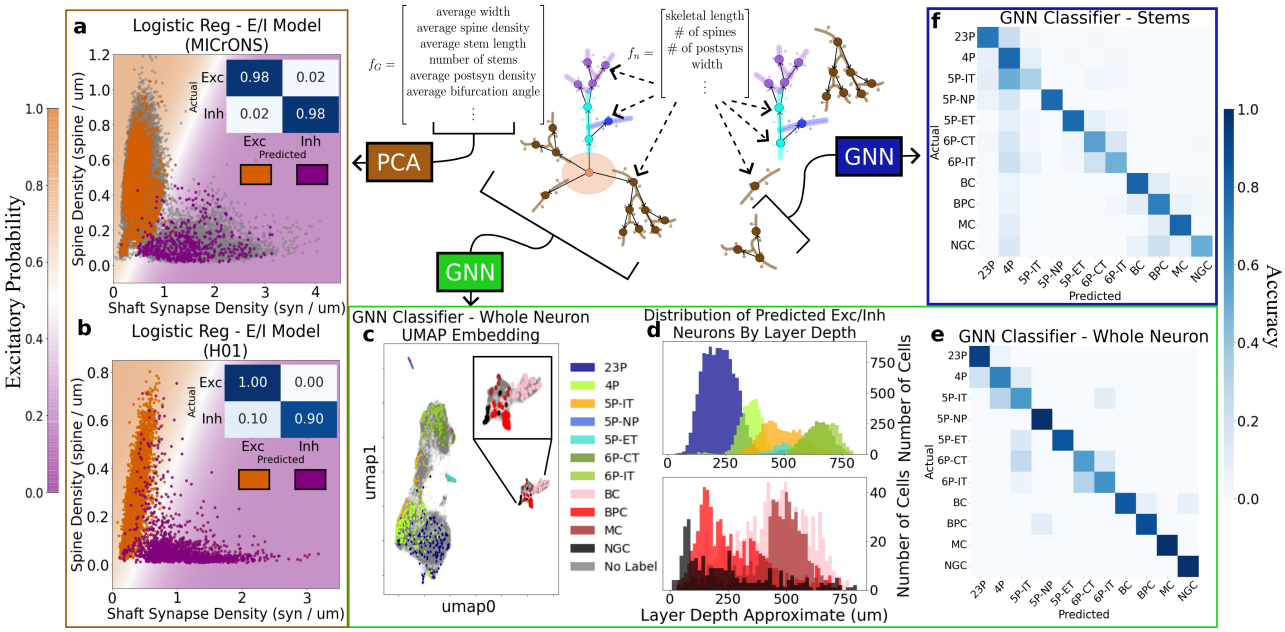


Fig. 4. Graph decomposition enables cell-type classification. a,b Two interpretable features identified by PCA were highly informative for excitatory/inhibitory classification: spine density (number of spines per μm of skeletal length) and shaft synapse density (number of synapses not on a spine per μm of skeletal length). Consistent with previous studies (Azouz et al., 1997; Kawaguchi et al., 2006), a logistic regression model trained on just these two features enabled linear discrimination of excitatory and inhibitory cells with high accuracy (same parameters for logistic regression model used for both MICrONS and H01 dataset). **c** A Graph Convolutional Neural Network (GCN) trained on manually-annotated cell types produces an embedding space with a continuum of excitatory neurons progressing from the top layers down to the bottom layers while keeping inhibitory neurons and some excitatory neurons with distinct morphology (5P-NP and 5P-ET) clearly separated (see Table 1 for the cell-type abbreviation glossary). **d** The depth of predicted cell types outside of the training volume are consistent with their expected laminar distribution even though no coordinate features are used in the GCN classifier. **e** Confusion Matrix of the test dataset for the neuron GCN classifier tested on n=178 held out neurons: 23P (n=33), 4P (n=51), 5P-IT (n=10), 5P-NP (n=4), 5P-ET (n=7), 6P-CT (n=19), 6P-IT (n=29), BC (n=13), BPC (n=9), MC (n=1), NGC (n=2). **f** Cell classes could also be determined using a dendritic subgraph of only one stem (branching segment connected to the soma) nearly as well as when using the entire dendritic tree, suggesting that the GCN identifies somewhat local features that enable classification. GCN tested on n=1023 test stems: 23P (n=230), 4P (n=301), 5P-IT (n=48), 5P-NP (n=18), 5P-ET (n=45), 6P-CT (n=116), 6P-IT (n=137), BC (n=65), BPC (n=30), MC (n=19), NGC (n=14).

10^{-2}, t-test p<10^{-2} (and still significant after the Bonferroni correction for multiple comparisons with a significance threshold of p<0.02).

Discussion

NEURD is an end-to-end automated pipeline capable of cleaning and annotating 3-D meshes from large electron microscopy volumes and pre-computing a rich set of morphological and connectomic features that are ready for many kinds of downstream analyses, even in the absence of extensive manual proofreading. Building on existing mesh software packages, NEURD adds a suite of neuroscience-specific mesh functions for soma identification, spine detection, and spine segmentation that are applicable across multiple data sets, as well as workflows for skeletonization and mesh correspondence that complement existing tools. The atomic unit (node) in the NEURD decomposition graph is either a single non-branching segment of axon or dendrite, or a soma, and each of these nodes is automatically decorated with a large number of features. We demonstrate the utility of this highly-annotated coarse-grained graph representation for many downstream analyses and tasks, including splitting multiple-soma merges, cell typing, automated proofreading of merge errors, and compartment labeling. We highlight the ability to implement interpretable heuristic rules in these workflows, but NEURD graphs can also be used as inputs for machine-learning based approaches, as we demonstrate in the context of cell-type classification. Our hope is that offering many automatically-computed features at the level of coarse-grained NEURD graphs will make these daunting datasets more accessible for a larger group of researchers. The set of features generated by NEURD is easily extensible, and the illustrations here have only tapped a tiny bit of their potential for scientific discovery.

Several previous studies have proposed post-hoc methods for automated proofreading including merge and split error detection and correction. Some of these methods make use of Convolutional Neural Networks (CNNs) (either supervised Zung et al. 2017; Gonda et al. 2021 or unsupervised Rolnick et al. 2017), reinforcement learning methods (Nguyen et al., 2022), or other machine learning approaches (Schubert et al., 2019; Schmidt et al., 2022; Berman et al., 2022). Others make use of heuristic rules applied to neural skeletons (Meirovitch et al., 2016; Sicat et al., 2013), and at least one approach uses both skeleton heuristics and CNNs (Matejek et al., 2019). Compared to automated methods applied directly to the EM segmentation that may be heavily memory-intensive, NEURD benefits from the lightweight features computed from mesh representations, allowing analysis to scale to larger volumes more easily. Methods based on the EM segmentation may have other advantages,

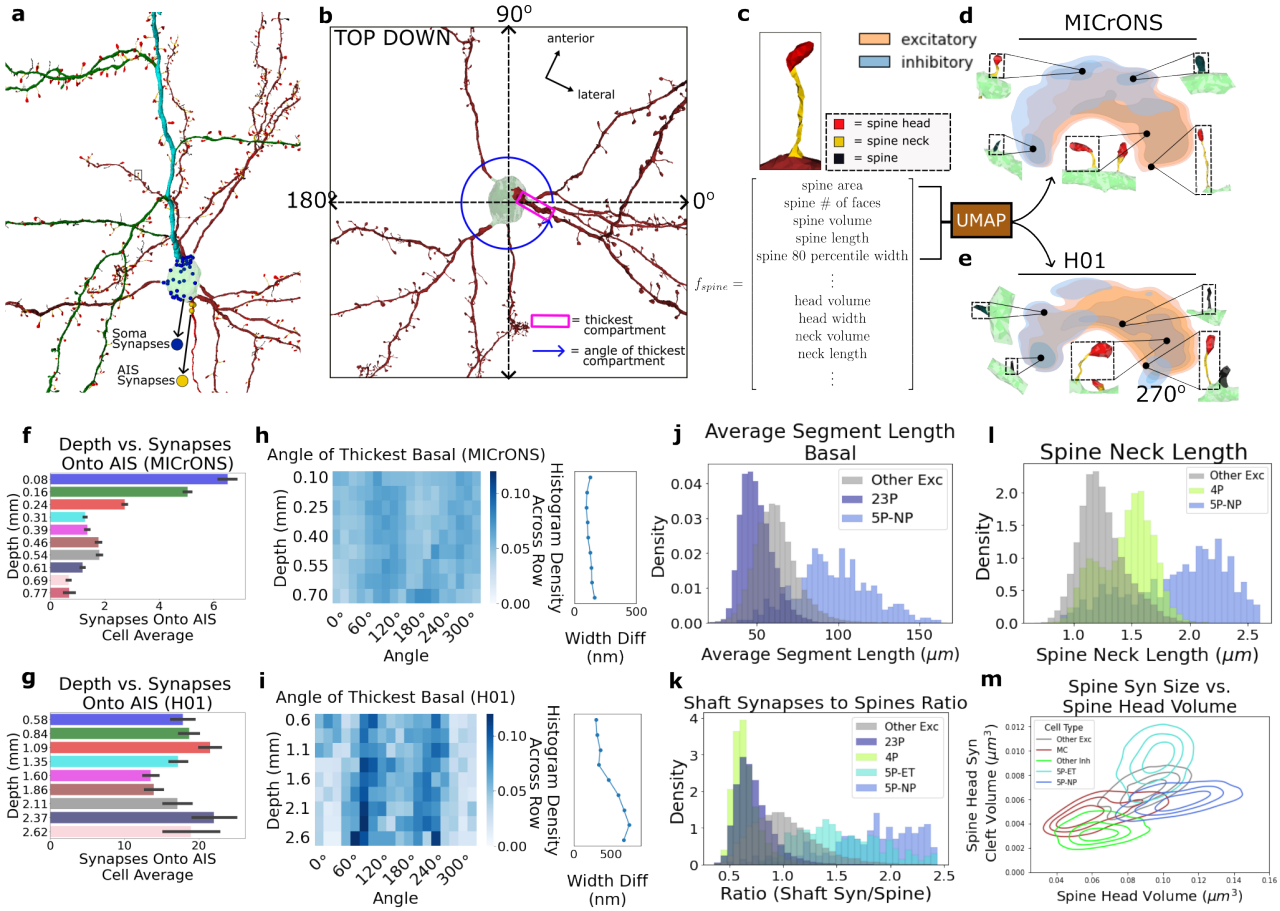


Fig. 5. Morphological analysis enabled by NEURD feature extraction. **a** Cleaned and annotated neuron mesh; soma synapses in blue, axon initial segment (AIS) synapses in yellow, basal dendrite in brown, apical trunk in aqua, oblique branches in green. Spine heads in red, spine necks in yellow and non-segmented spine in black along dendritic segments. **b** Top down view of the neuron in **a**. An example of geometric analysis: the thickest basal branch is boxed in pink and the xz angle of these branches are indicated with a blue circular angle marker. **c** Most spines are annotated with interpretable features such as head volume, spine skeletal length, and spine neck length while some spines (shown in black) which are smaller (typically under 0.7 um) or unable to be segmented, lack the head/neck features. **d,e** Kernel density estimation of UMAP embedding of spines sampled from MICrONS and H01 dataset using spine features from panel **c** (without head or neck features). The embeddings show a similar embedding structure between the two datasets in terms of spine shape and inhibitory/excitatory class, similar to previous work clustering a non-parametric representation of postsynaptic shapes (Seshamani et al., 2020). **f,g** Average number of synapses onto the axon initial segment (AIS) of cells at different laminar depths (mean +/- std) for the MICrONS and H01 volume. Same depth color scheme used in Fig. 21. **h,i** Histograms showing the distribution of mean skeletal angle of the thickest basal stem. Each row is a normalized histogram for a specific depth bin. The H01 dataset shows more bimodal structure (especially in the deeper layers) than the MICrONS dataset, consistent with a previous report (Shapson-Coe et al., 2021). We find that this pattern is also visible in more superficial layers of H01, but is less obvious because the width difference between the widest and the second-widest basal branch is much larger in deep layers ("Width Diff" plot to right of heat maps). **j** Distribution of the average skeletal length of non-branching dendritic segments for stems of different cell-type subclasses (see Table 1 for the cell-type abbreviation glossary). **k** The ratio of non-spine synapses to spine counts varies across cell types. **l** Distribution of average spine neck length of different excitatory cell type subclasses. As expected, 5P-NP cells have the longest average neck lengths, but 4P cells also display significantly longer necks than other excitatory cell types. **m** KDE plots (kernel density estimation, estimates the shape and distribution of the discrete dataset) of spines on different cell type subclasses, comparing each spine head's volume and the size of the largest synapse on that spine head. These plots reveal differences in both the distribution and scaling of synapse sizes and spine heads across cell types. For example Martinotti cells (red) have larger spine head synapses than other inhibitory spines (light green) and the head volume and synapse size scale at a rate more similar to other excitatory cells (grey).

for example access to intracellular features that may aid proofreading, and these methods could potentially still be employed upstream of NEURD. Although our method is based primarily on heuristic rules, NEURD can support a variety of downstream methods. Multiple proofreading approaches can benefit from NEURD's pre-computed feature set that includes cell types, compartments, spine information, and skeleton geometry. The coarse-scale graph structure facilitates flexible querying at the level of non-branching segments and the characteristics of the branch points between them. In the context of automated proofreading, having a pre-computed feature set at multiple scales can provide a clearer view of the context in which a potential segmentation error occurs. For instance, determining whether a thin, aspiny projection from a dendrite is the true axon or a merge error may be ambiguous without additional information about the cell type of the neuron, the distance of the candidate axon from the soma, and the spine density of the parent dendritic branch. This information can make the probability of a merge much more clear. We demonstrate how this information can be leveraged together in rules defined as graph filters with interpretable parameters. We find that many of these graph filters generalize across the two different volumes, either with or without tuning of thresholds. These parameters can also be tuned in interpretable ways for more or less conservative proofreading, or for

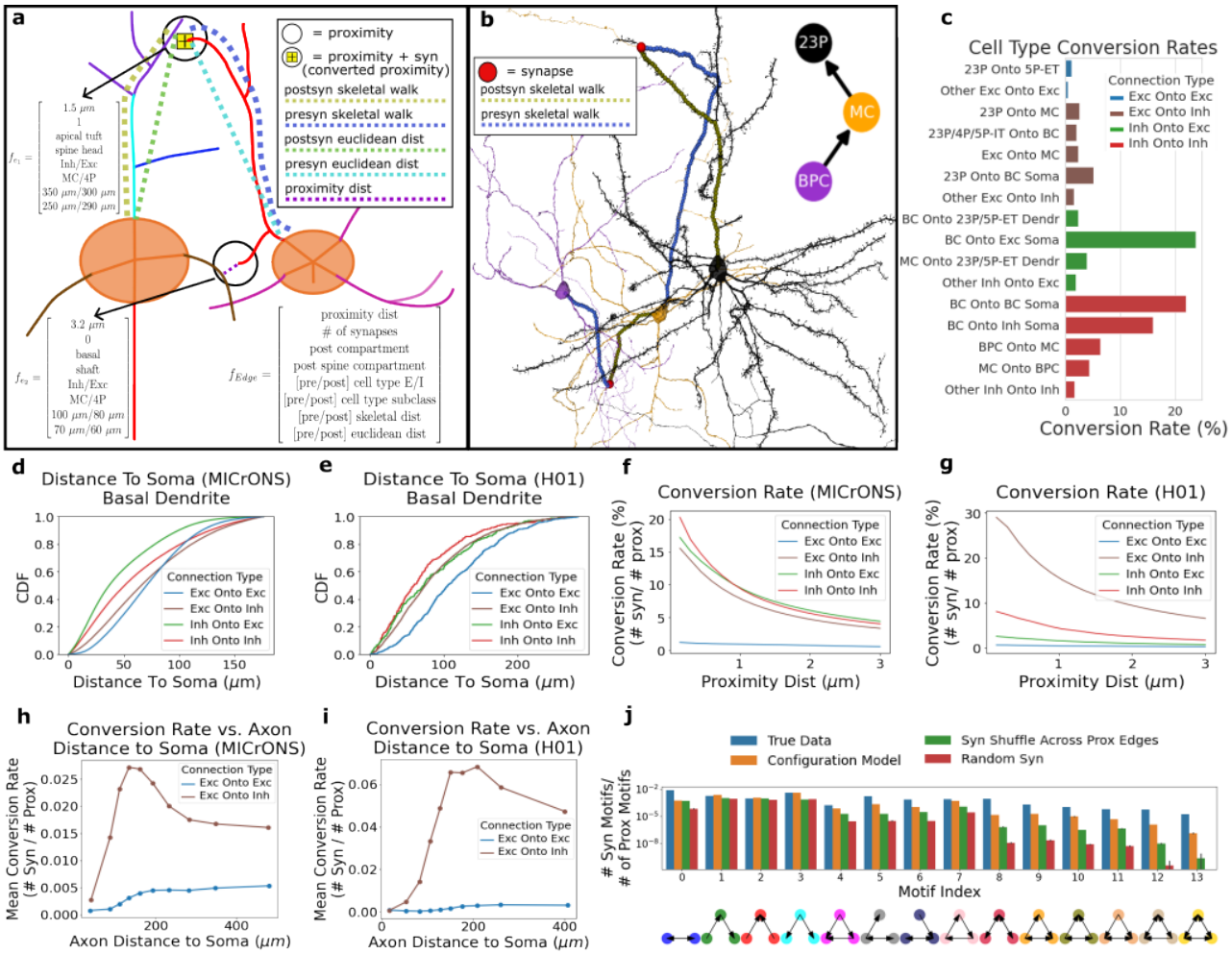


Fig. 6. Connectivity analysis enabled by NEURD a Schematic illustrating two proximities between a pair of neurons where one neuron's axon passes within 5 μm radius of a target neuron's dendrite. Only one proximity actually includes a synapse, thus in this example the "conversion rate" is 50%. Each proximity creates an edge in a proximity graph between the two neurons. Edge features for proximities include the number of synapses, the proximity distance between the closest presyn and postsyn skeletal point, postsyn compartment, pre/postsyn skeletal walk and euclidean distances, etc. **b** Cell-type specific connections and motifs in the MICrONS dataset can be found by querying the annotated connectivity graph. NEURD allows for visualization of these connection paths and motifs so they can be quickly inspected. **c** Conversion rates for different cell type subclasses and compartments in the MICrONS dataset are consistent with previous cell type specific connectivity work (cell type labels generated from GNN classifier, see Table 1 for the cell-type abbreviation glossary). **d,e** Cumulative density function (CDF) of the postsynaptic skeletal walk distance distribution for different exc/inh connection combinations (apical and soma synapses are excluded). In both datasets, excitatory inputs are further along the dendrite from the soma. **f,g** Conversion Rates (synapses / proximities) for different exc/inh combinations. The x-axis represents the maximum distance that is considered a "proximity". **h,i** Mean conversion rate as a function of distance from the synapse to the presynaptic cell along the axon. **j** Proximities enable the computation of null distributions for higher-order motifs even with incompletely-reconstructed neurons. The frequency of reciprocal connections or edge-dense three-node motifs was enriched compared to null graphs (with the same number of nodes and edges) where synaptic degree distribution is held the same but edges are shuffled (configuration model), where the synaptic edges are shuffled only between neurons with an existing proximity edge, or where synapses are randomly shuffled between neurons regardless of proximity (250 random graph samples for each null distribution comparison).

different downstream applications.

Our present implementation does not address some types of errors in automated segmentation. For example, it cannot presently handle merge errors that result in a co-linear segment of skeleton that we interpret as a single non-branching segment. Second, because NEURD's present implementation is focused on removing false merges, it is unable to fix incomplete neural processes. In future work, we plan to try using NEURD to perform automated extensions, by "over-merging" candidate segments at a truncation point, and then correcting the resulting "merge error" to choose the best candidate for extension. Another current limitation is that NEURD only takes advantage of the information available in neuronal meshes, which is restricted to features that are visible at the level of the cell membrane. Using the segmentation information to loop back to the raw EM data might allow us to capture some of the additional rich ultrastructural features that are available in EM data, for example the density of organelles such as mitochondria, in order to add additional annotations to the NEURD graph. We performed extensive validation of our automated proofreading approach to determine the precision and recall of our error correction, and we have highlighted the reassuring agreement of many of our findings with previous literature. However, as a general disclaimer it is important to note that some of the results we have presented here that are based on NEURD's automatic proofreading could

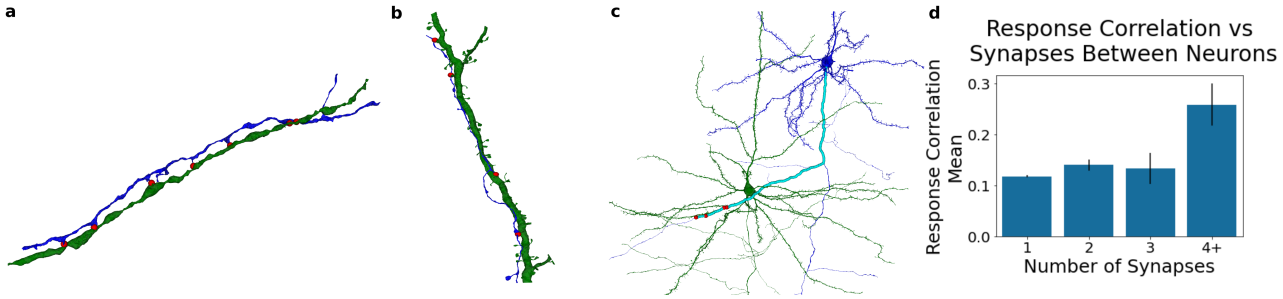


Fig. 7. Functional Connectomics Illustration: High-degree cell pairs a Example multi-synaptic connection (n=7 synapses) from an excitatory to inhibitory neuron in the H01 dataset **b** Example multi-synaptic connection (n=4 synapses) from excitatory to excitatory neuron in the MICrONS dataset. **c** Example of a highly spatially clustered multisynaptic connection (n=4 synapses) on a postsynaptic basal dendrite from a neuron cleaned with automated proofreading in the MICrONS dataset (presynaptic skeletal walk shown in aqua, synapses shown in red) **d** Distribution of response correlation between pairs of functionally matched excitatory neurons in the MICrONS dataset. Response correlation is significantly larger for pairs of neurons with 4 or more synapses connecting them (n=11 pairs) compared to those with 1, 2, or 3 synapses (n=5350, 280, 34 pairs respectively).

differ from the results we would obtain if the same neurons were manually proofread. For any particular scientific question, researchers using these volumes will need to weigh the relative importance of precision, recall, and the number of neurons that it is feasible to proofread using manual or automated methods.

Thanks to the efforts of trained proofreaders at APL, we were able to perform extensive validation of a set of these rules "in situ", which enabled us to speed up semi-supervised splitting of multiple somas, and also enabled us to identify a subset of rules with high accuracy that could be batch-applied automatically to the current segmentation of the MICrONS volume. As far as we know the latter is by far the most extensive set of post-hoc edits generated by an automatic proofreading method that have actually been applied to improve the quality of an EM segmentation. Although EM segmentation methods continue to improve, post-hoc workflows for automated proofreading are likely to continue to be necessary given the amount of neuronal wiring in even a $1mm^3$ volume, and having an interpretable and tunable pipeline where we know what errors are easy or difficult for these methods to fix may even inform choices about what to optimize during the segmentation pipeline itself.

NEURD graphs are useful for cell typing. Previous work discusses differences in spine and synapse characteristics between cell type classes (Azouz et al., 1997; Kawaguchi et al., 2006). Based on these findings and our exploration with PCA, we find that the pre-computed features of spine density and shaft synapse density enable a linear classifier to distinguish these coarse cell classes using the same parameters for both the H01 and MICrONS dataset (Fig.4a,b). More fine-scale subclass classification can be performed with a simple GNN operating directly on the annotated NEURD graph (Fig.4c - f). Even higher performance may be achieved in the future with different classifier architectures. Similar approaches may also enable classification of orphan dendrite or axon segments within the EM volume, greatly expanding the scale of connectivity principles that can be explored in these datasets.

Highly-annotated NEURD graphs provide a compact representation of many features that are useful for all kinds of morphological analysis, making it possible to extract more of the incredible value that these large datasets offer for looking at morphological characteristics of neurons. For example, a simple query indicates that the percentage of pyramidal cells with axons protruding from dendrites (rather than the soma) is higher in the mouse volume (17.8%) than the human volume (8%), which closely replicates the findings of a previous study focused on this question (Wahle et al., 2022). Several of the morphological properties of cell types shown in Fig. 5j-m and Fig. 20 replicate observations from previous studies (Elabbady et al., 2022; Kawaguchi et al., 2006; Villa and Nedivi, 2016). Additionally, using the spine metrics extracted by NEURD, we were able to replicate many of the findings of (Harris and Stevens, 1989; Arellano et al., 2007) concerning synapse size and spine head volume correlation, and we also show that these scaling rules and others depend on cell type (Fig.5m, Fig. 23). Looking at how synapses onto the AIS and soma vary across species, we find the expected lower rate of soma synapses on human neurons than mouse (Wildenberg et al., 2021), and replicate the expected distribution of AIS synapses across depth in the mouse dataset (Schneider-Mizell et al., 2021). We also find that the human dataset does not show a similar change over depth in AIS synapses, and that the human dataset AIS is more densely innervated than in the mouse volume (Fig.5f-g, Fig. 21c). Finally, we demonstrate the use of a query combining geometric information and branch-level characteristics to replicate the previously-reported bias in the orientation of the thickest basal segment in the H01 dataset (Shapson-Coe et al., 2021). We extend this finding with an observation that this bias is actually consistent across all depths but is just less salient in upper layers because the relative size of the thickest and second-thickest basal dendrites changes smoothly across depth (Fig.5h-i).

At the level of the connectome, not only do we have access to a large number of high fidelity synapses and proximities but also pre-computed node (neuron) and edge (synapse) features, enabling a variety of connectomics analyses. For instance, using the cell type node labels and the skeletal walk length edge features, we confirmed previous work (Hwang et al., 2021; Megıas et al., 2001) describing different distal and proximal preferences for different excitatory and inhibitory connection types (Fig.6d-g). Furthermore, using the proximity controls computed on the cleaned skeletons, we also were able to observe

a consistent trend across datasets showing the propensity for forming connections from excitatory to inhibitory neurons peaks around 200 μm away from the soma, potentially consistent with a pattern of surround suppression (Fig.6h,i). Additionally, with the cell type node labels and spine compartment and synapse size edge labels, we confirmed a variety of expected findings about synaptic and spine head size: excitatory to excitatory connections have the largest synapses, synapse size correlates with spine heads for excitatory sources but not inhibitory sources, and spine heads with inhibitory synapses generally are multi-synaptic spines where the inhibitory synapse is typically much smaller than the largest synapse on the spine head (Fig. 24). With spine features (spine length, neck length, etc.) thoroughly documented throughout the volume and immediately available for queries, these properties can be investigated at the level of cell types and compartments to help understand the specificity of these synaptic features. Finally, by using proximities between cleaned skeletons to compute conversion rates as an estimate of connection probability, we replicate a variety of previous findings about the connection probability between both coarse (excitatory/inhibitory) and fine cell types (Jiang et al., 2015; Pfeffer et al., 2013; Schneider-Mizell et al., 2023). A variety of exciting questions about cell type specificity can be addressed with this kind of data.

We illustrate the power of combining automated proofreading to generate a clean but incomplete graph, with proximities to serve as a null distribution to account for this incompleteness. Together this powerful approach can begin to reveal principles of pairwise and higher-order connectivity motifs in dense reconstructions. We demonstrate our ability to take advantage of automated proofreading to identify a rare two-node motif (four or more synapses connecting two excitatory neurons) that would be buried in noise in the unproofread volume. This just illustrates one example of the potential for future discovery, especially given the matched functional properties available in the MICrONS dataset. Additionally, with the cleaned connectome, we were able to start computing the number of reciprocal and triangle motifs (the simplest of higher-order motifs) in the datasets, and we observe a general overexpression of these motifs in comparison to proximity controls and some standard null models, as previously reported in (Udvary et al., 2022; Song et al., 2005; Perin et al., 2011; Milo et al., 2002) However, the ability to expand this work to include cell type colorings of these motifs and add proximity based controls will enable investigation of more complicated motif questions, unleashing the power of these extraordinary datasets for a larger community of researchers.

In conclusion, this study is meant to illustrate the scientific impact of the diverse analyses that NEURD enables for these two EM volumes. A requirement for all these analyses is the automation of post-segmentation proofreading validated against extensive manual examples. This validation confirmed that NEURD eliminates the large majority of catastrophic false-positive neuropil and synapse locations that are the result of merge errors in the segmentation. The cleaned meshes output from the automated proofreading can be stored locally for analysis, but eventually we hope to use NEURD as the basis of a proofreading framework that can apply edits back into a dynamic segmentation graph for all users. We have made progress towards this goal on the MICrONS dataset, where approximately 150,000 edits suggested by NEURD have been applied back to the dynamic segmentation. In addition to proofreading, NEURD provides a novel holistic pipeline for neuron annotation spanning levels from spines (including spine head and neck labeling), neuronal compartments (i.e. apical trunk, basal etc), cell type-specific morphological analysis, and connectome-level features (like axon-dendrite proximities).

Other approaches to automated proofreading have been previously demonstrated, and few of the individual features computed by NEURD are novel on their own. The novelty of NEURD is the tight integration of proofreading and annotation in a single workflow that can be applied at scale. Designed for the next generation of massive, densely-segmented EM volumes, NEURD offers novel opportunities for scientific discovery by enabling complex queries across richly-annotated features at different scales. We demonstrate the power of this integration for morphological (Fig. 5, 21, 22, 23), and connectomic (Fig 6, 7, 23, 24, 25, 26, 27) analyses, most of which involved queries against pre-computed features, as well as new questions combining these features (21c, 21d, 6h,6i, 7d, 5m, 24g, 24h). As additional reconstructed volumes are released spanning species and brain areas, our ability to extract scientific insights will depend critically on integrated and scalable frameworks that make the features of neurons and networks accessible for analysis and also provide a rich feature space for future machine learning applications.

Methods

Data Management. For simplified data management and querying of input neuron reconstruction meshes, NEURD intermediate decomposition graphs, and all derived statistics and data products, we utilized the DataJoint Python package (Yatsenko et al., 2015, 2018)

Mesh Preprocessing. NEURD operates on 3-D meshes which are represented in a standard form as lists of vertices and faces in 3-D coordinates. A connected mesh component is a set of faces and vertices in which all faces have at least one adjacent edge to another face. Segmentation algorithms may not output a single connected component as a mesh, but instead may generate several disconnected submeshes, each of which is a subset of faces that is a connected component. NEURD is generally robust to discontinuous meshes, meshes of different resolutions, and several kinds of meshing errors.

The resolution of meshes delivered as part of the MICrONS and H01 datasets was sufficiently high that we performed an initial decimation of the mesh (reduced to 25% for MICrONS and 18% for H01) to speed up subsequent computations while retaining all the detail necessary for morphological characterization even of fine axons and spine necks. This decimation was

performed using the MeshLab Quadric Edge Collapse Decimation function (Cignoni et al., 2008). Following decimation, we separated this decimated mesh into connected components. We next applied a Poisson Surface Reconstruction (Cignoni et al., 2008) to each connected component. This can be thought of as "shrink-wrapping" the mesh - it smooths discontinuities on the surface of the mesh and ensures that each connected component is "water-tight" (i.e. no gaps or missing faces). This pre-processing facilitates the subsequent decomposition steps.

Glia, Nuclei Removal. Glia and nuclei submeshes are identified and filtered away using ambient occlusion functions (Cignoni et al., 2008) to identify regions with a high density of inside faces. Inside faces are mesh faces that are almost fully surrounded by other mesh faces. For example, glia that are merged onto neurons appear as cavities filled with a high density of mesh faces, and are distinct from the hollow reconstructions of most excitatory and inhibitory neurons. Similarly, the mesh representation of the soma surrounds the nucleus mesh and thus nuclei are almost entirely made up of inside faces. Therefore to identify glia and nuclei in the reconstructed meshes, we look for large connected components with high percentages of inside faces as candidates. To determine whether mesh faces are internal or external, we simulate an external "light source" that emits from all angles and we compute the exposure each face receives. This metric is thresholded to classify faces as either inside or outside faces, and submeshes made up mostly of inside faces are candidates for removal. We then apply additional thresholds on the candidate submeshes volume and number of faces to classify them as a glia mesh, nuclei mesh or neither. Finally, for glia we include all floating meshes within the bounding box of the submesh or within a search radius (3000 nm) of any faces of the submesh. This post-hoc mesh agglomeration serves to clean up the areas around glia segmentation, which can be very unconnected and non-standard.

Soma detection. Soma detection is run on any segment containing at least one detected nucleus (note that nucleus detection was performed previously as part of the segmentation and annotation workflow Consortium et al. 2021; Shapson-Coe et al. 2021). To detect the soma, we first perform a temporary heavy decimation of the mesh to remove small features and facilitate detection of the large somatic compartment. We then segment this low-resolution mesh into contiguous submeshes using the CGAL mesh segmentation algorithm (Yaz and Loriot, 2023). This function not only provides the specific faces in each submesh but also an estimate of the width of the submesh as an SDF value (Shape Diameter Function, a measure of diameter at every face of the mesh surface Yaz and Loriot 2023). We then filter all the resulting submeshes for soma candidates by restricting to those within a set size (number of faces), SDF range, bounding box length and volume threshold. We restrict to candidates that are sufficiently spherical to represent the general shape of a soma, but liberal enough to account for somas that are partially reconstructed, commonly observed at the edge of the volume. Once we identify candidate somas in the low-resolution mesh, we return to the original mesh representation (prior to starting soma detection) and apply a final size and width threshold. Given the initial restriction to segments with at least one detected nucleus, if we are not able to detect at least one soma after this process, we relax the thresholds slightly and iterate until a soma is detected or a threshold limit for number of attempts is reached.

Decomposition. With the glia and nuclei submeshes identified, we filter those away from the original mesh, which may cause splitting into additional connected components. We identify connected components containing at least one soma submesh (note that some segmentations may contain multiple somas prior to soma splitting). Mesh fragments that are not connected to somas may be floating meshes inside the soma (which are filtered away using the same ambient occlusion methods described for glia above), or detached mesh pieces of neural processes that can be stitched to the neuron representation later. For each of the soma-containing meshes we filter away the soma submeshes and identify connected components of these meshes as stems. Any stem submesh must contain at least one set of connected adjacent edges and common vertices shared with a soma submesh ("border vertices"). We construct a connectivity graph where edges only exist between stem and soma nodes if there exists border vertices between the stem and the soma. Through the graph constructed in this manner, stems that provide paths between multiple somas can be easily identified for subsequent splitting (see below).

Each of the stem submeshes is then processed into a skeleton - a 3-D "line-segment" representation that is a set of vertices and edges. We use the Meshparty package for this initial round of skeletonization because of its efficient implementation, and because it provides both a width estimate and a correspondence between the faces of the original mesh and each vertex in the skeleton (Dorkenwald, 2022). This skeleton is then further divided into branches (non-branching subskeletons). The corresponding meshes of branches with an average width greater than a threshold are re-skeletonized with a higher-fidelity method that yields skeletons which pass through the hollow centers of the mesh to provide a better estimation of the location and curvature of the surrounding mesh. This is particularly important for some neural processes, for example wide apical trunks where a skeleton that is not centered within the mesh could be displaced nearly a μm from the actual center of the trunk. This higher-fidelity method is performed using the CGAL Triangulated Surface Mesh Skeletonization algorithm (Gao et al., 2023).

For mesh correspondence and width determinations of all skeletons, the NEURD algorithm employs a custom mesh correspondence workflow based on the following steps: First, each non-branching segment of the skeleton is divided into smaller pieces, and for each piece a cylindrical search radius is used to identify the mesh correspondence. The closest distance between the skeleton and corresponding mesh faces is computed at multiple points along the skeletal segment, and these are averaged to get a mean radius. Finally, all mesh correspondences of sub-branches are combined into the mesh correspondence of the

branch. Concatenating the widths along the sub-branches forms a width array along the branch, with the entire branch width determined from the average of the array. This method results in one face of the original branch possibly corresponding to more than one branch's mesh correspondence, so the algorithm employs a final graph propagation step from unique mesh correspondence faces to allow branches to claim the previously conflicting faces.

The procedure described above yields a collection of disconnected non-branching skeletal segments as well as their associated mesh correspondence. The skeleton of finer-diameter processes is the initial MeshParty Dorkenwald (2022) skeleton which tracks along the mesh surface, while larger-diameter processes have skeletons that track through the center of their volume. These pieces are then all stitched together into a single connectivity graph where each non-branching segment is a node, and the edges between them represent their connectivity. Any conflicts in the mesh correspondence of adjacent nodes at stitching points is again resolved, yielding a smooth and connected mesh representation of the entire stem where each mesh face is associated with a single node (non-branching segment). This entire process is repeated for every stem submesh connected to the soma. Finally, all floating meshes outside the soma are decomposed in the same manner as the stems, and then appended to the existing skeleton if they have any faces within a threshold distance of another node (for example, in the MICrONS dataset the maximum stitch distance was set at 8 μm).

The soma(s) and decomposed stems of a neuron are then represented as a NetworkX graph object (Hagberg et al., 2008). In the ideal scenario there is a single soma root node with multiple stem subgraphs, and each stem subgraph is a directed tree structure representing the connectivity between non-branching segments of the skeleton from the most proximal branch connecting to the soma to the most distal leaves of the axonal or dendritic process. In less ideal cases (which are common), cycles may exist in the skeleton due to self-touches of the axonal or dendritic process (close proximities of neurites that are incorrectly meshed together), and multiple somas may be included in a single segmented object. Handling of these cases is described in "Multi-Soma and Multi-Touch Splitting" below. The soma node contains the soma submesh and SDF values generated during the soma extraction, and each branch node in each stem stores the mesh correspondence, skeleton and width array for that node. Using these raw features, many more features of these branches can be extracted (for example spines), and additional annotations can be added (such as synapses).

Spine Detection. The non-branching segments produced by the mesh decomposition of each node present an ideal scenario for spine detection. Briefly, we started by using an existing mesh segmentation algorithms (Fabri and Rineau, 2023) which calculates a local estimate of the volume for each face of the mesh (SDF), applies a Gaussian Mixture Model (GMM) to the distribution of SDF values to enable a soft clustering of faces to k clusters, and finally minimizes an energy function defined by the alpha-expansion graph-cut algorithm to finish with a hard cluster assignment over the mesh. This last step takes a smoothness parameter controlling the likelihood that adjacent faces with concave edges will be more or less likely to be clustered together. We found that setting the number of clusters to 5 and smoothness value to 0.08 was optimal for both the MICrONS and H01 dataset to produce an over-segmentation of the branch mesh to serve as input for the next spine detection step. Then, we convert the branch segmentation into a graph representation (branch segmentation graph) where the nodes are submeshes of the segmentation and edges exist between submeshes with adjacent faces. The dendritic shaft subgraph is determined by establishing the longest contiguous shaft line path in the graph (from most likely node candidates defined by size, width and diameter thresholds), and then spine candidates are identified as subgraphs (not in the shaft path) based on size, volume, and distance from the mesh surface. The final product of this stage is the individual spine submeshes and then subsequently calculated spine statistics (volume, length, area). Based on these statistics, in some cases we perform an additional processing step that divides the submesh of larger spines into a spine head and spine neck. At the completion of this workflow, each mesh face in the node receives a spine label of head, neck, shaft or just "spine" (if no head and neck segmentation could be performed). Finally, the width of each branch is recomputed after removing spines that may have previously confounded that measurement. A step by step tutorial of how to optimize the parameters (and a complete explanation of each parameter available for tuning) for the spine detection and head/neck segmentation are included in the tutorials of the Github repository online.

Synapse Addition. Synapses from the reconstruction pipeline are mapped to the closest mesh face of the closest branch. Any annotations of the associated face (for example) spine head, spine neck or shaft can then be propagated to the synapse. In addition, the closest skeletal point on the associated branch object is computed to define an anchor point for the synapse on the neuron's skeleton. This anchor point enables computation of metrics such as skeletal walk distance to the closest spine, closest neurite branch point (upstream or dowstream) or skeletal walk distance to the soma.

Multi-Soma and Multi-Touch Splitting. After the initial decomposition, cycles may exist in the stem graphs due to self-touches in the decomposed mesh (regions where neurites pass very close to each other, resulting in inappropriate connectivity of faces in the mesh representation). Furthermore, stems may include edges with multiple somas if two or more somas are merged together in the same mesh object. This is a challenging problem that requires a general solution since stems can be both multi-touch and multi-soma of any degree. For example, some apical stems of neurons in the MICrONS dataset were initially connected to 9 or more somas due to close mesh proximities with apical tufts of other cells. The aim of this stage in

the processing pipeline is to split the stem objects optimally, while attributing the correct portion of the stem mesh and skeleton to the correct neurite.

The workflow for splitting both multi-touches and multi-somas proceeds as follows: For every stem identified as having a multi-touch or multi-soma connection, the process first starts by identifying cyclic or soma-to-soma paths. The best edge to cut on the path is then determined using a series of heuristic rules that are applied in the order listed below:

1. Break any edge on the multi-touch or multi-soma path where the angle between the skeleton vectors of two adjacent branches on the path is larger than some threshold (reflecting the fact that neurite processes generally do not typically abruptly double-back).

2. Break any edge on the multi-touch or multi-soma path where more than two downstream branches exist at a branch point and the best match for skeletal branch angle and width is not on the multi-touch or multi-soma path.

3. Break any edge on the multi-touch or multi-soma path where there is a difference in width between two nodes along the path greater than a threshold amount.

After an edge is removed based on any of these rules, the process restarts and the graph representation is checked again to see if cyclic or multi-soma paths still exist. The process is repeated until no such paths exist. If no candidate edge is identified by using these rules then, depending on the user settings, the the stem may be completely discarded from the neuron object or cut at the very first or last branch.

This splitting algorithm is not guaranteed to optimally split all multi-soma or multi-touch paths, but residual errors from a sub-optimal split may be cleaned by further proofreading steps. As with any automated proofreading, the rule and relevant parameters that determined the edit are stored for subsequent evaluation and use.

Excitatory/Inhibitory Classification. Once each neuron object has a single soma, the NEURD workflow moves on to an initial round of coarse cell classification, determining whether each neuron is excitatory or inhibitory. Performing the classification at this point in the workflow enables the use of subsequent proofreading or annotation algorithms that are excitatory- or inhibitory-specific. For example, axon identification (see below) is much easier if the coarse E/I type of the cell is known beforehand. The cell class is determined via logistic regression on two features: postsynaptic shaft density (number of synapses onto dendritic shafts per μm of skeletal distance on the postsynaptic dendrite) and spine density (number of spines per μm of skeletal distance on the postsynaptic neuron). These two features enabled linearly-separable elliptical clusters for excitatory and inhibitory cells. To enable this classification prior to proofreading, we applied two restrictions to the unproofread graph that reduce potential confounds due to merge errors and ambiguity between axon and dendrite. First, we restrict to larger dendrites using a simple width threshold to not include potential orphan axon merges, and second, we restrict to the proximal dendrite within a limited skeletal walk distance from the soma center. The latter reduces confounds due to dendritic merge errors which are more common at the distal branches (Fig. 3d). When compared to human E/I labels, the classifier results are shown in Fig. 4a,b. These results are also robust against an approximate 10:1 and 1.8:1 excitatory to inhibitory class imbalance in the labeled MICrONS and H01 datasets respectively. Additionally, because a logistic regression is the classifier, there is a confidence score associated with the classification (which could be thresholded downstream to ensure higher fidelity labels).

Non-Neuronal Filtering. While all the neurons in the H01 dataset were hand-checked as neurons and manually annotated for cell types, the MICrONS dataset initially was not. Consequently, segments with nuclei in the MICrONS dataset could include blood vessels, glia cells, or agglomerations of orphan axons without a neuron mesh due to an incorrect nucleus merge, and we did observe this frequently in the version of MICrONS processed in this study, version 374 (but this issue is now largely resolved in the most current data release with tables that indicate which segments are predicted as non-neuronal, using a method independent from NEURD, and with more accurate nuclei merging). Therefore, in order to filter away almost all of these segments without sacrificing a significant amount of valid neurons, we found a suitable filter using cell type classification (predicted by our logistic regression model), number of soma synapses and the mean dendritic branch length (this may very well be a MICrONS-specific filter). Specifically, the filter excluded the following: all segments with less than 3 soma synapses, excitatory cells with less than 17 soma synapses and a mean dendritic branch length less than 35 μm, and inhibitory cells with less than 17 soma synapses and a mean dendritic branch length less than 28 μm. This filter then removed approximately 14,000 segments from all of our downstream analysis.

Axon Identification. The goal of this stage in the pipeline is to identify at most one connected component subgraph that represents the axon of the cell. In the absence of merge and split errors, identifying the axon would be a simple process of identifying the subgraph with presynaptic connections, but un-proofread datasets pose a number of challenges to this approach.

1. Due to partial reconstruction of cells, the true axon may not exist or only the axon initial segment (AIS) may exist. In this case there would be no true presynaptic connections from the cell.

2. Postsynaptic synapses on dendritic segments may be incorrectly labeled as presynaptic connections if the synapse classifier is incorrect.

3. Orphan axons may be incorrectly merged onto the cell's dendrite or soma. These frequent merge errors add incorrect presynaptic connections onto dendrites that make identifying the true axon subgraph more difficult. Anecdotally, if the algorithm simply chose the connected component subgraph with the highest presynaptic density, this would almost always be an orphan-axon-onto-dendrite merge error.

Our approach to axon identification is thus motivated by the following neuroscience "rules" which we implement as heuristic selection criteria. Note that in this and following sections "up" or "higher" refers to the pial direction, while "down" or "lower" refers to the white matter.

1. Axons can either project directly from the soma or from a proximal dendritic branch.

2. The axon is the only compartment (possibly excluding the soma) that forms presynaptic connections.

3. The width of axon segments are typically thinner than most dendritic branches

4. Axon segments do not have spines (although boutons may have similar features)

5. Axons receive postsynaptic inputs at the AIS, but these typically have low postsynaptic density compared to dendrites. (Note that we found the latter was not necessarily true in the H01 dataset and adjusted accordingly.)

6. For excitatory cells, the axon typically projects directly from the soma or from dendritic stems that originate from the deeper half of the soma.

7. For excitatory cells, the AIS starts at most 14 μm skeletal distance from the soma and the general skeleton vector of the AIS typical projects downwards. The AIS does not split into multiple branches close to the soma.

8. For inhibitory cells the AIS can start much farther away (up to 80 μm skeletal distance from soma) and can come off the soma or dendritic branch. The inhibitory AIS has very low postsynaptic and presynaptic density.

NEURD identifies candidate axonal submeshes based on a combination of these heuristics applied in a cell-type-specific manner. For example, if the neuron being analyzed is excitatory, the search space of potential axonal stems is restricted to only those with a projection angle from the soma greater than 70 degrees relative to the top of the volume. Candidate AIS branches must fall within a maximum and minimum width range, they must have a synapse density below a threshold value, and they must be within a threshold skeletal distance from the soma that dictated by the cell type. If multiple potential candidates exist, the best potential axonal subgraph is selected based on longest skeletal length and closest proximity to the soma. Subgraphs that meet the heuristic criteria for axons but are not chosen as the actual axon of the cell in question retain a label of "axon-like" which facilitates subsequent proofreading.

An additional round of skeletonization is performed once the axon is detected. This re-skeletonization better captures fine details of the axonal trajectory and enables auto-proofreading methods to catch more subtle axon-to-axon merge errors.

Automatic Proofreading. The goal of the automatic proofreading stage is to identify merge errors and remove all downstream branches (split errors resulting in missing portions of the axon or dendrite are not addressed by this process). NEURD implements a series of heuristic proofreading rules to identify merge errors based on graph filters - configurations of nodes and attributes that typically indicate merge errors. The graph filters are either directed one-hop or zero-hop configurations where one-hop configurations consider a node and its immediate adjacent nodes and zero-hop configurations consider only the node's features itself. These graph filters have parameters that can be tuned for axons or dendrites, excitatory or inhibitory cells, or different data sources (H01 vs MICrONS). For example, a graph filter for resolving graph configurations of dendritic branches with 3 or more downstream nodes is useful for the H01 dataset which has a higher probability of dendritic merge errors than MICrONS. For each filter, the algorithm finds all branches that match the current error motif, and then the mesh, graph nodes and synapses associated with all those error branches are removed before proceeding to the next filter. Therefore, in the current successive order there is no overlap in error locations for different filters in the same run. Metadata for each correction is stored for subsequent review or for training non-heuristic models.

The following graph filters exist for proofreading axon submeshes. Those only used for excitatory cell types are indicated.

1. High Degree Branching: The filter identifies any node (below a potential width threshold to exclude myelinated sections) with more than two downstream nodes. The filter assumes this configuration results from a single or multiple crossing axon(s) merged at an intersection point, adding 2 or more additional downstream nodes. The filter aims to identify the one true downstream node. Possible upstream to downstream node pairings are filtered away if the width, synapse density or skeletal angle differ by threshold amounts. If multiple downstream nodes are viable, the algorithm attempts to find a

downstream candidate with the best match of skeletal angle or width. If no clear winner exists, the algorithm can either mark all downstream nodes as errors if the user wishes to be conservative, or can pick the best skeletal-match candidate. There are more rare scenarios where a myelinated axon has 2 collateral projections protruding very close to one another and these would be incorrectly filtered away, but a large majority of these occurances are simply merge errors.

2. Low Degree Branching Filter: The filter processes any subgraph with one upstream node and exactly two downstream nodes and is only attempted on non-myelinated axon sections (as determined by a width threshold). The method attempts to find one of the following subgraph features within this directed 3 node subgraph, and if a match occurs either the algorithm marks all of the downstream nodes as errors or attempts to determine the correct downstream pairing.

   (a) **Axon Webbing**: An error is detected by an overly-thin mesh at the branching point of an upstream to downstream node. The filter attempts to differentiate between natural branching with cell membrane that forms a "webbing like" appearance as opposed to merge errors where no such thickening occurs.

   (b) **T-Intersection**: An error is detected by the presence of downstream branches that are thicker than an upstream branch and the downstream branches are aligned and resemble a continuous non-branching axon segment.

   (c) **Double-Back** (excitatory only): An error is detected when a downstream node "doubles-back" towards the upstream node at an unnatural skeletal angle.

   (d) **Parallel Children (or Fork Divergence)**: An error is detected when the two downstream skeletons are nearly parallel without a natural gap between them.

   (e) **Synapse At Branching**: An error is detected if a synapse occurs very close to the branch point between upstream and downstream nodes; this usually indicates a merge of an orphan axon to a bouton segment.

3. Width Jump: Processes a subgraph with any number of downstream nodes (only applies to non-myelinated sections). Any downstream node with an absolute width difference between segments above a certain threshold is marked as an error.

The following graph filters exist for proofreading dendrite submeshes.

1. Axon on Dendrite: Nodes that were previously labeled “axon-like” during the process of axon identification (see above) that do not end up in the axon submesh are marked as errors.

2. High Degree Branching (H01 only, excitatory only, apical trunks excluded): An error is detected using the same algorithm as described above for the axonal High Degree Branching Filter except it is applied to dendritic nodes below a thresholded width.

3. Width Jump: An error is detected using the same graph filter as described for axons but with larger width difference thresholds

4. Double Back: An error is detected using the same algorithm and parameters as for axons above.

APL Validation of Multi-Soma Splitting and High Confidence Orphan Merge Edits. Our collaborators at APL (Johns Hopkins University Applied Physics Laboratory) helped extensively validate multiple aspects of the NEURD automated proofreading workflow. In particular, they provided information about the following:

1. Validation of specific edits in the context of multi-soma splitting.

2. Data about the time that proofreaders took to evaluate these split suggestions compared to other methods.

3. Validation of specific edits focused on axon-onto-dendrite or axon-onto-axon merges

This information made it possible to determine whether our suggestions for multi-soma split locations speed up the process (they do, more than three-fold), and if we could identify a set of heuristics and parameters for axon-on-dendrite and axon-on-axon merge corrections that could be executed with high confidence on the entire volume without human intervention (we identified two classes of edits with performance >95%, and more than 150,000 of them have been applied to the MICrONS volume to date; see Fig. 10).

A key method for both validating and applying automatic edits was the functionality in Neuroglancer (Perlman, 2019) which allows the placement of point annotations to define a split in the PyChunkedGraph segmentation (Dorkenwald et al., 2022a,b). To facilitate the proofreading process, APL created a web-based interface called NeuVue (Xenes et al., 2022) that allows for the efficient queuing, review and execution of split suggestions in Neuroglancer. We built the logic required to translate mesh errors identified by NEURD into split point annotations that can be executed by the NeuVue pipeline. This capability allowed proofreaders at APL to not only evaluate error locations identified by NEURD, but also a proposed set of points that could

be subsequently executed in the PyChunkedGraph to correct the error. For a more detailed description of the NeuVue review pipeline see (Xenes et al., 2022).

For multi-soma split edits, we generated point annotations for suggested splits that would contribute to separating neurons with between 2 and 6 possible somas in a single segment. APL had both experts and trained student proofreaders review these edits. The classifications for each of the edits was one of the following:

1. "yes": same split point annotations the proofreader would have chosen.
2. "yesConditional": split point is correct, but point annotations required very minor adjustment.
3. "errorNearby": split point is not correct but is very close by, and split point annotations require adjustment
4. "no": the correct split location was not at or near the suggested location

The expert proofreaders reviewed 5134 unique suggestions with no overlap between proofreaders, while the student proofreaders reviewed 2355 suggestions with some redundancy so the same suggested edit was seen by multiple student proofreaders and a majority vote determined the classification of the edit. The results of reviewing the first approximately 4000 of those edits are shown in Fig.10a and the accuracy was determined to be 76.12% when the "yes","yesConditional" and "errorNearby" categories were considered true positive classes. Additionally, because each split suggestion had an associated heuristic rule and set of parameters that was used to generate the suggestion, we were able to show that some rules were much higher fidelity than others, and that the parameters could be tuned to achieve a higher classification accuracy (Fig.10b,c).

To compare against the performance achieved with NEURD suggestions, multi-soma splits were also performed by expert proofreaders using a tool that highlighted the path along the neural processes connecting two somas. This comparison enabled us to measure if the NEURD suggestions could potentially speed up the soma-splitting process. Because a single segment with a multi-soma merge could contain more than 2 somas and because different merges may require a different amount of work and number of cuts to be applied, we measured the overall time spent reviewing all edits and estimated the additional time that would have been required to make the slight adjustments required in the case of "yesConditional" (+30s) or "errorNearby" (+60s), prior to executing the edit. Note that "no" classifications added time to the review process without any possibility of contributing to an actual edit. Based on these metrics, we divided total time by total number of edits to determine the mean time per edit. We then compared this metric to the mean time per edit when proofreaders used a standard pathfinding tool that displayed the skeletal path connecting multiple somas, and they had to search along this path to identify errors manually. We observed a more than three-fold speed up when using NEURD suggestions (Fig.10d).

Finally, outside the context of multi-soma splitting, APL proofreaders evaluated two kinds of merge error corrections that strip orphan axons from both excitatory and inhibitory neurons: axon-on-dendrite, and high degree axon-on-axon. The feedback on each error from proofreaders using the NeuVue pipeline allowed us to determine a subset of parameters that was correlated with high accuracy. Thresholds for the axon-on-dendrite included minimum parent width, distance from the soma, and skeletal length of the error segment. Thresholds for axon-on-axon included a minimum skeletal length, and a branching pattern that resembled a two line segment crossing, where the segments are closer to perpendicular in order to make the correct connectivity more obvious. For the review of orphan merge errors, an additional label was included in the true positive class: "yesPartial", which indicated that part but not all of the merge was removed by the split point annotations. The feedback from this effort provided our collaboration with enough evidence to then apply nearly 150,000 of these high-confidence automatic edits back into the current dynamic segmentation of the MICrONS dataset (Fig.10e).

Automatic Compartment Labeling. After automatic proofreading removes as many merge errors as possible, compartment labeling is performed for excitatory cells, classifying graph submeshes as apical trunk, apical tuft, basal, and oblique. NEURD first attempts to identify the apical trunk based on the geometry relative to the soma and total skeletal length. Branches downstream of the end of the apical trunk are classified as the apical tuft, and branches off of the trunk with a skeletal angle close to 90 degrees are labeled as oblique. If the criteria for a defined trunk is not met, then NEURD applies a generic "apical" label. Other dendrites are classified as basal. Additionally, for the MICrONS dataset if the soma center is close enough to the pia as defined by a depth threshold, there can be multiple generic "apical" stems protruding from the top of the soma if they each meet the required width and geometry thresholds.

Connectome-level features computed by NEURD. At the level of the connectome graph, nodes represent individual single-nucleus neurons and edges represent synaptic connections. In addition to the rich sub-cellular features that NEURD computes for the decomposition graph of each cell, NEURD provides a variety of features at the connectome graph level:

1. Node Attributes: a wide range of global properties measured for the individual cells (compartment skeletal lengths, synapses, bounding box, spine densities, synapse densities, average width, cell type, etc).
2. Edges: connections between neurons with a valid presynaptic connection and postsynaptic connection where neither were filtered away in the auto-proofreading stage

3. Edge Attributes: properties for each of the presynaptic and postsynaptic neurons (compartment, skeletal/euclidean distance to neuron's soma, size, spine label) and properties of the entire synaptic connection between neurons (euclidean/skeletal distance from soma of presynaptic neuron to soma of postsynaptic neuron, etc).

GNN Cell Typing. Using PyTorch geometric software (Fey and Lenssen, 2019) we implemented a Graph Neural Network architecture to build a supervised cell type classifier (including subclasses of excitatory and inhibitory cells) from dendritic graph structure in the NEURD decompositions. We trained this classifier using manual cell types from the Allen Institute for Brain Science (Schneider-Mizell et al., 2023). To create an input graph for the classifier we first removed the soma node and filtered away the axonal subgraph and any dendritic stems with less than 25 μm of total skeletal length. Each node was annotated with the following feature set:

1. Skeleton features, where theta and phi refer to polar coordinates of the skeleton vector in 3-D (skeleton_length, skeleton_vector_upstream_theta, skeleton_vector_upstream_phi, skeleton_vector_downstream_theta, skeleton_vector_downstream_phi)

2. Width features (width, width_upstream, width_downstream synapse)

3. Spine features (n_spines, spine_volume_sum, n_synapses_post, n_synapses_pre, n_synapses_head_postsyn, n_synapses_neck_postsyn, n_synapses_shaft_postsyn, n_synapses_no_head_postsyn, synapse_volume_shaft_postsyn_sum, synapse_volume_head_postsyn_sum, synapse_volume_no_head_postsyn_sum, synapse_volume_neck_postsyn_sum, synapse_volume_postsyn_sum)

For the whole neuron classifier, the soma volume and number of soma synapses for the neuron are added to each node's feature vector and also the starting stem angle (2-D angle between the vector from the soma center to the stem's root skeleton point and the vector in the direction of the pia) is added to each node in every stem. For the stem-based classifier, these three soma features are not included, and classification is performed on each stem individually.

The GNN architecture used as a 2 layer Graph Convolutional Network (128 hidden units for each layer, ReLU activation function) followed with one linear layer. The aggregation and update steps were implemented using self loops and symmetric normalization as shown here:

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}^{(k)} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}}\right)$$

where $\mathbf{h}_u^{(k)}$ is the embedding for node u at layer k, $\mathcal{N}(u)$ are the neighbors for node u, $\mathbf{W}^{(k)}$ is the learned weight matrix at layer k and σ is the chosen non-linearity. For graph pooling (to get one learned vector for each graph), the weighted average of all nodes after the final hidden layer was taken (weighted by the skeletal length of the node). A 60%, 20%, 20% split for training, validation and test sets was used for labeled datasets of n = 873 whole neurons and n = 4,114 stems (Fig.19)

Proximities. Identifying axon-dendrite proximities makes it possible to determine how often a pair of neurons capitalizes on an opportunity to form a synaptic connection. Proximities are regions where the axon of one neuron passes within a few microns of the dendrite of another neuron. They can be annotated with the same features (dendritic compartment, neural subtype) as synapses, regardless of whether a connection was formed (Fig. 6a). Proximities are identified for all neuron pairs in the volume. To reduce the number of pairwise computations, NEURD first checks whether the bounding box of the presynaptic axon skeleton and postsynaptic dendrite skeleton have any overlap. In order to reduce computation time in the the MICrONS dataset, presynaptic neurons are further restricted to those with at least five axonal synapses, and in the MICrONS volume postsynaptic neurons are restricted to neurons with at least 1 mm of dendritic length (this restriction excludes approximately 1% of all MICrONS neurons).

The proximity calculation is performed by converting the axonal skeleton of the presynaptic neuron and the postsynaptic skeleton to an array of coordinates without edges (at one μm skeletal walk resolution). A local width and compartment label is associated with every point, and the soma of the postsynaptic neuron is converted to a uniform sampling of the surface mesh face centers for its skeletal representation. The cleaned synaptic connections between the pre and post neuron are retrieved (if there were any) and then the main proximity loop begins.

1. The closest distance between a presynaptic coordinate and postsynaptic coordinate is computed (where the distances can be adjusted for postsynaptic width by subtracting the local width from the euclidean distance). This minimum distance is the current proximity distance. If the current proximity distance exceeds the thresholded maximum proximity distance (set at 5 μm for both the MICrONS and H01 dataset), then the loop is exited and no more proximities are computed, but otherwise the workflow proceeds.

2. The following metrics are computed or collected for each proximity: The presynaptic and postsynaptic coordinate of the minimum distance pair, the distance between these coordinates (the proximity distance), the compartment labels and width at the postsynaptic coordinate, the presynaptic and postsynaptic skeletal walk distance, the number of spines and synaptic connections within a three μm radius of the presynaptic and postsynaptic coordinates.

3. After these features are collected, the skeleton points within a set radius (10 μm) of the presynaptic proximity coordinate are filtered away from the array of axon presynaptic coordinates.

4. All proximity information is saved and the loop continues until the current proximity distance exceeds the threshold.

Functional Connectomics. We considered pairs of synaptically-connected functionally-matched cells available in the MICrONS dataset, restricting to pairs where both neurons met a minimal set of functional quality criteria (test score greater than 0.2 and an oracle score (correlation of a neurons response to the leave-one-out mean response across a repeated image stimuli) greater than 0.3, see Ding et al. 2023; Wang and Tolias 2023). Synaptic connections were discarded if they were not onto post-synaptic spines (to help guard against possible inhibitory merge errors resulting in increased connectivity between neurons). We then divided the pairs into groups based on whether they had 1, 2, 3 or 4+ synapses between them. The final number of functionally matched pyramidal pairs available from automatic proofreading alone were as follows: 1 synapse (5350), 2 synapses (280), 3 synapses (34) and 4+ synapses (11). We then investigated how the mean functional response correlation varies as a function of the four different multi-synaptic groups. The response correlation was calculated as detailed in (Ding et al., 2023; Wang and Tolias, 2023) through the *in silico* response correlation of their model.

Analysis Subsets and Distributions. All morphological, connectomic and functional connectomic analyses excluded the manually proofread neurons used for validating NEURD proofreading, a small fraction of cells that errored out during the NEURD preprocessing pipeline due to a variety of factors (corrupted mesh, manifold and watertight properties unable to be automatically fixed, etc.), and non-neuronal cells that were filtered away as described in the "Non-Neuronal Filtering" section. Additionally, we excluded some neurons with very incomplete reconstructions from some analyses. For example, only a subset of automatically proofread neurons with an axon length longer than 50 μm were used in the investigation of questions concerning synapses onto the axon initial segment, in order to not skew results due to a reconstruction bias. Complete n's for all analyses and different stages of the pipeline are included in the Appendix Table 2.

AUTHOR CONTRIBUTIONS

We use the CRediT system for author roles. Conceptualization: BC, JR. Methodology: BC, SP, JR. Software: BC. Validation: BC, DX, LMK, PKR, VAR, CAB, BW, FC. Formal analysis: BC. Investigation: BC. Resources: BC, SP, ZD, PGF, EW, CP, AK, SP, JAB, ALB, DB, JB, DJB, MAC, EC, SD, LE, AH, ZJ, CJ, DK, NK, SK, KL, KL, RL, TM, GM, EM, SSM, SM, BN, SP, CMSM, WS, MT, RT, NLT, WW, JW, SCY, WY, EF, EYW, FHS, HSS, FC, NMdC, RCR, XP, AST, JR. Data Curation: BC. Writing - Original Draft: BC, JR. Writing - Review Editing: BC, SD, CMSM, FC, NMdC, AST, JR. Visualization: BC, JR. Supervision: JR. Project administration: JR. Funding acquisition: HSS, NMdC, RCR, AST, JR.

ACKNOWLEDGEMENTS

The authors thank David Markowitz, the IARPA MICrONS Program Manager, who coordinated this work during all three phases of the MICrONS program. We thank IARPA program managers Jacob Vogelstein and David Markowitz for co-developing the MICrONS program. We thank Jennifer Wang, IARPA SETA for her assistance. The work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DoI/IBC) contract numbers D16PC00003, D16PC00004, D16PC0005, and 2017-17032700004-005. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. ABK acknowledges support from a training fellowship from the Gulf Coast Consortia, on the NLM Training Program in Biomedical Informatics & Data Science (T15LM007093). XP acknowledges support from NSF CAREER grant IOS-1552868. XP, ABK, and AST acknowledge support from NSF NeuroNex grant 1707400. AST also acknowledges support from the National Institute of Mental Health and National Institute of Neurological Disorders And Stroke under Award Number U19MH114830 and National Eye Institute award numbers R01 EY026927 and Core Grant for Vision Research T32-EY-002520-37. BC and JR are supported by RF1 MH130416 to AST, XP, JR. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

Software. Experiments and analysis are carried out with custom built data pipelines. The data pipeline is developed in Python with the following tools: DataJoint used for storing and managing data. Meshparty, CGAL and MeshLab were used for mesh processing (CGAL and MeshLab required python wrappers). DotMotif and NetSci was used to help query graph motifs. Numpy, Pandas, SciPy, Scikit-learn, and PyTorch were used for model training and statistical analysis. Matplotlib, Seaborn, Ipyvolume, and Neuroglancer were used for graphical visualization. Jupyter, Docker, and Kubernetes were used for code development and deployment.

Data availability.

All MICrONS data have already been released on BossDB (https://bossdb.org/project/microns-minnie, please also see https://www.microns-explorer.org/cortical-mm3) for details.

Code availability.

Code, documentation, and example tutorials are available at https://github.com/reimerlab/NEURD.git

Conflict of Interest. XP is a co founder of Upload AI, LLC, a company in which he has financial interests. AST is co founder of Vathes Inc., and UploadAI LLC companies in which he has financial interests. JR is co founder of Vathes Inc., and UploadAI LLC companies in which he has financial interests.

Bibliography

Alexander Shapson-Coe, Michał Januszewski, Daniel R. Berger, Art Pope, Yuelong Wu, Tim Blakely, Richard L. Schalek, Peter Li, Shuohong Wang, Jeremy Maitin-Shepard, Neha Karlupia, Sven Dorkenwald, Evelina Sjostedt, Laramie Leavitt, Dongil Lee, 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 connectomic study of a petascale fragment of human cerebral cortex. Technical report, bioRxiv, May 2021.

MICrONS Consortium, J. Alexander Bae, Mahaly Baptiste, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Brendan Celii, Erick Cobos, Forrest Collman, Nuno Maçarico da Costa, Sven Dorkenwald, Leila Elabbady, Paul G. Fahey, Tim Fliss, Emmanouil Froudarakis, Jay Gager, Clare Gamlin, Akhilesh Halageri, James Hebditch, Zhen Jia, Chris Jordan, Daniel Kapner, Nico Kemnitz, Sam Kinn, Selden Koolman, Kai Kuehner, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Sarah McReynolds, Elanine Miranda, Eric Mitchell, Shanka Subhra Mondal, Merlin Moore, Shang Mu, Taliah Muhammad, Barak Nehoran, Oluwaseun Ogedengbe, Christos Papadopoulos, Stelios Papadopoulos, Saumil Patel, Xaq Pitkow, Sergiy Popovych, Anthony Ramos, R. Clay Reid, Jacob Reimer, Casey M. Schneider-Mizell, H. Sebastian Seung, Ben Silverman, William Silversmith, Amy Sterling, Fabian H. Sinz, Cameron L. Smith, Shelby Suckow, Marc Takeno, Zheng H. Tan, Andreas S. Tolias, Russel Torres, Nicholas L. Turner, Edgar Y. Walker, Tianyu Wang, Grace Williams, Sarah Williams, Kyle Willie, Ryan Willie, William Wong, Jingpeng Wu, Chris Xu, Runzhe Yang, Dimitri Yatsenko, Fei Ye, Wenjing Yin, and Szi-chieh Yu. Functional connectomics spanning multiple areas of mouse visual cortex, August 2021. Pages: 2021.07.28.454025 Section: New Results.

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

Jingpeng Wu, William M. Silversmith, Kisuk Lee, and H. Sebastian Seung. Chunkflow: hybrid cloud processing of large 3D images by convolutional nets. *Nature Methods*, 18(4):328–330, April 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01088-5. Number: 4 Publisher: Nature Publishing Group.

Ran Lu, Aleksandar Zlateski, and H. Sebastian Seung. Large-scale image segmentation based on distributed clustering algorithms, June 2021. arXiv:2106.10795 [cs].

Thomas Macrina, Kisuk Lee, Ran Lu, Nicholas L. Turner, Jingpeng Wu, Sergiy Popovych, William Silversmith, Nico Kemnitz, J. Alexander Bae, Manuel A. Castro, Sven Dorkenwald, Akhilesh Halageri, Zhen Jia, Chris Jordan, Kai Li, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, William Wong, Szi-chieh Yu, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Erick Cobos, Forrest Collman, Leila Elabbady, Paul G. Fahey, Emmanouil Froudarakis, Daniel Kapner, Sam Kinn, Gayathri Mahalingam, Stelios Papadopoulos, Saumil Patel, Casey M. Schneider-Mizell, Fabian H. Sinz, Marc Takeno, Russel Torres, Wenjing Yin, Xaq Pitkow, Jacob Reimer, Andreas S. Tolias, R. Clay Reid, Nuno Maçarico da Costa, and H. Sebastian Seung. Petascale neural circuit reconstruction: automated methods, August 2021. Pages: 2021.08.04.455162 Section: New Results.

Sven Dorkenwald, Nicholas L Turner, Thomas Macrina, Kisuk Lee, Ran Lu, Jingpeng Wu, Agnes L Bodor, Adam A Bleckert, Derrick Brittain, Nico Kemnitz, William M Silversmith, Dodam Ih, Jonathan Zung, Aleksandar Zlateski, Ignacio Tartavull, Szi-Chieh Yu, Sergiy Popovych, William Wong, Manuel Castro, Chris S Jordan, Alyssa M Wilson, Emmanouil Froudarakis, JoAnn Buchanan, Marc M Takeno, Russel Torres, Gayathri Mahalingam, Forrest Collman, Casey M Schneider-Mizell, Daniel J Bumbarger, Yang Li, Lynne Becker, Shelby Suckow, Jacob Reimer, Andreas S Tolias, Nuno Macarico da Costa, R Clay Reid, and H Sebastian Seung. Binary and analog variation of synapses between cortical pyramidal neurons. *eLife*, 11:e76120, November 2022a. ISSN 2050-084X. doi: 10.7554/eLife.76120. Publisher: eLife Sciences Publications, Ltd.

Sven Dorkenwald, Claire E. McKellar, Thomas Macrina, Nico Kemnitz, Kisuk Lee, Ran Lu, Jingpeng Wu, Sergiy Popovych, Eric Mitchell, Barak Nehoran, Zhen Jia, J. Alexander Bae, Shang Mu, Dodam Ih, Manuel Castro, Oluwaseun Ogedengbe, Akhilesh Halageri, Kai Kuehner, Amy R. Sterling, Zoe Ashwood, Jonathan Zung, Derrick Brittain, Forrest Collman, Casey Schneider-Mizell, Chris Jordan, William Silversmith, Christa Baker, David Deutsch, Lucas Encarnacion-Rivera, Sandeep Kumar, Austin Burke, Doug Bland, Jay Gager, James Hebditch, Selden Koolman, Merlin Moore, Sarah Morejohn, Ben Silverman, Kyle Willie, Ryan Willie, Szi-chieh Yu, Mala Murthy, and H. Sebastian Seung. FlyWire: online community for whole-brain connectomics. *Nature Methods*, 19(1):119–128, January 2022b. ISSN 1548-7105. doi: 10.1038/s41592-021-01330-0. Number: 1 Publisher: Nature Publishing Group.

Wei-Chung Allen Lee, Vincent Bonin, Michael Reed, Brett J. Graham, Greg Hood, Katie Glattfelder, and R. Clay Reid. Anatomy and function of an excitatory network in the visual cortex. *Nature*, 532 (7599):370–374, April 2016. ISSN 1476-4687. doi: 10.1038/nature17192. Number: 7599 Publisher: Nature Publishing Group.

Ho Ko, Sonja B. Hofer, Bruno Pichler, Katherine A. Buchanan, P. Jesper Sjöström, and Thomas D. Mrsic-Flogel. Functional specificity of local synaptic connections in neocortical networks. *Nature*, 473 (7345):87–91, May 2011. ISSN 1476-4687. doi: 10.1038/nature09880. Number: 7345 Publisher: Nature Publishing Group.

Zhuokun Ding, Paul Fahey, Stelios Papadopoulos, Eric Wang, Alexander Kunnin, Brendan Celli, Christos Papadopoulos, Andersen Chang, Jiakun Fu, Zhiwei Ding, Saumil Patel, MICrONS Consortium, Emmanouil Froudarakis, Sebastian Seung, Forrest Collman, Nuno Ma\textbackslashc{c}arico da Costa, Clay Reid, Edgar Walker, Xaq Pitkow, Jacob Reimer, and Andreas Tolias. Functional connectomics reveals general wiring rule in mouse visual cortex, March 2023.

Andrea Giovannucci, Johannes Friedrich, Pat Gunn, Jérémie Kalfon, Brandon L Brown, Sue Ann Koay, Jiannis Taxidis, Farzaneh Najafi, Jeffrey L Gauthier, Pengcheng Zhou, Baljit S Khakh, David W Tank, Dmitri B Chklovskii, and Eftychios A Pnevmatikakis. CaImAn an open source tool for scalable calcium imaging data analysis. *eLife*, 8:e38173, January 2019. ISSN 2050-084X. doi: 10.7554/eLife.38173. Publisher: eLife Sciences Publications, Ltd.

Marius Pachitariu, Carsen Stringer, Mario Dipoppa, Sylvia Schröder, L. Federico Rossi, Henry Dalgleish, Matteo Carandini, and Kenneth D. Harris. Suite2p: beyond 10,000 neurons with standard two-photon microscopy, July 2017. Pages: 061507 Section: New Results.

Marius Pachitariu, Shashwat Sridhar, and Carsen Stringer. Solving the spike sorting problem with Kilosort, January 2023. Pages: 2023.01.07.523036 Section: New Results.

Jason E. Chung, Jeremy F. Magland, Alex H. Barnett, Vanessa M. Tolosa, Angela C. Tooker, Kye Y. Lee, Kedar G. Shah, Sarah H. Felix, Loren M. Frank, and Leslie F. Greengard. A Fully Automated Approach to Spike Sorting. *Neuron*, 95(6):1381–1394.e6, September 2017. ISSN 0896-6273. doi: 10.1016/j.neuron.2017.08.030.

Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21(9):1281–1289, September 2018. ISSN 1546-1726. doi: 10.1038/s41593-018-0209-y. Number: 9 Publisher: Nature Publishing Group.

Talmo D. Pereira, Nathaniel Tabris, Arie Matsliah, David M. Turner, Junyu Li, Shruthi Ravindranath, Eleni S. Papadoyannis, Edna Normand, David S. Deutsch, Z. Yan Wang, Grace C. McKenzie-Smith, Catalin C. Mitelut, Marielisa Diez Castro, John D'Uva, Mikhail Kislin, Dan H. Sanes, Sarah D. Kocher, Samuel S.-H. Wang, Annegret L. Falkner, Joshua W. Shaevitz, and Mala Murthy. SLEAP: A deep learning system for multi-animal pose tracking. *Nature Methods*, 19(4):486–495, April 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01426-1. Number: 4 Publisher: Nature Publishing Group.

Jeffrey E. Markowitz, Winthrop F. Gillis, Celia C. Beron, Shay Q. Neufeld, Keiramarie Robertson, Neha D. Bhagat, Ralph E. Peterson, Emalee Peterson, Minsuk Hyun, Scott W. Linderman, Bernardo L. Sabatini, and Sandeep Robert Datta. The Striatum Organizes 3D Behavior via Moment-to-Moment Action Selection. *Cell*, 174(1):44–58.e17, June 2018. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2018.04.019. Publisher: Elsevier.

Ruben Dries, Qian Zhu, Rui Dong, Chee-Huat Linus Eng, Huipeng Li, Kan Liu, Yuntian Fu, Tianxiao Zhao, Arpan Sarkar, Feng Bao, Rani E. George, Nico Pierson, Long Cai, and Guo-Cheng Yuan. Giotto: a toolbox for integrative analysis and visualization of spatial expression data. *Genome Biology*, 22(1):78, March 2021. ISSN 1474-760X. doi: 10.1186/s13059-021-02286-2.

Giovanni Palla, Hannah Spitzer, Michal Klein, David Fischer, Anna Christina Schaar, Louis Benedikt Kuemmerle, Sergei Rybakov, Ignacio L. Ibarra, Olle Holmberg, Isaac Virshup, Mohammad Lotfollahi, Sabrina Richter, and Fabian J. Theis. Squidpy: a scalable framework for spatial omics analysis. *Nature Methods*, 19(2):171–178, February 2022. ISSN 1548-7105. doi: 10.1038/s41592-021-01358-2. Number: 2 Publisher: Nature Publishing Group.

Wenjing Yin, Derrick Brittain, Jay Borseth, Marie E. Scott, Derric Williams, Jedediah Perkins, Christopher S. Own, Matthew Murfitt, Russel M. Torres, Daniel Kapner, Gayathri Mahalingam, Adam Bleckert, Daniel Castelli, David Reid, Wei-Chung Allen Lee, Brett J. Graham, Marc Takeno, Daniel J. Bumbarger, Colin Farrell, R. Clay Reid, and Nuno Macarico da Costa. A petascale automated imaging pipeline for mapping neuronal circuits with high-throughput transmission electron microscopy. *Nature Communications*, 11(1):4949, October 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-18659-3. Number: 1 Publisher: Nature Publishing Group.

Juan Carlos Tapia, Narayanan Kasthuri, Kenneth J. Hayworth, Richard Schalek, Jeff W. Lichtman, Stephen J. Smith, and JoAnn Buchanan. High-contrast en bloc staining of neuronal tissue for field emission scanning electron microscopy. *Nature Protocols*, 7(2):193–206, February 2012. ISSN 1750-2799. doi: 10.1038/nprot.2011.439. Number: 2 Publisher: Nature Publishing Group.

R. Schalek, N. Kasthuri, K. Hayworth, D. Berger, J. Tapia, J. Morgan, S. Turaga, E. Fagerholm, H. Seung, and J. Lichtman. Development of High-Throughput, High-Resolution 3D Reconstruction of Large-Volume Biological Tissue Using Automated Tape Collection Ultramicrotomy and Scanning Electron Microscopy. *Microscopy and Microanalysis*, 17(S2):966–967, July 2011. ISSN 1435-8115, 1431-9276. doi: 10.1017/S1431927611005708. Publisher: Cambridge University Press.

Christopher S Own, Matthew F Murfitt, Lawrence S Own, Derrick Brittain, Nuno da Costa, R Clay Reid, David G C Hildebrand, Brett Graham, and Wei-Chung Allen Lee. Reel-to-Reel Electron Microscopy: Latency-Free Continuous Imaging of Large Sample Volumes. *Microscopy and Microanalysis*, 21(S3):157–158, August 2015. ISSN 1431-9276. doi: 10.1017/S1431927615001580.

Kenneth J. Hayworth, Josh L. Morgan, Richard Schalek, Daniel R. Berger, David G. C. Hildebrand, and Jeff W. Lichtman. Imaging ATUM ultrathin section libraries with WaferMapper: a multi-scale approach to EM reconstruction of neural circuits. *Frontiers in Neural Circuits*, 8, 2014. ISSN 1662-5110.

Andreas Fabri and Laurent Rineau. CGAL and the Qt Graphics View Framework. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023.

Fernando Cacciola, Mael Rouxel-Labbé, Baskın Şenbaşlar, and Julian Komaromy. Triangulated Surface Mesh Simplification. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023.

Xiang Gao, Sébastien Loriot, and Andrea Tagliasacchi. Triangulated Surface Mesh Skeletonization. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023.

Sven Dorkenwald. sdorkenw/MeshParty, March 2022. original-date: 2018-09-11T23:38:46Z.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

Luis Pastor, Sofia Bayona, Juan P. Brito, María Cuevas, Isabel Fernaud, Sergio Emilio Galindo, Juan José García-Cantero, Francisco González de Quevedo, Susana Mata, Oscar David Robles, Angel Rodríguez, Pablo Toharia, and Ana Zdravkovic. A Unified Framework for Neuroscience Morphological Data Visualization. *Applied Sciences*, 11(10):4652, January 2021. ISSN 2076-3417. doi: 10.3390/app11104652. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

Casey M Schneider-Mizell, Stephan Gerhard, Mark Longair, Tom Kazimiers, Feng Li, Maarten F Zwart, Andrew Champion, Frank M Midgley, Richard D Fetter, Stephan Saalfeld, and Albert Cardona. Quantitative neuroanatomy for connectomics in Drosophila. *eLife*, 5:e12059, March 2016. ISSN 2050-084X. doi: 10.7554/eLife.12059. Publisher: eLife Sciences Publications, Ltd.

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. Publisher: Nature Publishing Group.

Daniel Xenes, Lindsey M. Kitchell, Patricia K. Rivlin, Rachel Brodsky, Hannah Gooden, Justin Joyce, Diego Luna, Raphael Norman-Tenazas, Devin Ramsden, Kevin Romero, Victoria Rose, Marisel Villafane-Delgado, William Gray-Roncal, and Brock Wester. NeuVue: A Framework and Workflows for High-Throughput Electron Microscopy Connectomics Proofreading, July 2022. Pages: 2022.07.18.500521 Section: New Results.

Casey M. Schneider-Mizell, Agnes Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Leila Elabbady, Daniel Kapner, Sam Kinn, Gayathri Mahalingam, Sharmishtaa Seshamani, Shelby Suckow, Marc Takeno, Russel Torres, Wenjing Yin, Sven Dorkenwald, J. Alexander Bae, Manuel A. Castro, Paul G. Fahey, Emmanouil Froudakis, Akhilesh Halageri, Zhen Jia, Chris Jordan, Nico Kemnitz, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, Stelios Papadopoulos, Saumil Patel, Xaq Pitkow, Sergiy Popovych, William Silversmith, Fabian H. Sinz, Nicholas L. Turner, William Wong, Jingpeng Wu, Szi-chieh Yu, MICrONS Consortium, Jacob Reimer, Andreas S. Tolias, H. Sebastian Seung, R. Clay Reid, Forrest Collman, and Nuno Maçarico da Costa. Cell-type-specific inhibitory circuitry from a connectomic census of mouse visual cortex, February 2023. Pages: 2023.01.23.525290 Section: New Results.

Casey M Schneider-Mizell, Agnes L Bodor, Forrest Collman, Derrick Brittain, Adam Bleckert, Sven Dorkenwald, Nicholas L Turner, Thomas Macrina, Kisuk Lee, Ran Lu, Jingpeng Wu, Jun Zhuang, Anirban Nandi, Brian Hu, JoAnn Buchanan, Marc M Takeno, Russel Torres, Gayathri Mahalingam, Daniel J Bumbarger, Yang Li, Thomas Chartrand, Nico Kemnitz, William M Silversmith, Dodam Ih, Jonathan Zung, Aleksandar Zlateski, Ignacio Tartavull, Sergiy Popovych, William Wong, Manuel Castro, Chris S Jordan, Emmanouil Froudarakis, Lynne Becker, Shelby Suckow, Jacob Reimer, Andreas S Tolias, Costas A Anastassiou, H Sebastian Seung, R Clay Reid, and Nuno Maçarico da Costa. Structure and function of axo-axonic inhibition. *eLife*, 10:e73783, December 2021. ISSN 2050-084X. doi: 10.7554/eLife.73783. Publisher: eLife Sciences Publications, Ltd.

Marissa A. Weis, Stelios Papadopoulos, Laura Hansel, Timo Lüddecke, Brendan Celii, Paul G. Fahey, J. Alexander Bae, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Erick Cobos, Forrest Collman, Nuno Maçarico da Costa, Sven Dorkenwald, Leila Elabbady, Emmanouil Froudarakis, Akhilesh Halageri, Zhen Jia, Chris Jordan, Dan Kapner, Nico Kemnitz, Sam Kinn, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, Saumil Patel, Xaq Pitkow, Sergiy Popovych, R. Clay Reid, Casey M. Schneider-Mizell, H. Sebastian Seung, William Silversmith, Fabian H. Sinz, Marc Takeno, Russel Torres, Nicholas L. Turner, William Wong, Jingpeng Wu, Wenjing Yin, Szi-chieh Yu, Jacob Reimer, Andreas S. Tolias, and Alexander S. Ecker. Large-scale unsupervised discovery of excitatory morphological cell types in mouse visual cortex, December 2022. Pages: 2022.12.22.521541 Section: New Results.

Sven Dorkenwald, Peter H. Li, Michał Januszewski, Daniel R. Berger, Jeremy Maitin-Shepard, Agnes L. Bodor, Forrest Collman, Casey M. Schneider-Mizell, Nuno Maçarico da Costa, Jeff W. Lichtman, and Viren Jain. Multi-Layered Maps of Neuropil with Segmentation-Guided Contrastive Learning, March 2022c. Pages: 2022.03.29.486320 Section: New Results.

A. Peters and M. L. Feldman. The projection of the lateral geniculate nucleus to area 17 of the rat cerebral cortex. I. General description. *Journal of Neurocytology*, 5(1):63–84, February 1976. ISSN 0300-4864. doi: 10.1007/BF01176183.

K A Martin and D Whitteridge. Form, function and intracortical projections of spiny neurones in the striate visual cortex of the cat. *The Journal of Physiology*, 353:463–504, August 1984. ISSN 0022-3751.

Federico Scala, Dmitry Kobak, Matteo Bernabucci, Yves Bernaerts, Cathryn René Cadwell, Jesus Ramon Castro, Leonard Hartmanis, Xiaolong Jiang, Sophie Laturnus, Elanine Miranda, Shalaka Mulherkar, Zheng Huan Tan, Zizhen Yao, Hongkui Zeng, Rickard Sandberg, Philipp Berens, and Andreas S. Tolias. Phenotypic variation of transcriptomic cell types in mouse motor cortex. *Nature*, 598(7879):144–150, October 2021. ISSN 1476-4687. doi: 10.1038/s41586-020-2907-3. Number: 7879 Publisher: Nature Publishing Group.

Hanchuan Peng, Peng Xie, Lijuan Liu, Xiuli Kuang, Yimin Wang, Lei Qu, Hui Gong, Shengdian Jiang, Anan Li, Zongcai Ruan, Liya Ding, Zizhen Yao, Chao Chen, Mengya Chen, Tanya L. Daigle, Rachel Dalley, Zhangcan Ding, Yanjun Duan, Aaron Feiner, Ping He, Chris Hill, Karla E. Hirokawa, Guodong Hong, Lei Huang, Sara Kebede, Hsien-Chi Kuo, Rachael Larsen, Phil Lesnar, Longfei Li, Qi Li, Xiangning Li, Yaoyao Li, Yuanyuan Li, An Liu, Donghuan Lu, Stephanie Mok, Lydia Ng, Thuc Nghi Nguyen, Qiang Ouyang, Jintao Pan, Elise Shen, Yuanyuan Song, Susan M. Sunkin, Bosiljka Tasic, Matthew B. Veldman, Wayne Wakeman, Wan Wan, Peng Wang, Quanxin Wang, Tao Wang, Yaping Wang, Feng Xiong, Wei Xiong, Wenjie Xu, Min Ye, Lulu Yin, Yang Yu, Jia Yuan, Jing Yuan, Zhixi Yun, Shaoqun Zeng, Shichen Zhang, Sujun Zhao, Zijun Zhao, Zhi Zhou, Z. Josh Huang, Luke Esposito, Michael J. Hawrylycz, Staci A. Sorensen, X. William Yang, Yefeng Zheng, Zhongze Gu, Wei Xie, Christof Koch, Qingming Luo, Julie A. Harris, Yun Wang, and Hongkui Zeng. Morphological diversity of single neurons in molecularly defined cell types. *Nature*, 598(7879):174–181, October 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03941-1.

Nathan W. Gouwens, Staci A. Sorensen, Fahimeh Baftizadeh, Agata Budzillo, Brian R. Lee, Tim Jarsky, Lauren Alfiler, Katherine Baker, Eliza Barkan, Kyla Berry, Darren Bertagnolli, Kris Bickley, Jasmine Bomben, Thomas Braun, Krissy Brouner, Tamara Casper, Kirsten Crichton, Tanya L. Daigle, Rachel Dalley, Rebecca A. de Frates, Nick Dee, Tsega Desta, Samuel Dingman Lee, Nadezhda Dotson, Tom Egdorf, Lauren Ellingwood, Rachel Enstrom, Luke Esposito, Colin Farrell, David Feng, Olivia Fong, Rohan Gala, Clare Gamlin, Amanda Gary, Alexandra Glandon, Jeff Goldy, Melissa Gorham, Lucas Graybuck, Hong Gu, Kristen Hadley, Michael J. Hawrylycz, Alex M. Henry, DiJon Hill, Madie Hupp, Sara Kebede, Tae Kyung Kim, Lisa Kim, Matthew Kroll, Changkyu Lee, Katherine E. Link, Matthew Mallory, Rusty Mann, Michelle Maxwell, Medea McGraw, Delissa McMillen, Alice Mukora, Lindsay Ng, Lydia Ng, Kiet Ngo, Philip R. Nicovich, Aaron Oldre, Daniel Park, Hanchuan Peng, Osnat Penn, Thanh Pham, Alice Pom, Zoran Popović, Lydia Potekhina, Ramkumar Rajanbabu, Shea Ransford, David Reid, Christine Rimorin, Miranda Robertson, Kara Ronellenfitch, Augustin Ruiz, David Sandman, Kimberly Smith, Josef Sulc, Susan M. Sunkin, Aaron Szafer, Michael Tieu, Amy Torkelson, Jessica Trinh, Herman Tung, Wayne Wakeman, Katelyn Ward, Grace Williams, Zhi Zhou, Jonathan T. Ting, Anton Arkhipov, Uygar Sümbül, Ed S. Lein, Christof Koch, Zizhen Yao, Bosiljka Tasic, Jim Berg, Gabe J. Murphy, and Hongkui Zeng. Integrated Morphoelectric and Transcriptomic Classification of Cortical GABAergic Cells. *Cell*, 183(4):935–953.e19, November 2020. ISSN 1097-4172. doi: 10.1016/j.cell.2020.09.057.

C. R. Gamlin, C. M. Schneider-Mizell, M. Mallory, L. Elabbady, N. Gouwens, G. Williams, A. Mukora, R. Dalley, A. Bodor, D. Brittain, J. Buchanan, D. Bumbarger, D. Kapner, S. Kinn, G. Mahalingam, S. Seshamani, M. Takeno, R. Torres, W. Yin, P. R. Nicovich, J. A. Bae, M. A. Castro, S. Dorkenwald, A. Halageri, Z. Jia, C. Jordan, N. Kemnitz, K. Lee, K. Li, R. Lu, T. Macrina, E. Mitchell, S. S. Mondal, S. Mu, B. Nehoran, S. Popovych, W. Silversmith, N. L. Turner, W. Wong, J. Wu, S. Yu, J. Berg, T. Jarsky, B. Lee, H. S. Seung, H. Zeng, R. C. Reid, F. Collman, N. M. da Costa, and S. A. Sorensen. Integrating EM and Patch-seq data: Synaptic connectivity and target specificity of predicted Sst transcriptomic types, March 2023. Pages: 2023.03.22.533857 Section: New Results.

Leila Elabbady, Sharmishtaa Seshamani, Shang Mu, Gayathri Mahalingam, Casey Schneider-Mizell, Agnes Bodor, J. Alexander Bae, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Erick Cobos, Sven Dorkenwald, Paul G. Fahey, Emmanouil Froudarakis, Akhilesh Halageri, Zhen Jia, Chris Jordan, Dan Kapner, Nico Kemnitz, Sam Kinn, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Eric Mitchell, Shanka Subhra Mondal, Barak Nehoran, Stelios Papadopoulos, Saumil Patel, Xaq Pitkow, Sergiy Popovych, Jacob Reimer, William Silversmith, Fabian H. Sinz, Marc Takeno, Russel Torres, Nicholas Turner, William Wong, Jingpeng Wu, Wenjing Yin, Szi-chieh Yu, Andreas Tolias, H. Sebastian Seung, R. Clay Reid, Nuno Maçarico Da Costa, and Forrest Collman. Quantitative Census of Local Somatic Features in Mouse Visual Cortex, July 2022. Pages: 2022.07.20.499976 Section: New Results.

Khaled Al-Thelaya, Marco Agus, Nauman Ullah Gilal, Yin Yang, Giovanni Pintore, Enrico Gobbetti, Corrado Calí, Pierre J. Magistretti, William Mifsud, and Jens Schneider. InShaDe: Invariant Shape Descriptors for visual 2D and 3D cellular and nuclear shape analysis and classification. *Computers & Graphics*, 98:105–125, August 2021. ISSN 0097-8493. doi: 10.1016/j.cag.2021.04.037.

Valentyna Zinchenko, Johannes Hugger, Virginie Uhlmann, Detlev Arendt, and Anna Kreshuk. MorphoFeatures for unsupervised exploration of cell types, tissues, and organs in volume electron microscopy. *eLife*, 12:e80918, February 2023. ISSN 2050-084X. doi: 10.7554/eLife.80918.

Sharmishtaa Seshamani, Leila Elabbady, Casey Schneider-Mizell, Gayathri Mahalingam, Sven Dorkenwald, Agnes Bodor, Thomas Macrina, Daniel Bumbarger, JoAnn Buchanan, Marc Takeno, Wenjing Yin, Derrick Brittain, Russel Torres, Daniel Kapner, Kisuk lee, Ran Lu, Jinpeng Wu, Nuno daCosta, Clay Reid, and Forrest Collman. Automated Neuron Shape Analysis from Electron Microscopy, May 2020. arXiv:2006.00100 [cs].

R. Azouz, C. M. Gray, L. G. Nowak, and D. A. McCormick. Physiological properties of inhibitory interneurons in cat striate cortex. *Cerebral Cortex (New York, N.Y.: 1991)*, 7(6):534–545, September 1997. ISSN 1047-3211. doi: 10.1093/cercor/7.6.534.

Xuerong Xiao, Maja Djurisic, Assaf Hoogi, Richard W. Sapp, Carla J. Shatz, and Daniel L. Rubin. Automated dendritic spine detection using convolutional neural networks on maximum intensity projected microscopic volumes. *Journal of neuroscience methods*, 309:25–34, November 2018. ISSN 0165-0270. doi: 10.1016/j.jneumeth.2018.08.019.

Meghan K. Driscoll, Erik S. Welf, Andrew R. Jamieson, Kevin M. Dean, Tadamoto Isogai, Reto Fiolka, and Gaudenz Danuser. Robust and automated detection of subcellular morphological motifs in 3D microscopy images. *Nature Methods*, 16(10):1037–1044, October 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0539-z. Number: 10 Publisher: Nature Publishing Group.

Firdaus Janoos, Kishore Mosaliganti, Xiaoyin Xu, Raghu Machiraju, Kun Huang, and Stephen T. C. Wong. Robust 3D reconstruction and identification of dendritic spines from optical microscopy imaging. *Medical Image Analysis*, 13(1):167–179, February 2009. ISSN 1361-8423. doi: 10.1016/j.media.2008.06.019.

Peng Shi, Yue Huang, and Jinsheng Hong. Automated three-dimensional reconstruction and morphological analysis of dendritic spines based on semi-supervised learning. *Biomedical Optics Express*, 5(5):1541–1553, April 2014. ISSN 2156-7085. doi: 10.1364/BOE.5.001541.

Subhadip Basu, Punam Kumar Saha, Matylda Roszkowska, Marta Magnowska, Ewa Baczynska, Nirmal Das, Dariusz Plewczynski, and Jakub Wlodarczyk. Quantitative 3-D morphometric analysis of individual dendritic spines. *Scientific Reports*, 8(1):3545, February 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-21753-8. Number: 1 Publisher: Nature Publishing Group.

Ruth Benavides-Piccione, Isabel Fernaud-Espinosa, Victor Robles, Rafael Yuste, and Javier DeFelipe. Age-Based Comparison of Human Dendritic Spine Structure Using Complete Three-Dimensional Reconstructions. *Cerebral Cortex*, 23(8):1798–1810, August 2013. ISSN 1047-3211. doi: 10.1093/cercor/bhs154.

K. M. Harris and J. K. Stevens. Dendritic spines of CA 1 pyramidal cells in the rat hippocampus: serial electron microscopy with reference to their biophysical characteristics. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, 9(8):2982–2997, August 1989. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.09-08-02982.1989.

Jon I. Arellano, Ruth Benavides-Piccione, Javier DeFelipe, and Rafael Yuste. Ultrastructure of Dendritic Spines: Correlation Between Synaptic and Spine Morphologies. *Frontiers in Neuroscience*, 1(1): 131–143, October 2007. ISSN 1662-4548. doi: 10.3389/neuro.01.1.1.010.2007.

Xiaojun Wang, Jason Tucciarone, Siqi Jiang, Fangfang Yin, Bor-Shuen Wang, Dingkang Wang, Yao Jia, Xueyan Jia, Yuxin Li, Tao Yang, Zhengchao Xu, Masood A. Akram, Yusu Wang, Shaoqun Zeng, Giorgio A. Ascoli, Partha Mitra, Hui Gong, Qingming Luo, and Z. Josh Huang. Genetic Single Neuron Anatomy Reveals Fine Granularity of Cortical Axo-Axonic Cells. *Cell Reports*, 26(11):

3145–3159.e5, March 2019. ISSN 2211-1247. doi: 10.1016/j.celrep.2019.02.040.

Melis Inan, Lidia Blázquez-Llorca, Angel Merchán-Pérez, Stewart A. Anderson, Javier DeFelipe, and Rafael Yuste. Dense and Overlapping Innervation of Pyramidal Neurons by Chandelier Cells. *The Journal of Neuroscience*, 33(5):1907–1914, January 2013. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.4049-12.2013.

Gregg A. Wildenberg, Matt R. Rosen, Jack Lundell, Dawn Paukner, David J. Freedman, and Narayanan Kasthuri. Primate neuronal connections are sparse in cortex as compared to mouse. *Cell Reports*, 36(11):109709, September 2021. ISSN 2211-1247. doi: 10.1016/j.celrep.2021.109709.

Charles E. Ribak, James E. Vaughn, and Roeert P. Barber. Immunocytochemical localization of GABAergic neurones at the electron microscopical level. *The Histochemical Journal*, 13(4):555–582, July 1981. ISSN 1573-6865. doi: 10.1007/BF01002711.

Javier DeFelipe and Isabel Fariñas. The pyramidal neuron of the cerebral cortex: Morphological and chemical characteristics of the synaptic inputs. *Progress in Neurobiology*, 39(6):563–607, December 1992. ISSN 0301-0082. doi: 10.1016/0301-0082(92)90015-7.

Taekyung Kwon, Angel Merchán-Pérez, Emiliano M Rial Verde, José-Rodrigo Rodríguez, Javier DeFelipe, and Rafael Yuste. Ultrastructural, Molecular and Functional Mapping of GABAergic Synapses on Dendritic Spines and Shafts of Neocortical Pyramidal Neurons. *Cerebral Cortex*, 29(7):2771–2781, July 2019. ISSN 1047-3211. doi: 10.1093/cercor/bhy143.

Daniel Udvary, Philipp Harth, Jakob H. Macke, Hans-Christian Hege, Christiaan P. J. de Kock, Bert Sakmann, and Marcel Oberlaender. The impact of neuron morphology on cortical network architecture. *Cell Reports*, 39(2):110677, April 2022. ISSN 2211-1247. doi: 10.1016/j.celrep.2022.110677.

Yuriy Mishchenko, Tao Hu, Josef Spacek, John Mendenhall, Kristen M. Harris, and Dmitri B. Chklovskii. Ultrastructural analysis of hippocampal neuropil from the connectomics perspective. *Neuron*, 67 (6):1009–1020, September 2010. ISSN 0896-6273. doi: 10.1016/j.neuron.2010.08.014.

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. doi: 10.1016/j.cell.2015.06.054.

Solange P. Brown and Shaul Hestrin. Intracortical circuits of pyramidal neurons reflect their long-range axonal targets. *Nature*, 457(7233):1133–1136, February 2009. ISSN 1476-4687. doi: 10.1038/nature07658. Number: 7233 Publisher: Nature Publishing Group.

Nuno Maçarico da Costa and Kevan A. C. Martin. How Thalamus Connects to Spiny Stellate Cells in the Cat's Visual Cortex. *Journal of Neuroscience*, 31(8):2925–2937, February 2011. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.5961-10.2011. Publisher: Society for Neuroscience Section: Articles.

Xiaolong Jiang, Shan Shen, Cathryn R. Cadwell, Philipp Berens, Fabian Sinz, Alexander S. Ecker, Saumil Patel, and Andreas S. Tolias. Principles of connectivity among morphologically defined cell types in adult neocortex. *Science (New York, N.Y.)*, 350(6264):aac9462, November 2015. ISSN 1095-9203. doi: 10.1126/science.aac9462.

Soohyun Lee, Illya Kruglikov, Z. Josh Huang, Gord Fishell, and Bernardo Rudy. A disinhibitory circuit mediates motor integration in the somatosensory cortex. *Nature Neuroscience*, 16(11):1662–1670, November 2013. ISSN 1546-1726. doi: 10.1038/nn.3544. Number: 11 Publisher: Nature Publishing Group.

Tamás F. Freund and István Katona. Perisomatic Inhibition. *Neuron*, 56(1):33–42, October 2007. ISSN 0896-6273. doi: 10.1016/j.neuron.2007.09.012.

Carsten K. Pfeffer, Mingshan Xue, Miao He, Z. Josh Huang, and Massimo Scanziani. Inhibition of inhibition in visual cortex: the logic of connections between molecularly distinct interneurons. *Nature Neuroscience*, 16(8):1068–1076, August 2013. ISSN 1546-1726. doi: 10.1038/nn.3446. Number: 8 Publisher: Nature Publishing Group.

Yang-Sun Hwang, Catherine Maclachlan, Jérôme Blanc, Anaëlle Dubois, Carl C. H. Petersen, Graham Knott, and Seung-Hee Lee. 3D Ultrastructure of Synaptic Inputs to Distinct GABAergic Neurons in the Mouse Primary Visual Cortex. *Cerebral Cortex (New York, N.Y.: 1991)*, 31(5):2610–2624, March 2021. ISSN 1460-2199. doi: 10.1093/cercor/bhaa378.

M Megıas, Zs Emri, T. F Freund, and A. I Gulyás. Total number and distribution of inhibitory and excitatory synapses on hippocampal CA1 pyramidal cells. *Neuroscience*, 102(3):527–540, February 2001. ISSN 0306-4522. doi: 10.1016/S0306-4522(00)00496-6.

J. G. Parnavelas, K. Sullivan, A. R. Lieberman, and K. E. Webster. Neurons and their synaptic organization in the visual cortex of the rat. *Cell and Tissue Research*, 183(4):499–517, October 1977. ISSN 1432-0878. doi: 10.1007/BF00225663.

Luke Campagnola, Stephanie C. Seeman, Thomas Chartrand, Lisa Kim, Alex Hoggarth, Clare Gamlin, Shinya Ito, Jessica Trinh, Pasha Davoudian, Cristina Radaelli, Mean-Hwan Kim, Travis Hage, Thomas Braun, Lauren Alfiler, Julia Andrade, Phillip Bohn, Rachel Dalley, Alex Henry, Sara Kebede, Alice Mukora, David Sandman, Grace Williams, Rachael Larsen, Corinne Teeter, Tanya L. Daigle, Kyla Berry, Nadia Dotson, Rachel Enstrom, Melissa Gorham, Madie Hupp, Samuel Dingman Lee, Kiet Ngo, Philip R. Nicovich, Lydia Potekhina, Shea Ransford, Amanda Gary, Jeff Goldy, Delissa McMillen, Trangthanh Pham, Michael Tieu, La'Akea Siverts, Miranda Walker, Colin Farrell, Martin Schroedter, Cliff Slaughterbeck, Charles Cobb, Richard Ellenbogen, Ryder P. Gwinn, C. Dirk Keene, Andrew L. Ko, Jeffrey G. Ojemann, Daniel L. Silbergeld, Daniel Carey, Tamara Casper, Kirsten Crichton, Michael Clark, Nick Dee, Lauren Ellingwood, Jessica Gloe, Matthew Kroll, Josef Sulc, Herman Tung, Katherine Wadhwani, Krissy Brouner, Tom Egdorf, Michelle Maxwell, Medea McGraw, Christina Alice Pom, Augustin Ruiz, Jasmine Bomben, David Feng, Nika Hejazinia, Shu Shi, Aaron Szafer, Wayne Wakeman, John Phillips, Amy Bernard, Luke Esposito, Florence D. D'Orazi, Susan Sunkin, Kimberly Smith, Bosiljka Tasic, Anton Arkhipov, Staci Sorensen, Ed Lein, Christof Koch, Gabe Murphy, Hongkui Zeng, and Tim Jarsky. Local connectivity and synaptic dynamics in mouse and human neocortex. *Science*, 375(6585):eabj5861, March 2022. doi: 10.1126/science.abj5861. Publisher: American Association for the Advancement of Science.

Davi D. Bock, Wei-Chung Allen Lee, Aaron M. Kerlin, Mark L. Andermann, Greg Hood, Arthur W. Wetzel, Sergey Yurgenson, Edward R. Soucy, Hyon Suk Kim, and R. Clay Reid. Network anatomy and in vivo physiology of visual cortical neurons. *Nature*, 471(7337):177–182, March 2011. ISSN 1476-4687. doi: 10.1038/nature09802. Number: 7337 Publisher: Nature Publishing Group.

Rita Bopp, Nuno Maçarico da Costa, Björn M. Kampa, Kevan A. C. Martin, and Morgane M. Roth. Pyramidal Cells Make Specific Connections onto Smooth (GABAergic) Neurons in Mouse Visual Cortex. *PLOS Biology*, 12(8):e1001932, August 2014. ISSN 1545-7885. doi: 10.1371/journal.pbio.1001932. Publisher: Public Library of Science.

Helene Schmidt, Anjali Gour, Jakob Straehle, Kevin M. Boergens, Michael Brecht, and Moritz Helmstaedter. Axonal synapse sorting in medial entorhinal cortex. *Nature*, 549(7673):469–475, September 2017. ISSN 1476-4687. doi: 10.1038/nature24005. Number: 7673 Publisher: Nature Publishing Group.

Sen Song, Per Jesper Sjöström, Markus Reigl, Sacha Nelson, and Dmitri B. Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS biology*, 3(3):e68, March 2005. ISSN 1545-7885. doi: 10.1371/journal.pbio.0030068.

Rodrigo Perin, Thomas K. Berger, and Henry Markram. A synaptic organizing principle for cortical neuronal groups. *Proceedings of the National Academy of Sciences*, 108(13):5419–5424, March 2011. doi: 10.1073/pnas.1016051108. Publisher: Proceedings of the National Academy of Sciences.

R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, October 2002. doi: 10.1126/science.298.5594.824. Publisher: American Association for the Advancement of Science.

Yasuo Kawaguchi, Fuyuki Karube, and Yoshiyuki Kubota. Dendritic branch typing and spine expression patterns in cortical nonpyramidal cells. *Cerebral Cortex (New York, N.Y.: 1991)*, 16(5):696–711, May 2006. ISSN 1047-3211. doi: 10.1093/cercor/bhj015.

M. E. Chicurel and K. M. Harris. Three-dimensional analysis of the structure and composition of CA3 branched dendritic spines and their synaptic relationships with mossy fiber boutons in the rat hippocampus. *The Journal of Comparative Neurology*, 325(2):169–182, November 1992. ISSN 0021-9967. doi: 10.1002/cne.903250204.

Eric Wang and Andreas Tolias. Foundation model of the cortex, March 2023.

Jonathan Zung, Ignacio Tartavull, Kisuk Lee, and H. Sebastian Seung. An Error Detection and Correction Framework for Connectomics. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Felix Gonda, Xueying Wang, Johanna Beyer, Markus Hadwiger, Jeff W. Lichtman, and Hanspeter Pfister. VICE: Visual Identification and Correction of Neural Circuit Errors, May 2021. arXiv:2105.06861 [cs].

David Rolnick, Yaron Meirovitch, Toufiq Parag, Hanspeter Pfister, Viren Jain, Jeff W. Lichtman, Edward S. Boyden, and Nir Shavit. Morphological Error Detection in 3D Segmentations, May 2017. arXiv:1705.10882 [cs, q-bio, stat].

Khoa Tuan Nguyen, Ganghee Jang, Tran Anh Tuan, and Won-ki Jeong. RLCorrector: Reinforced Proofreading for Cell-level Microscopy Image Segmentation, March 2022. arXiv:2106.05487 [cs].

Philipp J. Schubert, Sven Dorkenwald, Michał Januszewski, Viren Jain, and Joergen Kornfeld. Learning cellular morphology with neural networks. *Nature Communications*, 10(1):2736, June 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-10836-3. Number: 1 Publisher: Nature Publishing Group.

Martin Schmidt, Alessandro Motta, Meike Sievers, and Moritz Helmstaedter. RoboEM: automated 3D flight tracing for synaptic-resolution connectomics, September 2022. Pages: 2022.09.08.507122 Section: New Results.

Jules Berman, Dmitri B. Chklovskii, and Jingpeng Wu. Bridging the Gap: Point Clouds for Merging Neurons in Connectomics. In Ender Konukoglu, Bjoern Menze, Archana Venkataraman, Christian Baumgartner, Qi Dou, and Shadi Albarqouni, editors, *Proceedings of The 5th International Conference on Medical Imaging with Deep Learning*, volume 172 of *Proceedings of Machine Learning Research*, pages 150–159. PMLR, July 2022.

Yaron Meirovitch, Alexander Matveev, Hayk Saribekyan, David Budden, David Rolnick, Gergely Odor, Seymour Knowles-Barley, Thouis Raymond Jones, Hanspeter Pfister, Jeff William Lichtman, and Nir Shavit. A Multi-Pass Approach to Large-Scale Connectomics, December 2016. arXiv:1612.02120 [cs, q-bio].

Ronell B. Sicat, Markus Hadwiger, and Niloy J. Mitra. Graph Abstraction for Simplified Proofreading of Slice-based Volume Segmentation. 2013. ISSN 1017-4656. doi: 10.2312/conf/EG2013/short/077-080. Accepted: 2014-01-26T14:58:25Z Publisher: The Eurographics Association.

Brian Matejek, Daniel Haehn, Haidong Zhu, Donglai Wei, Toufiq Parag, and Hanspeter Pfister. Biologically-Constrained Graphs for Global Connectomics Reconstruction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2084–2093, June 2019. doi: 10.1109/CVPR.2019.00219. ISSN: 2575-7075.

Petra Wahle, Eric Sobierajski, Ina Gasterstädt, Nadja Lehmann, Susanna Weber, Joachim HR Lübke, Maren Engelhardt, Claudia Distler, and Gundela Meyer. Neocortical pyramidal neurons with axons emerging from dendrites are frequent in non-primates, but rare in monkey and human. *eLife*, 11:e76101, April 2022. ISSN 2050-084X. doi: 10.7554/eLife.76101. Publisher: eLife Sciences Publications, Ltd.

Katherine L. Villa and Elly Nedivi. Excitatory and Inhibitory Synaptic Placement and Functional Implications. In Kazuo Emoto, Rachel Wong, Eric Huang, and Casper Hoogenraad, editors, *Dendrites*,

pages 467–487. Springer Japan, Tokyo, 2016. ISBN 978-4-431-56048-7 978-4-431-56050-0. doi: 10.1007/978-4-431-56050-0_18.

Dimitri Yatsenko, Jacob Reimer, Alexander S. Ecker, Edgar Y. Walker, Fabian Sinz, Philipp Berens, Andreas Hoenselaar, Ronald James Cotton, Athanassios S. Siapas, and Andreas S. Tolias. DataJoint: managing big scientific data using MATLAB or Python. *bioRxiv*, page 031658, November 2015. doi: 10.1101/031658.

Dimitri Yatsenko, Edgar Y. Walker, and Andreas S. Tolias. DataJoint: A Simpler Relational Data Model. *arXiv:1807.11104 [cs]*, July 2018. arXiv: 1807.11104.

Paolo Cignoni, Massimiliano Corsini, and Guido Ranzuglia. MeshLab: an Open-Source 3D Mesh Processing System. *ERCIM News*, 2008(73), 2008.

Ilker O. Yaz and Sébastien Loriot. Triangulated Surface Mesh Segmentation. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023.

Eric Perlman. Visualizing and Interacting with Large Imaging Data. *Microscopy and Microanalysis*, 25(S2):1374–1375, August 2019. ISSN 1431-9276, 1435-8115. doi: 10.1017/S1431927619007608. Publisher: Cambridge University Press.

Matthias Fey and Jan Eric Lenssen. Fast Graph Representation Learning with PyTorch Geometric, May 2019. original-date: 2017-10-06T16:03:03Z.

Supplementary Figures

a MICrONS Mesh Processing Validation

category	precision	recall	f1	# datapoints
soma	1.000	1.000	1.000	195
axon	0.995	0.968	0.981	192
glia	1.000	1.000	1.000	188
spine	0.896	0.870	0.883	22226
spine (> 0.7 um)	0.942	0.978	0.960	18030

b H01 Mesh Processing Validation

category	precision	recall	f1	# datapoints
soma	0.990	0.990	0.990	102
glia	1.000	1.000	1.000	100
axon	0.942	0.875	0.907	76
spine	0.831	0.765	0.796	7945
spine (> 0.7 um)	0.892	0.953	0.922	5851

Fig. 8. Mesh Processing Pipeline Validation a (MICrONS) Validation scores of automatic submesh (compartment) identification in comparison to human labels. This data set was produced by randomly presenting processed mesh segments to human annotators who evaluated automatic labels as true positive, true negative, false positive, or false negative (TP,TN,FP,or FN) for each structure. Only glia merges larger than the volume of a 5 μm radius sphere were considered glia merges in this processing step. The "spine (> 0.7 μm)" row reports the agreement between the automatic spine detection and human spine labeling for spines with a skeletal length greater than 0.7 μm. Below this threshold there was disagreement even among human proofreaders about whether small protrusions should be classified as a spine or not. Precision = true positives / (true positives + false positives), Recall = true positives / (true positives + false negatives), F1 = (precision * recall)/(precision + recall). The F1 statistic serves as a weighting between the recall and precision to ensure a degenerative solution is not achieved (Ex: marking all samples as positives in order to maximize recall at the detriment of precision). **b** Identical validation scores for H01 dataset.

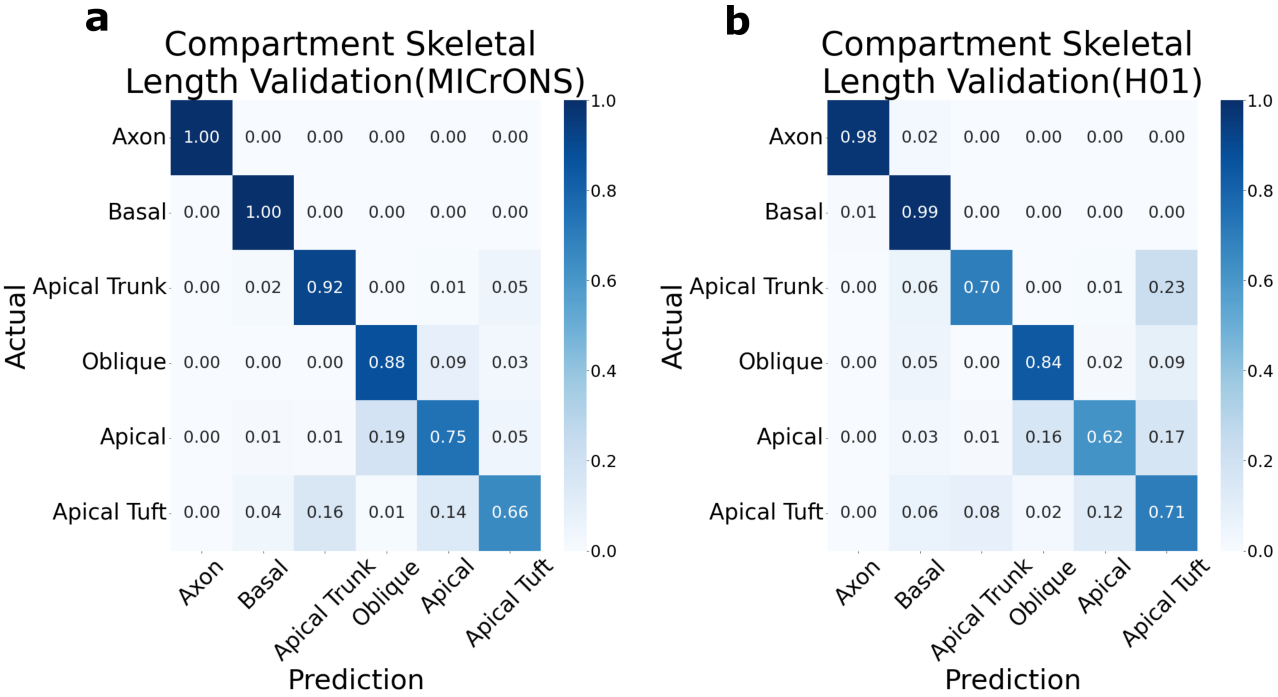
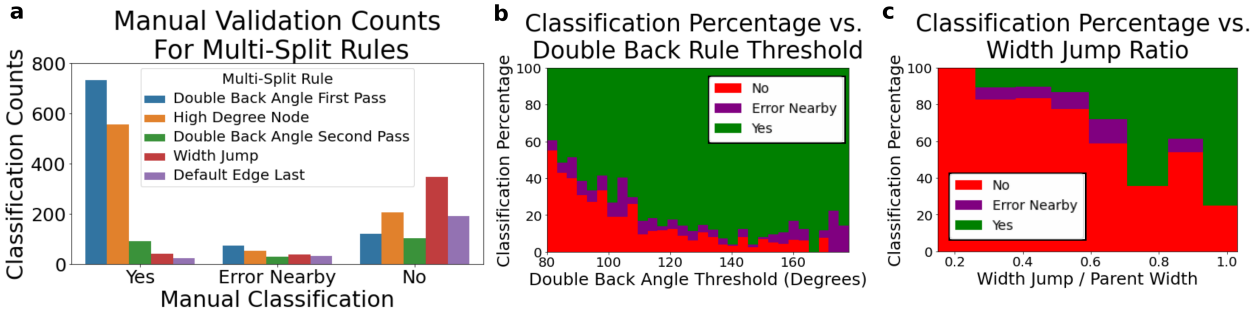


Fig. 9. Automatic Submesh (Compartment) Labeling Validation a (MICrONS) Confusion matrix comparing automatic submesh labeling to human labels (random sampling of 158 processed cells). Here the TP,FP,TN,FN metrics are computed using skeletal length agreement for each compartment. Therefore, cells with longer stems or cells with more stems of a certain compartment type more heavily influenced the scores due to the skeletal length weighting. **b** (H01) Confusion Matrix comparing automatic submesh labeling to human labeled submeshes (random sampling of 89 processed cells). Across both datasets, compartment labeling was nearly perfect for separating the axon, basal, and apical supercategory (the union of oblique, apical, apical tuft and apical trunk) compartments, but was less consistent for separating the sub-compartments of the apical supercategory.



d

Time Per Edit During Multi-Soma Splitting
Auto Proof vs. Soma to Soma Path Guided

Method	Time (s)	# of Edits	Time per Edit (s/edit)	Speed Ratio vs No Auto Proof
Soma to Soma Path Guided	10285126	19035	540.33	1.00
Auto Proof Guided	816796	5736	142.40	3.79

e

High Confidence Merge Edits Validation

Merge Type	Accuracy (Excitatory)	# of Datapoints (Excitatory)	Accuracy (Inhibitory)	# of Datapoints (Inhibitory)
High Degree Axon	0.99	201	0.99	296
Axon on Dendrite	0.94	869	0.96	107

Fig. 10. Proofreading Validation All validation was performed by the proofreading team at Johns Hopkins University Applied Physics Laboratory (APL). In an initial round of validation, suggested error locations were evaluated in the context of splitting multi-soma cells in the MICrONS volume. As a result we were able to measure both the accuracy of these proofreading rules and the speed benefits of a semi-supervised approach compared to fully-manual proofreading. Additionally, the accuracy of two automatic proofreading rules with high-confidence parameters (axon on dendrite, high degree axon on axon merges) were evaluated. **a** Validation of split locations predicted by automatic multi-split algorithm. "Yes" (indicates that the proposed split can be executed immediately), "Error Nearby" indicates that the split location is correct within 20 μm, but that the human proofreader slightly modified the suggested split points (this threshold was determined so that every suggestion in this category would prove useful to and increase the speed of human proofreaders after their attention is drawn to the relevant location), and "No" indicates that the true split location was far from the predicted location or no merge error was detected by the human proofreader). The heuristic splitting rules are applied in the order indicated by the legend. The automated proofreading accuracy varied substantially over the different heuristic rules with an overall accuracy of 76.12 % when "Yes" and "Error Nearby" are considered true positives. The best-performing rules can be selected for different datasets. **b** Even for a single rule, thresholds can be tuned to optimize performance. Manual classification of split locations predicted by the "Double Back" rule as a function of the angle measured at each predicted location illustrates that a higher accuracy could be achieved by setting a higher threshold for this algorithm. **c** Manual classification of split locations predicted by the width jump rule as a function of the width jump at each predicted location, illustrating another example where interpretable thresholds can be adjusted for higher precision. **d** Time statistics collected as humans performed manual tasks of splitting multi-soma neurons either using a tool that showed the path along the mesh between two somas or using the suggested split locations from our automatic multi-split algorithm. The speed at which humans could apply cuts in the correct locations more than tripled when using suggestions provided by the NEURD multi-split algorithm. Note: the validation is measured as the average amount of time for a single edit in the multi-soma splitting process; a single multi-soma split might require 20 or more edits to completely resolve the merge. **e** Accuracy of two automatic proofreading rules with high-confidence parameters (high degree axon on axon, axon on dendrite).

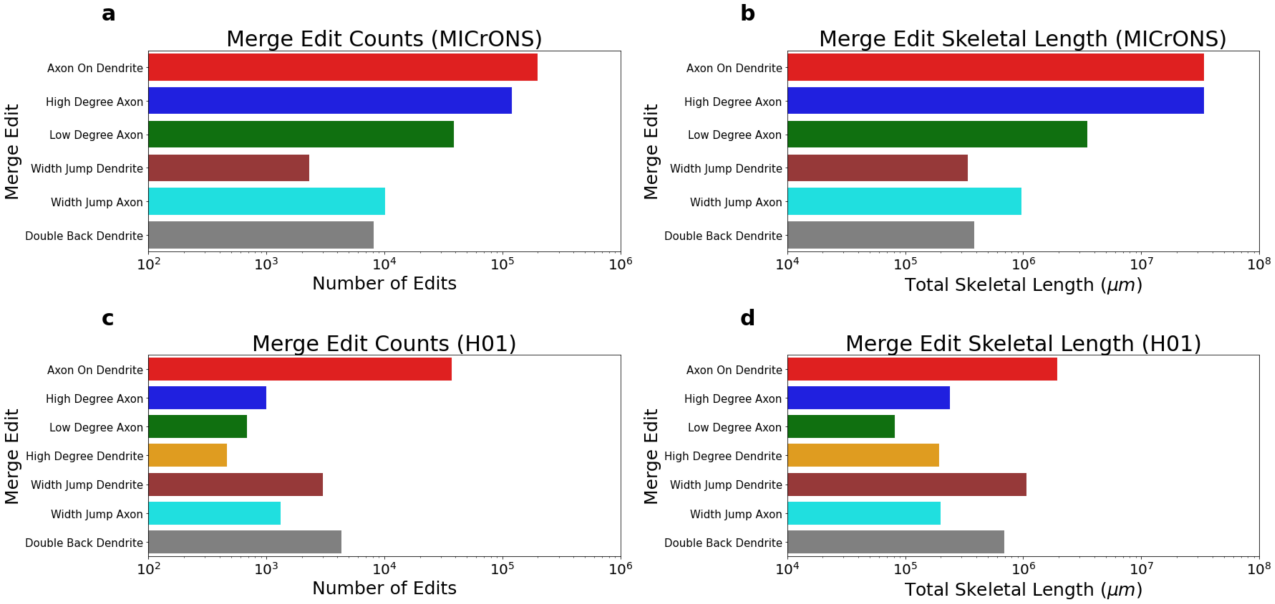


Fig. 11. Counts and total skeletal length of merge errors corrected during automated proofreading. The different heuristic rules are presented from top to bottom in the order that they are implemented in the automated proofreading workflow. All error segments identified are then returned to the set of non-nucleus-associated fragments and are not included in the morphological or connectivity analyses. Note that errors identified by rules later in the workflow are excluded from these statistics if they are found on already-errored segments identified earlier in the workflow. **a** (MICrONS) Total number of separate locations where a specific heuristic rule corrected a merge error. **b** (MICrONS) Total skeletal length eliminated by each heuristic rule. **c** (H01) Total number of separate locations where a specific heuristic rule corrected a merge error. **d** (H01) Total skeletal length eliminated by each heuristic rule.

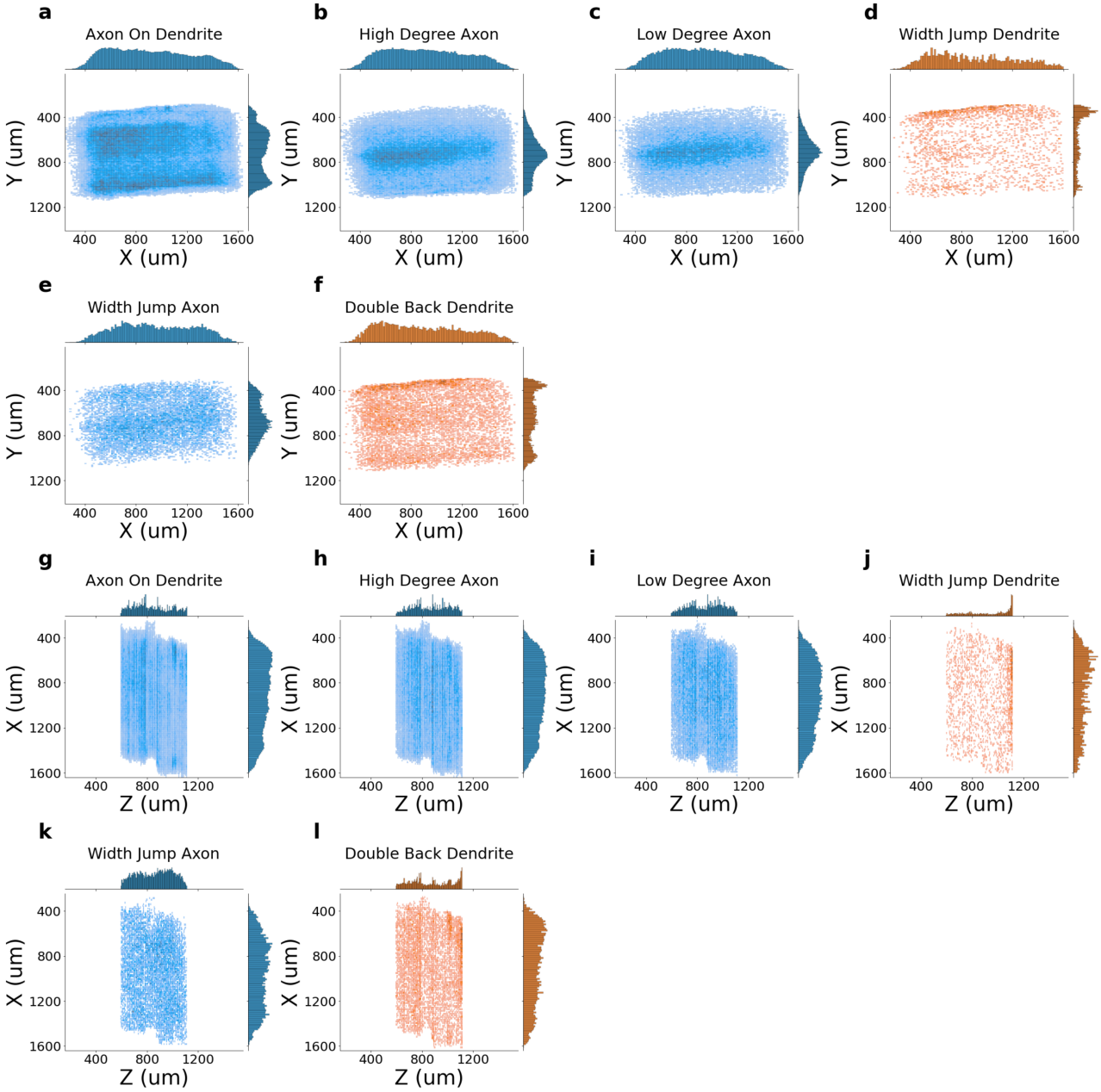


Fig. 12. Spatial Distribution of MICrONS Merge Edit Locations. The distribution of locations show biases in the volume for certain types of merge edits. These spatial biases may be due to segmentation or slicing defects, or differences in the concentration of different kinds of neuropil throughout the volume. **a - f** X,Y merge edit locations for different heuristic rules **g - l** X,Z merge edit locations for different heuristic rules

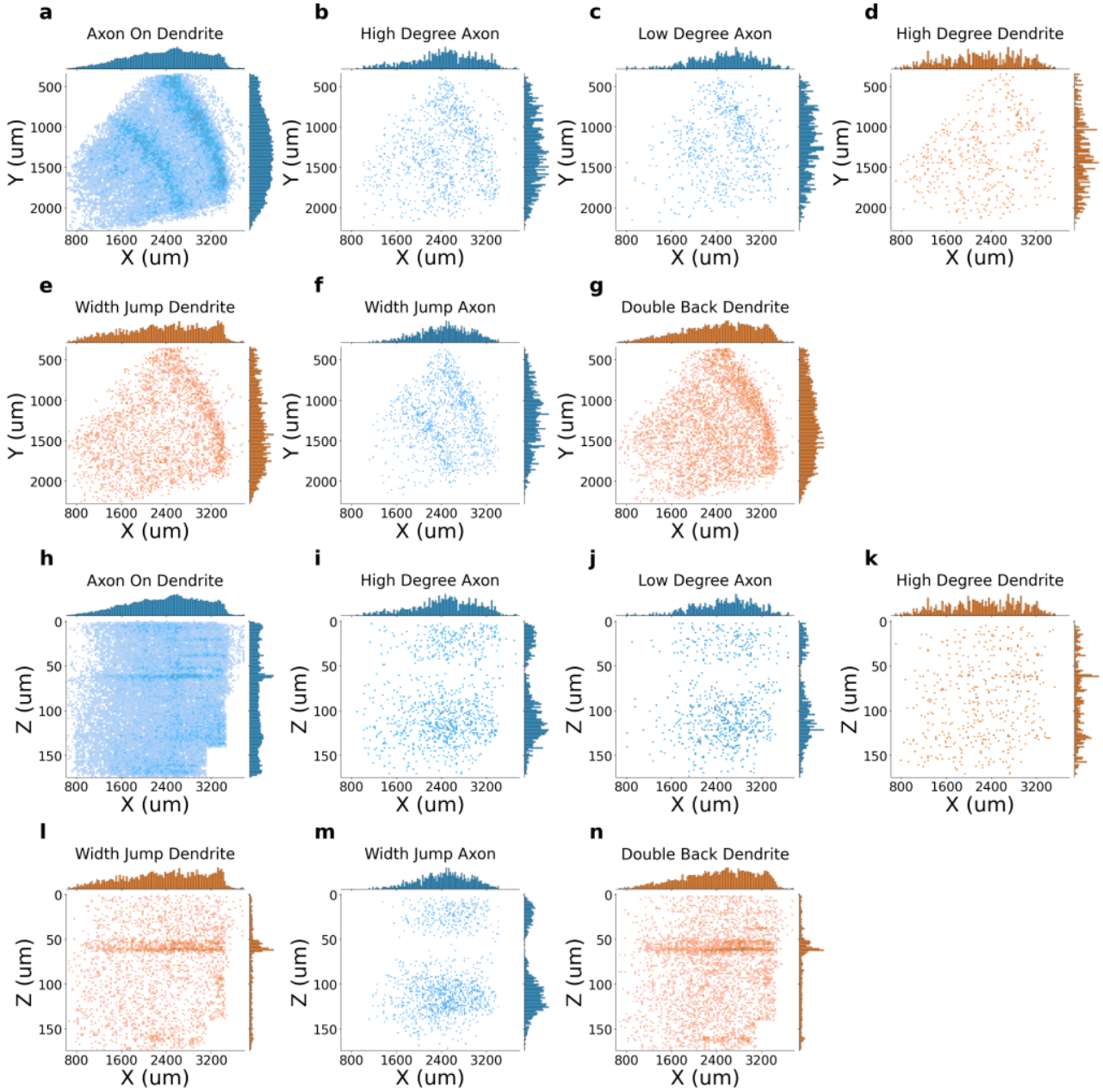


Fig. 13. Spatial Distribution of H01 Merge Edit Locations. The distribution of locations show biases in the volume for certain types of merge edits. These spatial biases may be due to segmentation or slicing defects, or differences in the concentration of different kinds of neuropil throughout the volume. **a - g** X,Y merge edit locations for different heuristic rules **h - n** X,Z merge edit locations for different heuristic rules

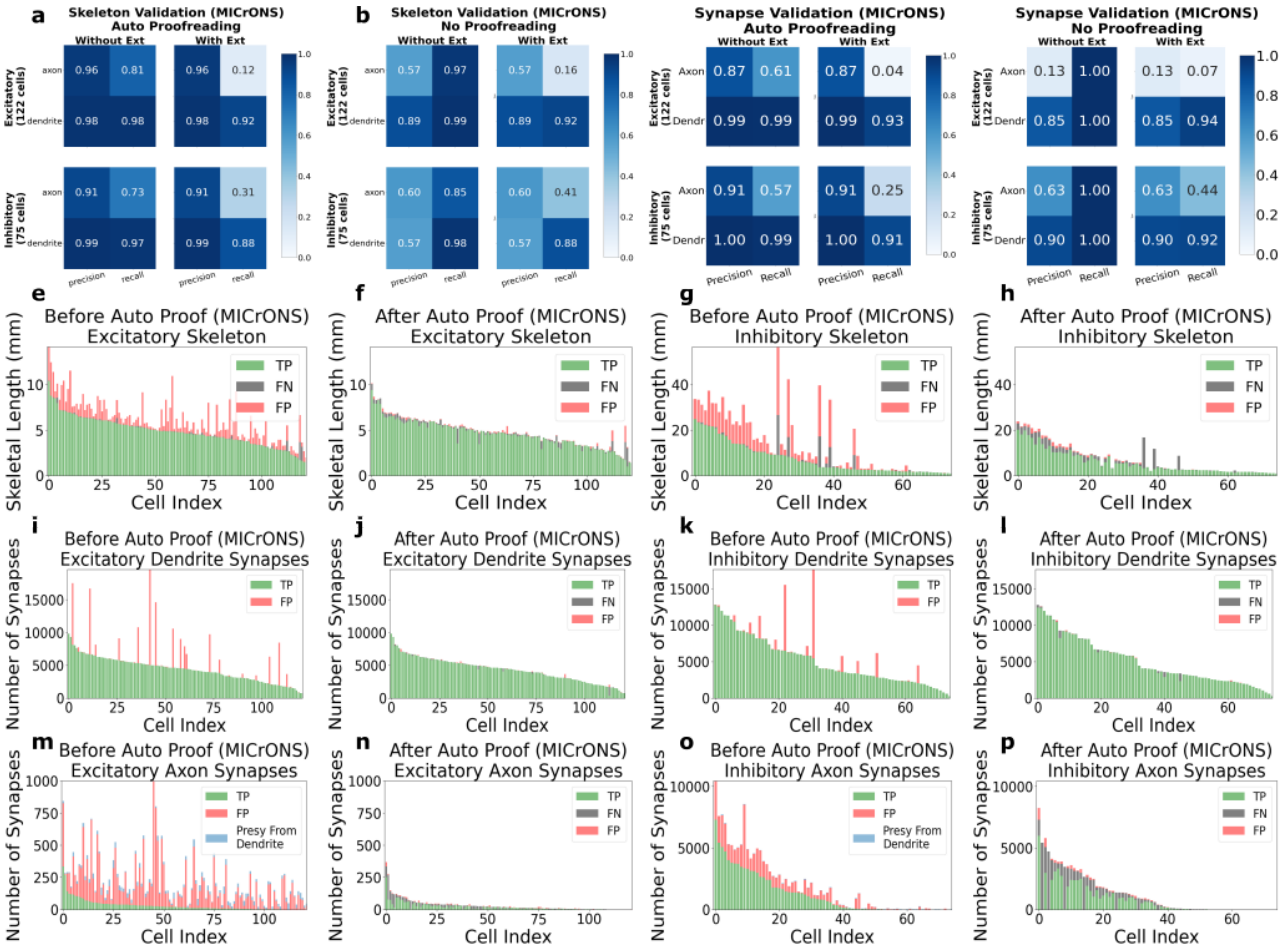


Fig. 14. Supplemental MICrONS Auto Proofreading Validation. Validation metrics and visualizations of the automatic proofreading step when comparing the edits made by the automatic proofreading algorithm to edits made by human proofreaders. The two different columns ("With Ext" and "Without Ext") for the confusion matrices in panels a - d represent comparisons with two possible sources of human proofreading ground truth. The "With Ext" column refers to the skeleton or synapse state after automatic proofreading (panels a,c) or in the raw un-proofread segmentation (panels b,d) compared to the state after a human proofreader both cleaned the existing cell of merge errors and added back missing axon and dendrite segments. The "Without Ext" column performs this same comparison, but assuming that human proofreaders ONLY cleaned merge errors (without performing any extensions). The number of cells in the test set were 122 excitatory and 75 inhibitory. Histograms (panels e - p) give a visual representation of the metrics reported in the precision/recall tables (panels a - d). False Negative (FN) classifications can exist before automatic proofreading because of dropping axon/dendritic segments in the mesh and graph processing pipeline prior to the automatic proofreading step. Note: neurons with multi-soma merges are included in these visualizations and metrics. **a** The precision/recall metrics comparing the skeleton length of cells after automatic proofreading for the Axon/Dendrite compartments and for different exc/inh cell types when compared to human proofreading with extension and without. **b** The precision/recall metrics comparing the skeleton length of cells with no automatic proofreading. **c** The precision/recall metrics comparing the synapse counts of cells with automatic proofreading. **d** The precision/recall metrics comparing the synapse counts of cells with no proofreading. **e - h** TP/FN/FP classification of each test cell's skeletons before and after automatic proofreading for both excitatory and inhibitory cells, demonstrating a large percentage of the FP skeleton segments are removed after the process. **i - l** TP/FN/FP classification of each test cell's dendrite synapses (postsyns) before and after automatic proofreading for both excitatory and inhibitory cells, demonstrating a large percentage of the FP postsyns are removed after the application of dendrite proofreading heuristics. **m - p** TP/FN/FP classification of each test cell's axon synapses (presyn) before and after automatic proofreading for both excitatory and inhibitory cells, demonstrating a large percentage of the FP presyns are removed after the application of axon proofreading heuristics. Those axon presyns located not on the main axon but on dendritic segments are filtered away and designated as "Presyn From Dendrite", which does not include the heuristic rule of "Axon on Dendrite" but instead just filters away any presyns located on dendritic segments that were not filtered away using the heuristic rules.

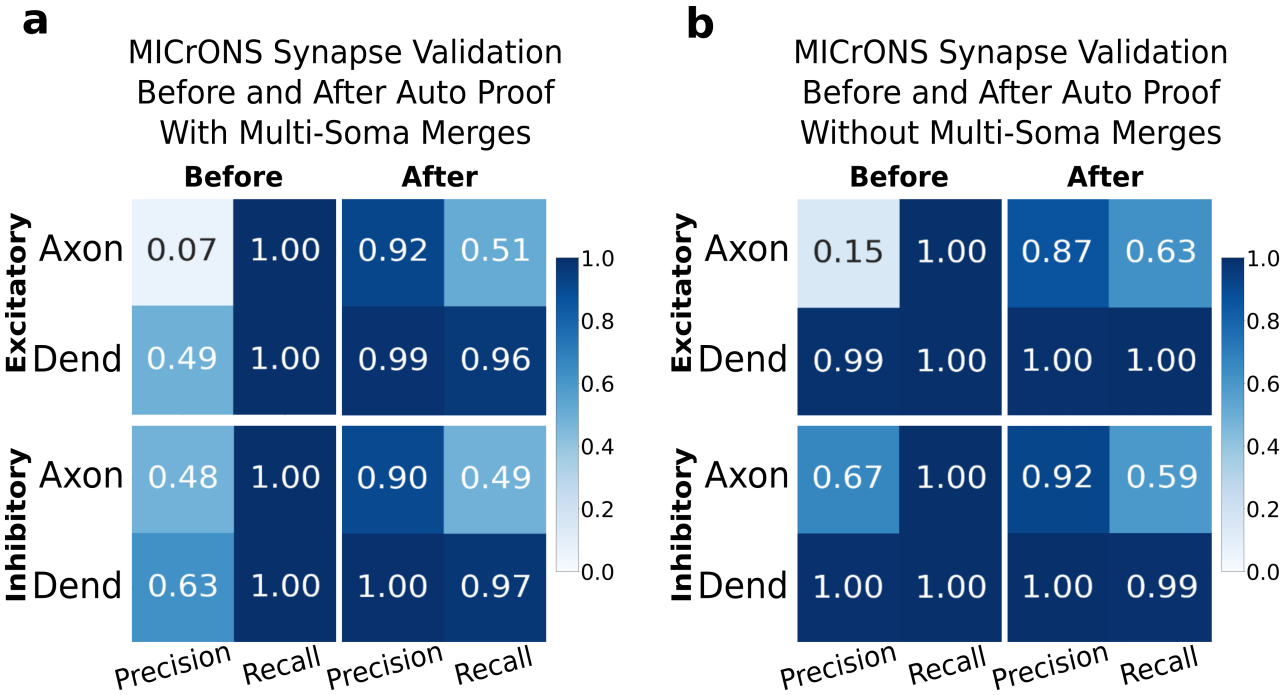


Fig. 15. Supplemental MICrONS Multi-Soma Auto Proofreading Validation. The precision/recall metrics comparing the synapse counts of cells before and after automatic proofreading for the Axon/Dendrite compartments and for different excitatory/inhibitory cell types when compared to human proofreading. **a** Validation when only considering neurons with at least one soma merge to the main segment (19 excitatory, 12 inhibitory). **b** Validation when only considering neurons with no soma merged to the main segment (103 excitatory, 76 inhibitory).

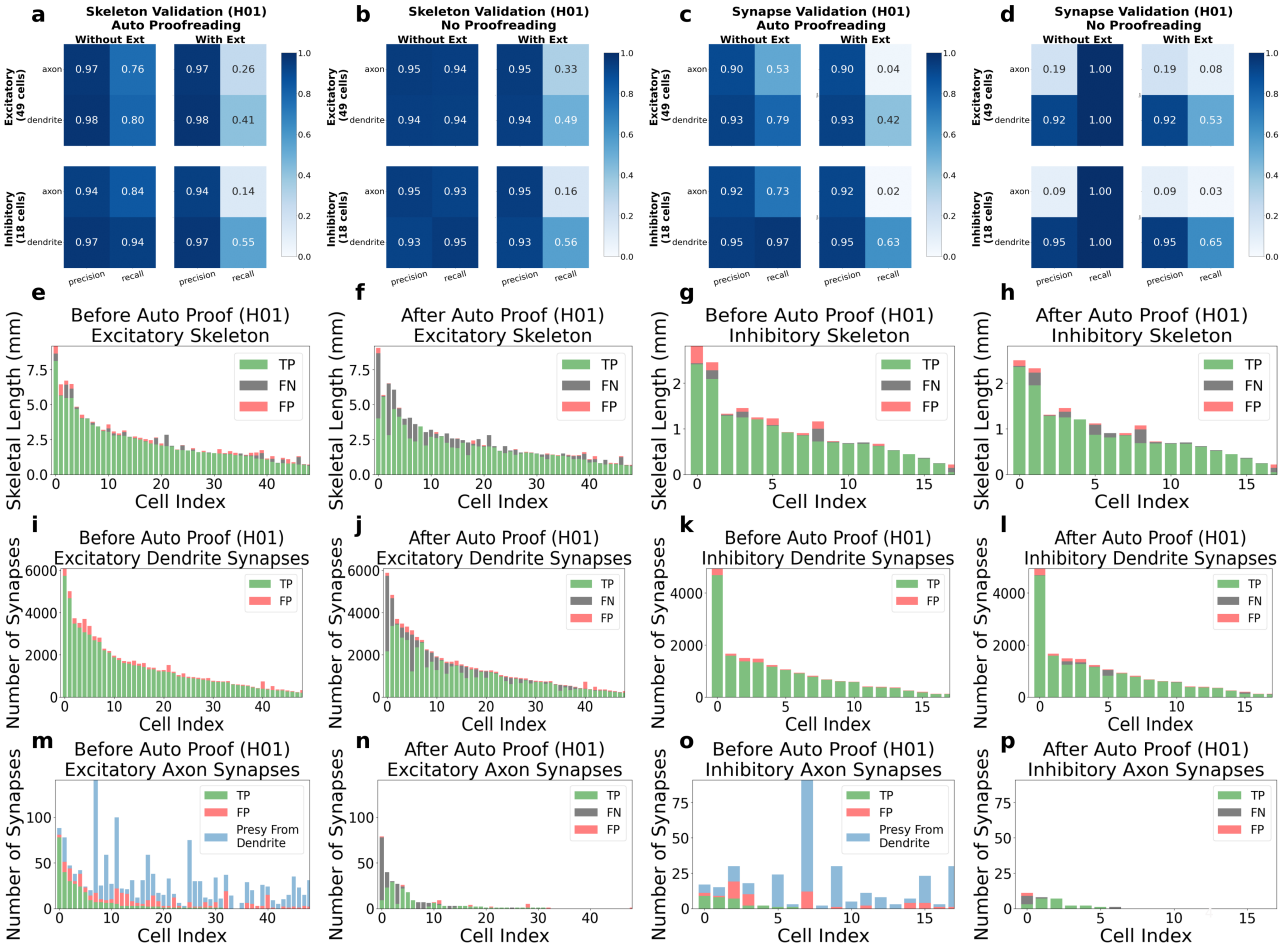


Fig. 16. Supplemental H01 Auto Proofreading Validation. Validation metrics and visualizations of the automatic proofreading step when comparing the edits made by the automatic proofreading algorithm to edits made by human proofreaders. The two different columns ("With Ext" and "Without Ext") for the confusion matrices in panels a - d represent comparisons with two possible sources of human proofreading ground truth. The "With Ext" column refers to the skeleton or synapse state after automatic proofreading (panels a,c) or in the raw un-proofread segmentation (panels b,d) compared to the state after a human proofreader both cleaned the existing cell of merge errors and added back missing axon and dendrite segments. The "Without Ext" column performs this same comparison, but assuming that human proofreaders ONLY cleaned merge errors (without performing any extensions). The number of cells in the test set were 49 excitatory and 18 inhibitory. Histograms (panels e - p) give a visual representation of the metrics reported in the precision/recall tables (panels a - d). FN classifications can exist before automatic proofreading because of dropping axon/dendritic segments in the mesh and graph processing pipeline prior to the automatic proofreading step. Note: While perfectly extending all axonal and dendritic processes is not yet possible, the extent to which neurons were extended in the manually proofread set from the H01 dataset are much less extensively extended in comparison to those of the MICrONS dataset; therefore, the recall numbers for the "With Ext" categories in the H01 validation are much more likely an over-estimate in comparison with those of the MICrONS dataset. **a** The precision/recall metrics comparing the skeleton length of cells after automatic proofreading for the Axon/Dendrite compartments and for different exc/inh cell types when compared to human proofreading with extension and without. **b** The precision/recall metrics comparing the skeleton length of cells with no automatic proofreading. **c** The precision/recall metrics comparing the synapse counts of cells with automatic proofreading. **d** The precision/recall metrics comparing the synapse counts of cells with no proofreading. **e - h** TP/FN/FP classification of each test cell's skeletons before and after automatic proofreading for both excitatory and inhibitory cells, demonstrating a large percentage of the FP skeleton segments are removed after the process. **i - l** TP/FN/FP classification of each test cell's dendrite synapses (postsyns) before and after automatic proofreading for both excitatory and inhibitory cells, demonstrating a large percentage of the FP postsyns are removed after the application of dendrite proofreading heuristics. **m - p** TP/FN/FP classification of each test cell's axon synapses (presyn) before and after automatic proofreading for both excitatory and inhibitory cells, demonstrating a large percentage of the FP presyns are removed after the application of axon proofreading heuristics. Those axon presyns located not on the main axon but on dendritic segments are filtered away and designated as "Presyn From Dendrite", which does not include the heuristic rule of "Axon on Dendrite" but instead just filters away any presyns located on dendritic segments that were not filtered away using the heuristic rules.

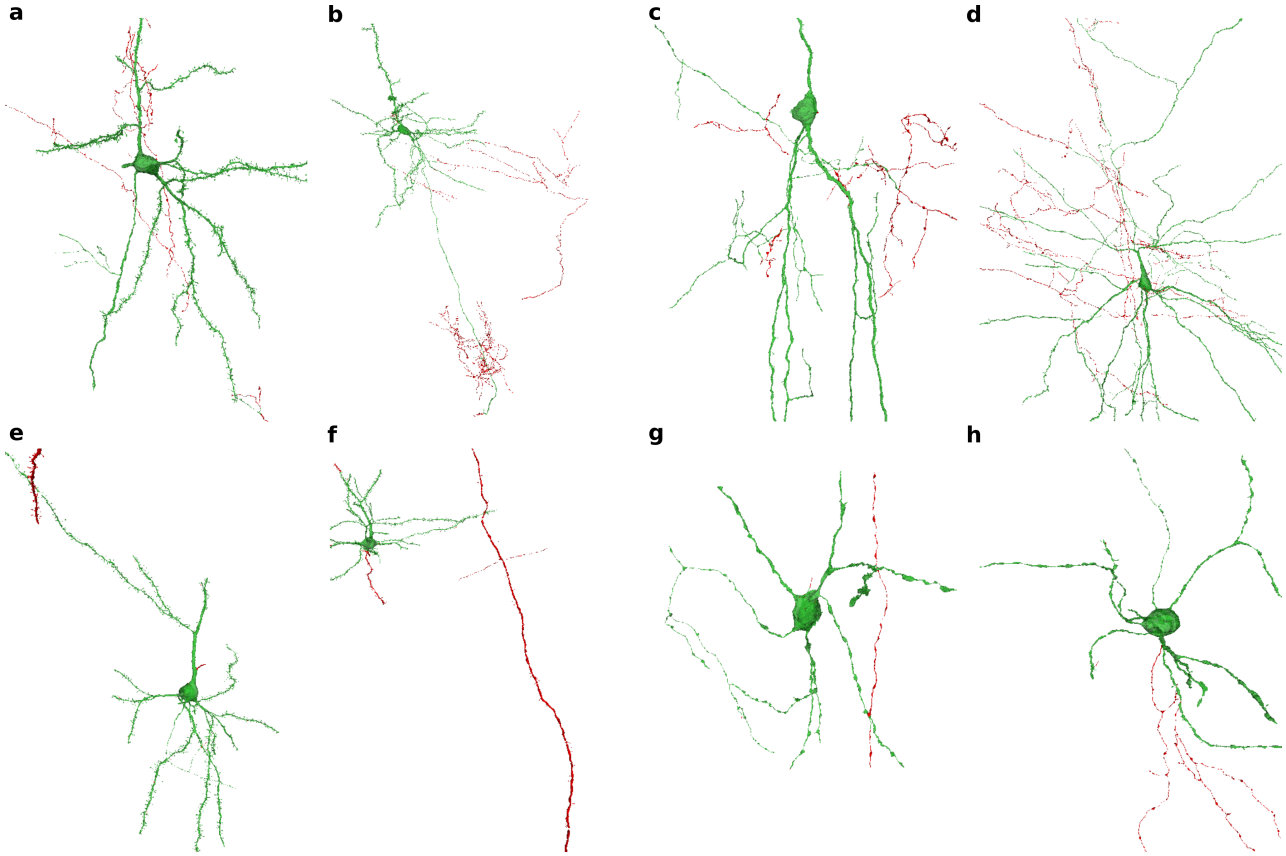


Fig. 17. Proofread Neuron Examples with Merge Errors Labeled Examples of excitatory and inhibitory neurons from both the MICrONS and H01 dataset after automatic proofreading (green) with the removed merge errors shown (red). **a,b** (MICrONS) Example excitatory cells in the 50th and 90th percentile of merge error skeletal length removed. **c,d** (MICrONS) Example inhibitory cells in the 50th and 90th percentile of merge error skeletal length removed. **e,f** (H01) Example excitatory cells in the 50th and 90th percentile of merge error skeletal length removed. **g,h** (H01) Example inhibitory cells in the 50th and 90th percentile of merge error skeletal length removed.

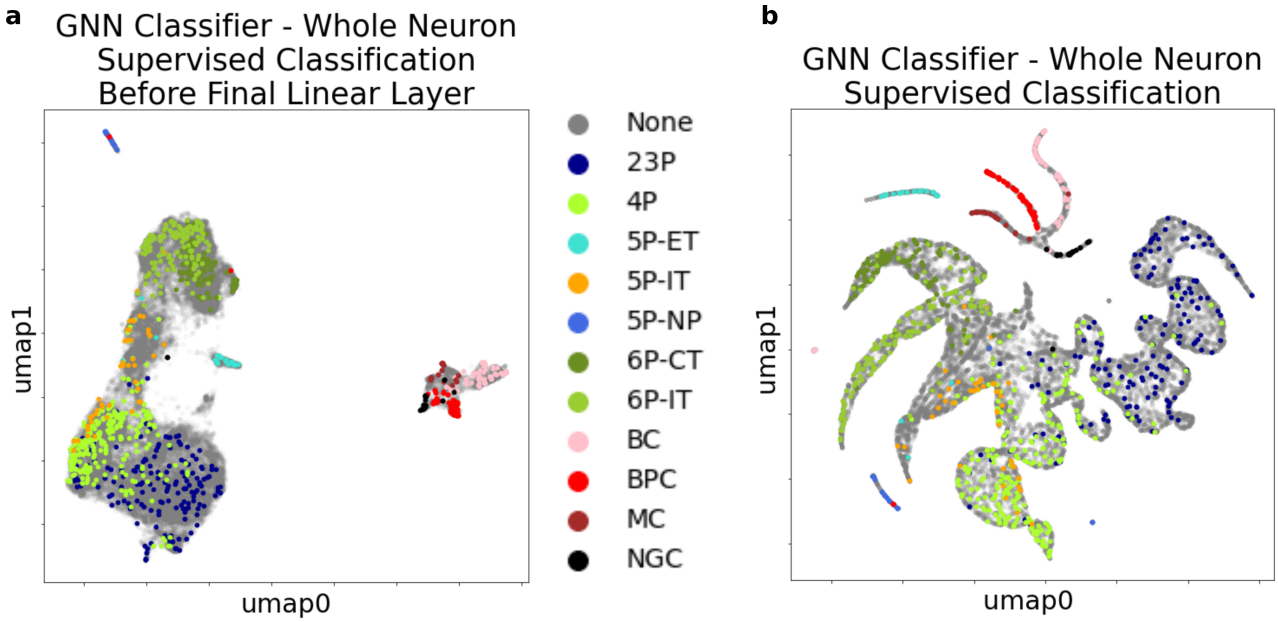


Fig. 18. GNN Classifier Whole Neuron UMAP Embeddings. a Embeddings before the final linear layer and softmax function with hand labeled cells from (Schneider-Mizell et al., 2023) overlaid (these labels were used for the training and validation process of the GNN). Cell-type separation is evident at this stage indicating that the classifier has learned useful features prior to the readout. See Table 1 for the cell-type abbreviation glossary. **b** Embeddings after the final linear layer with cell type labels from (Schneider-Mizell et al., 2023).

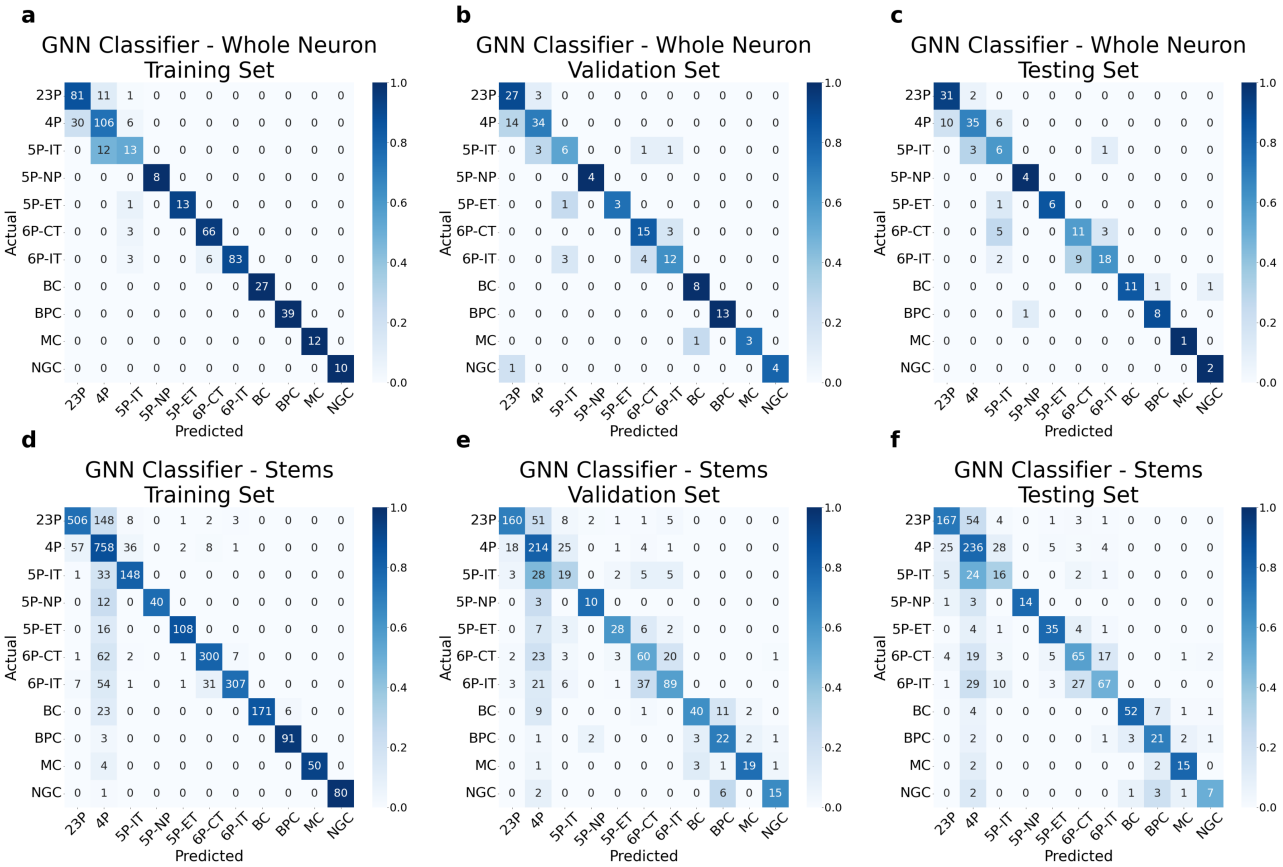


Fig. 19. GNN Classifier Train/Validation/Test Confusion. Confusion matrix of predicted and actual neuron counts from each of the 60:20:20 Train/Validation/Test splits used in the supervised training process of the Graph Neural Network cell type classifier. All ground truth labels were from hand-annotated classes described in (Schneider-Mizell et al., 2023). Color intensities are from normalized values in reference to the summation of a given row. See Table 1 for the cell-type abbreviation glossary. **a-c** Confusion matrix (neuron counts) for the GNN Classifier using the entire neuron graph including the somatic root node. **d-f** Confusion matrix (stem counts) for the GNN Classifier applied to a single stem (a single projection from the soma), without any information about the soma itself.

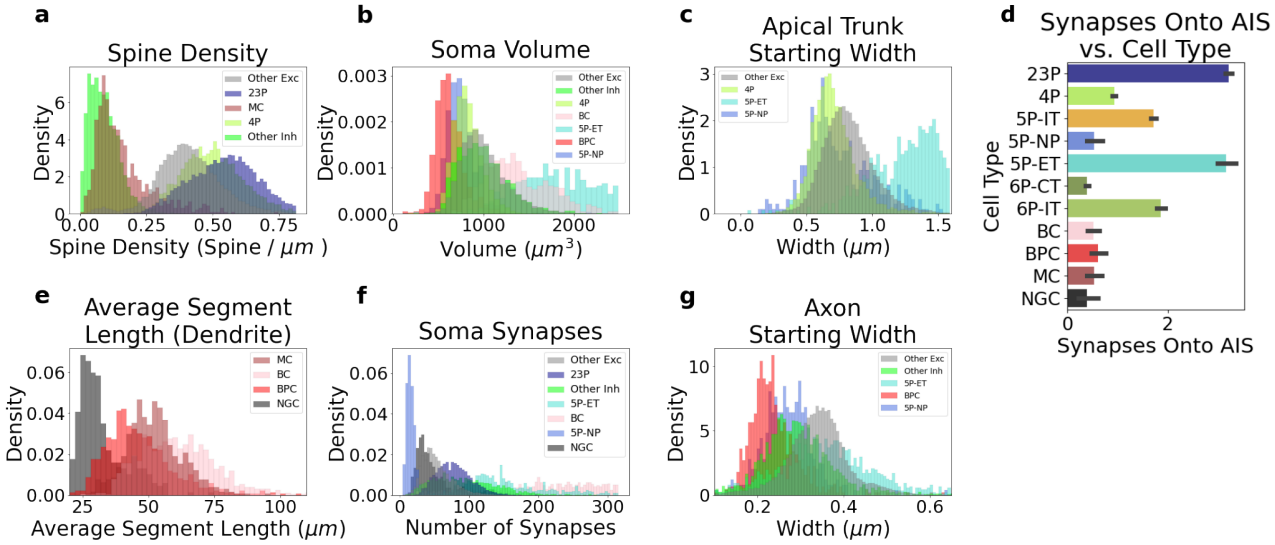


Fig. 20. Various morphological features computed by NEURD. Histograms and bar graphs of a variety of salient features computed by NEURD, colored by the labels generated from the GNN classifier. Some plots are replicating previous work from (Elabbady et al., 2022). **a** Spine density (number of spines per μm of skeletal length) distributions from automatic spine detection. As has been previously reported (Azouz et al., 1997), layer 2/3 pyramidal cells are more densely spiny than layer 4, and MC spine density is higher than other inhibitory cells. **b** Soma volume computed during the soma detection step in the NEURD mesh processing pipeline. As expected, 5P-NP, 4P and BPC generally have smaller somas than other cells from their same excitatory or inhibitory class while 5P-ET and BC are larger than other cells in the same class (Elabbady et al., 2022). **c** Width measurements generated from the average distance of the inner skeleton to the mesh surface (radius approximation) at the beginning of the apical trunk protrusion. Compared to other cell types, 4P and 5P-NP cells have smaller trunks, while 5P-ET are larger. **d** Average number of synapses onto the axon initial segment (AIS) for different cell types. As expected, 23P, 5P-ET, and 5P-IT cell types are more densely innervated on their AIS (Schneider-Mizell et al., 2021). AIS is defined as within 10 μm - 40 μm skeletal distance of the soma, and error bars are standard deviation. **e** Average skeletal length of non-branching dendritic segments for stems of different cell type subclasses, illustrating that NGC have significantly shorter distances between branch points in their dendrites than other inhibitory cells. **f** Distributions of synapses onto the soma illustrating the expected larger average number of soma synapses for 5P-ET and BC and smaller numbers for 5P-NP and NGC (Elabbady et al., 2022). **g** Distributions of radius approximation for the start of the axon protrusion from either a dendrite or the soma, showing smaller typical widths for 5P-NP and BPC and larger starting widths for 5P-ET.

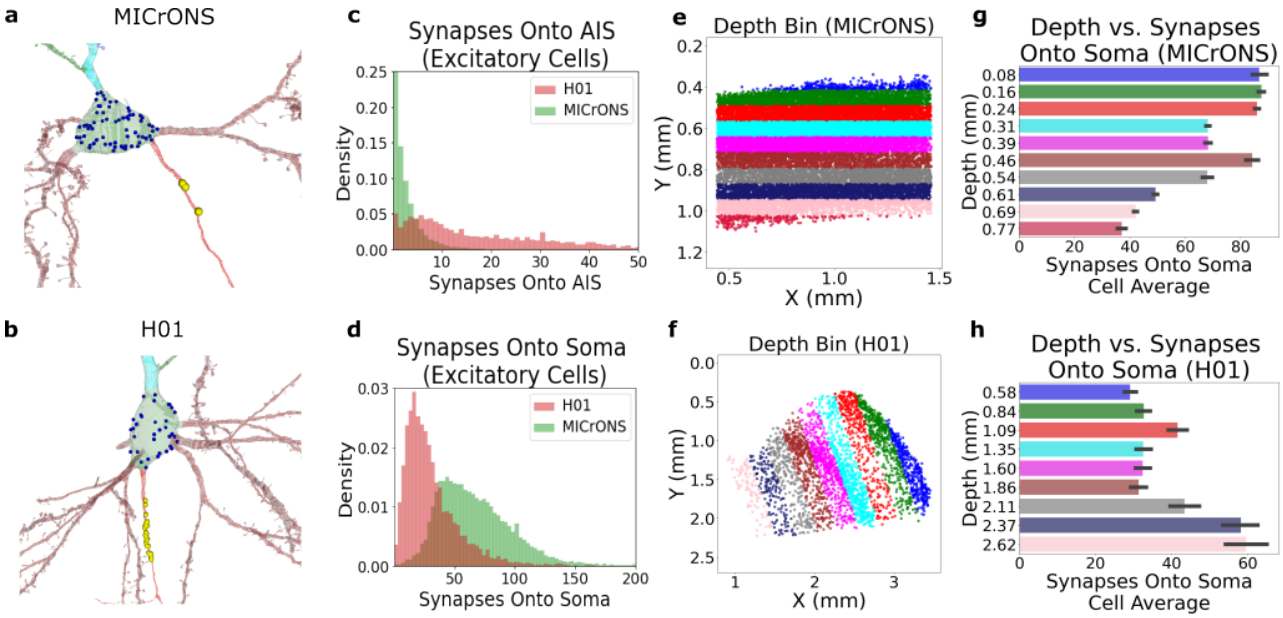


Fig. 21. Comparison of Synapses onto AIS and Soma a-b Example neuron with synapses onto the axon (AIS synapses) in yellow and synapses onto soma in blue. Example neurons are both in the 75th percentile of the number of AIS synapses in their respective volumes. **c** Distribution of the number of AIS synapses per cell compared across datasets, emphasizing the increased innervation of the AIS for neurons in the H01 dataset in comparison to MICrONS. **d** Distribution of the number of soma synapses per cell. As expected, neurons in the MICrONS volume have more identified synapses onto their soma, despite the smaller surface area of mouse somas compared to human (Wildenberg et al., 2021). **e-f** Depth bins used for analysis of both synapses onto AIS vs depth (Main Fig. 5f,g) and synapses onto soma vs. depth (this figure, panels g,h). **g-h** Average number of synapses onto the soma of cells varies across depth (mean +/- std), decreasing in deeper layers of the MICrONS volume, but increasing in deeper layers of the H01 dataset.

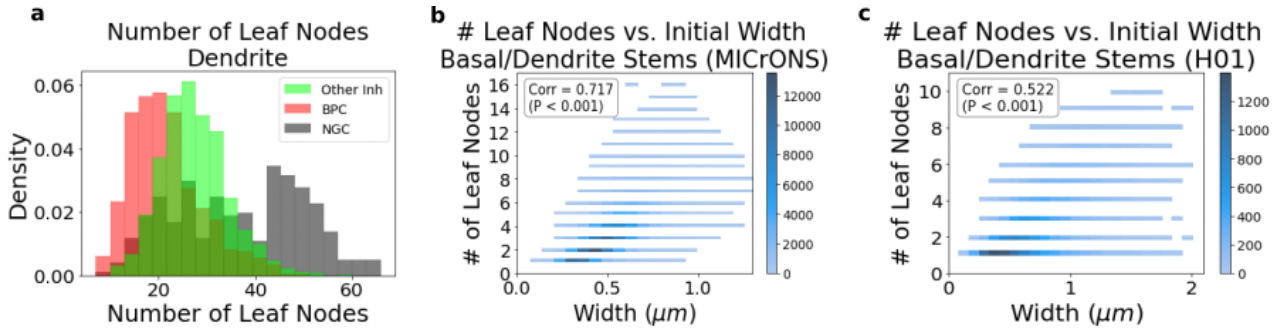


Fig. 22. Neuron Dendritic Branching Characteristics. Measurements related to leaf nodes (terminating ends of the dendritic stem) excluding apical dendrites. **a** Distributions of the number of total leaf nodes for the non-apical dendrites of each neuron separated by inhibitory cell type. As expected, NGC cells have the most leaf nodes of any inhibitory cell type, while BPC have fewer leaf nodes compared to other interneurons. **b-c** Histogram for all the non-apical dendritic stems of every neuron in the volume comparing the initial width of the stem to the number of leaf nodes. For both the MICrONS (b) and H01 (c), there is a significant positive correlation.

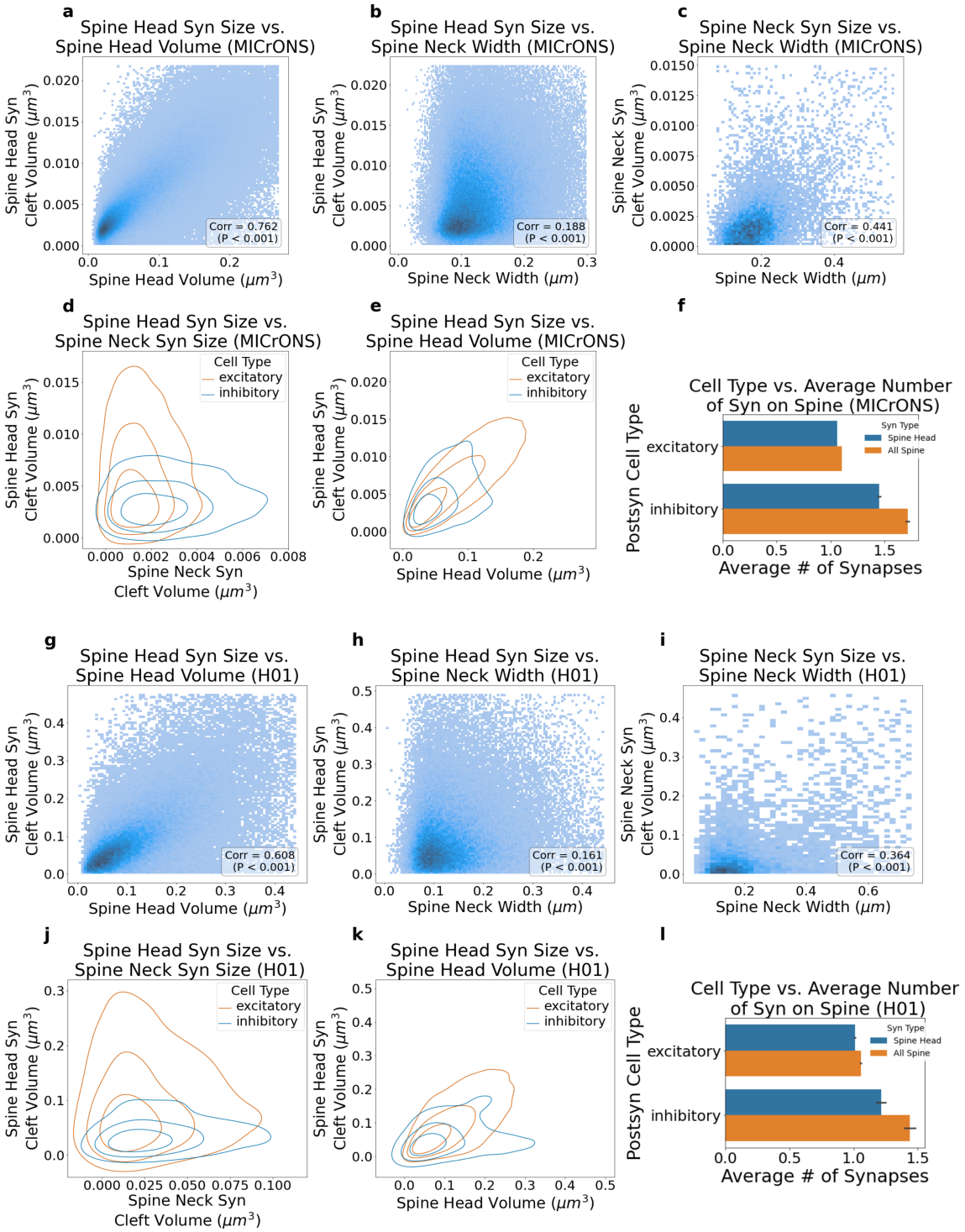


Fig. 23. Postsynaptic Spine Feature Analysis Here we compare the distributions and correlations of certain spine features; replicating and expanding on previous work. The MICrONS dataset is analyzed in **a-f** and the same analysis is repeated for H01 in **g-l**. **a-b,g-h** As expected, for synapses onto the spine **head**, the size of the synaptic cleft and the volume of the spine head mesh are strongly positively correlated, while cleft size and neck width are positively but more weakly correlated (Harris and Stevens, 1989; Arellano et al., 2007). **c,i** For synapses onto the spine **neck**, the width of the spine neck and the synaptic cleft volume of synapses are positively correlated. **d,j** KDE of the joint distribution of the spine neck synaptic cleft volume with spine head synaptic cleft volume for different postsynaptic cell types (exc/inh), illustrating the different joint distributions for each cell type. For synapses onto spine **heads**, synaptic size has a wider range for excitatory cell spines than inhibitory cell spines in both volumes. For synapses onto spine **necks**, the range of synaptic size is larger for inhibitory cells in the MICrONS volume, but similar in the H01 dataset. **e,k** Spine head volume is positively correlated with spine head synaptic cleft volume for both excitatory and inhibitory neurons in both datasets. **f,l** Average number of synapses on all spines and spine heads for different cell types, indicating that in both datasets inhibitory spines receive more synapses per spine than excitatory spines.

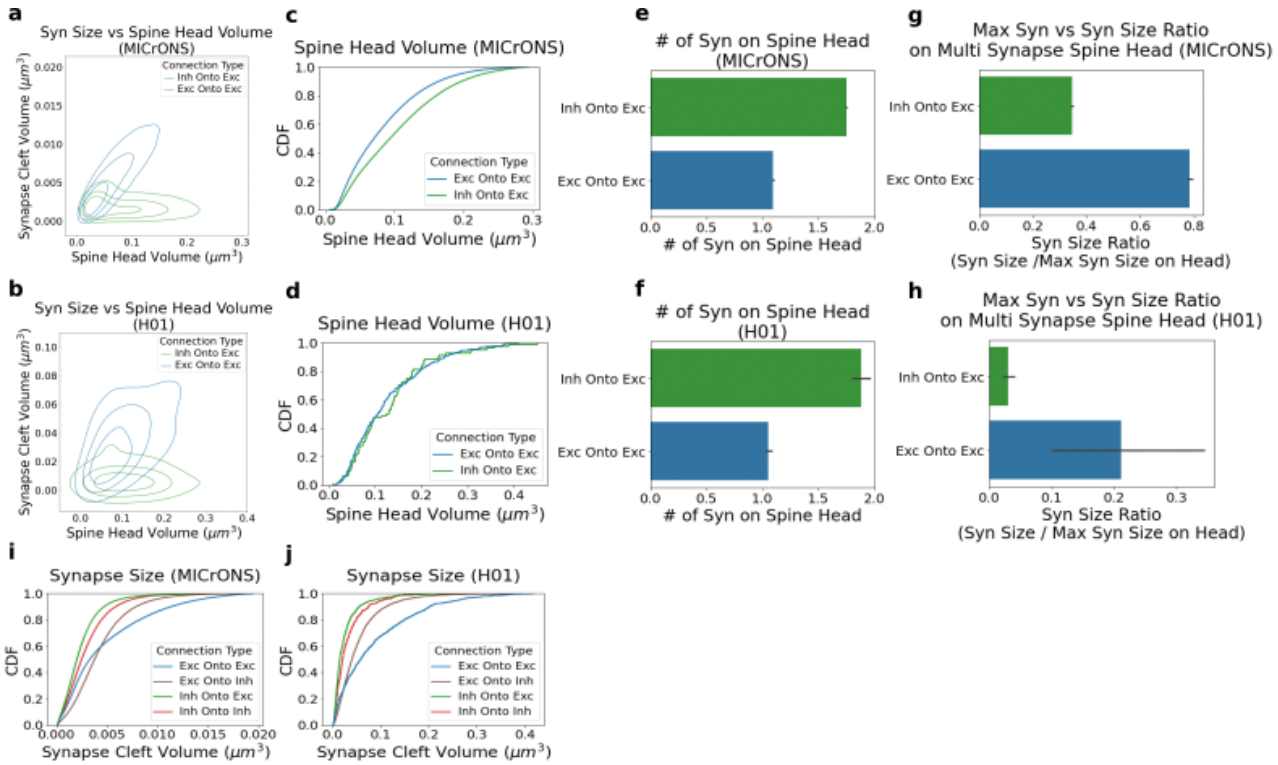


Fig. 24. Spine and Synapse Connectivity Analysis. We revisted the spine analysis in Fig. 23 taking into account information about the identity of the presynaptic neuron for each synapse. **a-b** KDE distribution relating the postsynaptic spine head volume and the synapse cleft volume for synapses onto excitatory cells given different presynaptic cell types. In both datasets a significant positive correlation between spine head volume and synapse size is observed only when the source cell is excitatory but not when the source is inhibitory. **c-d** CDF of the spine head volume for postsynaptic excitatory cells given different presynaptic cell types. For the MICrONS dataset, inhibitory presynaptic cells typically target larger spine heads but this trend is not significant in H01. **e-f** As a possible explanation of why inhibitory cells target larger spine heads, a plot of the average number of synapses on a spine head conditioned on the presynaptic cell type shows that spine heads targeted by inhibitory neurons generally have two synapses as opposed to a mean closer to one synapse per spine for excitatory synapses. **g,h** Expanding on the observation that inhibitory cells typically synapse onto spines with more than one synapse, for spines with multiple synapses, we plot the relative size of a spine head synapse to the size of the largest synapse on that same spine head (mean +/- std) given different presynaptic cell types. We observe that the synapse from an inhibitory source is typically much smaller than the largest synapse on the spine head in both the MICrONS and H01 dataset. **i,j** CDF of the distribution of synapse cleft volumes for different connections types show a similar trend between the MICrONS and H01 dataset where the synapses with excitatory presynaptic cells are typically larger than inhibitory cells and synapses onto inhibitory cells are typically smaller than those onto excitatory.

a

Graph	MICrONS (Raw)	MICrONS (Auto)	H01 (Raw)	H01 (Auto)
# nodes (Exc/Inh)	44683 / 5104	44683 / 5104	2085 / 1236	2085 / 1236
# out edges (Exc/Inh)	2608436 / 1462930	263643 / 692756	12567 / 3924	5646 / 1236
mean in degree (Exc/Inh)	76.37 / 129.14	17.76 / 31.89	3.45 / 7.53	0.88 / 4.09
mean out degree (Exc/Inh)	58.38 / 286.62	5.90 / 135.70	6.03 / 3.17	2.71 / 1.00
mean axon sk length (μm)	1044 / 3256	421 / 1397	593 / 329	510 / 251
mean dendr sk length (μm)	3220 / 3150	2618 / 2330	3248 / 1369	2876 / 1229
% nodes in giant component	99.95	99.58	97.68	83.14
mean s.p. (shortest path)	3.61	4.78	9.20	4.37
mean s.p. undirected	3.06	3.87	6.47	9.73
95th percentile s.p.	5.00	7.00	16.00	9.25
95th percentile s.p. undirected	4.00	5.00	11.00	18.00
mean s.p. along Exc edges	3.90	8.33	13.15	1.00
mean s.p. along Exc edges undirected	3.24	5.31	7.39	9.43
mean s.p Exc-Exc/Inh-Inh	3.60 / 3.50	4.80 / 4.43	9.30 / 9.38	5.10 / 1.83
mean s.p Exc-Exc/Inh-Inh undirected	3.06 / 3.00	3.88 / 3.80	6.44 / 6.51	9.68 / 9.75

Fig. 25. Connectome Network Statistics Table a Network statistics of the MICrONS and H01 connectomes where the nodes are neurons and the edges are the synapses between them (neurons with manual proofreading and their associated synapses are excluded). The term "raw" refers to synaptic data before any processing with NEURD, "auto" refers to the connectome produced after the decomposition pipeline and automated proofreading, "sk" abbreviates "skeletal" when referring to skeletal distances, and "sp" abbreviates "shortest path".

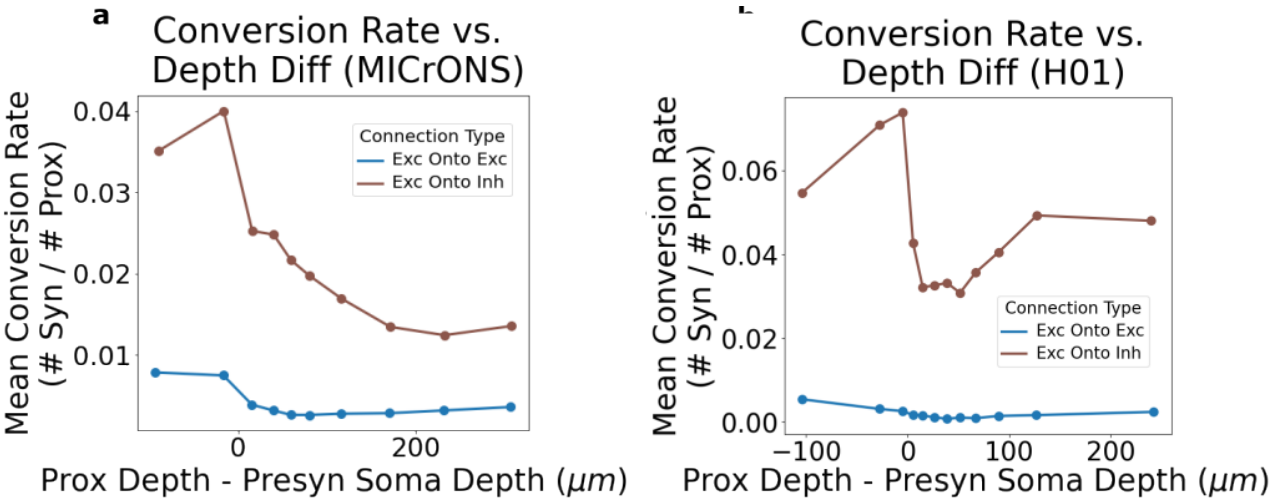


Fig. 26. Conversion Rate vs Proximity Relative Depth. Proximities are binned (approximate equal depth bins) in terms of their relative depth to the presynaptic soma center (proximity depth - presynaptic soma depth) and then the mean conversion rate (number of synapses / number of proximities) for that bin is computed for different connection types. **a** Conversion rate as a function of relative proximity depth in the MICrONS volume, demonstrating that the conversion rate for excitatory connections onto both excitatory and inhibitory postsynaptic cells is greater when the proximity is above the soma (for both connection types) **b** Same plot for H01; the conversion rate as a function of relative proximity depth, demonstrating a greater conversion rate above the soma than below, but with an additional reduction in conversion rates close to the soma that is not seen in MICrONS.

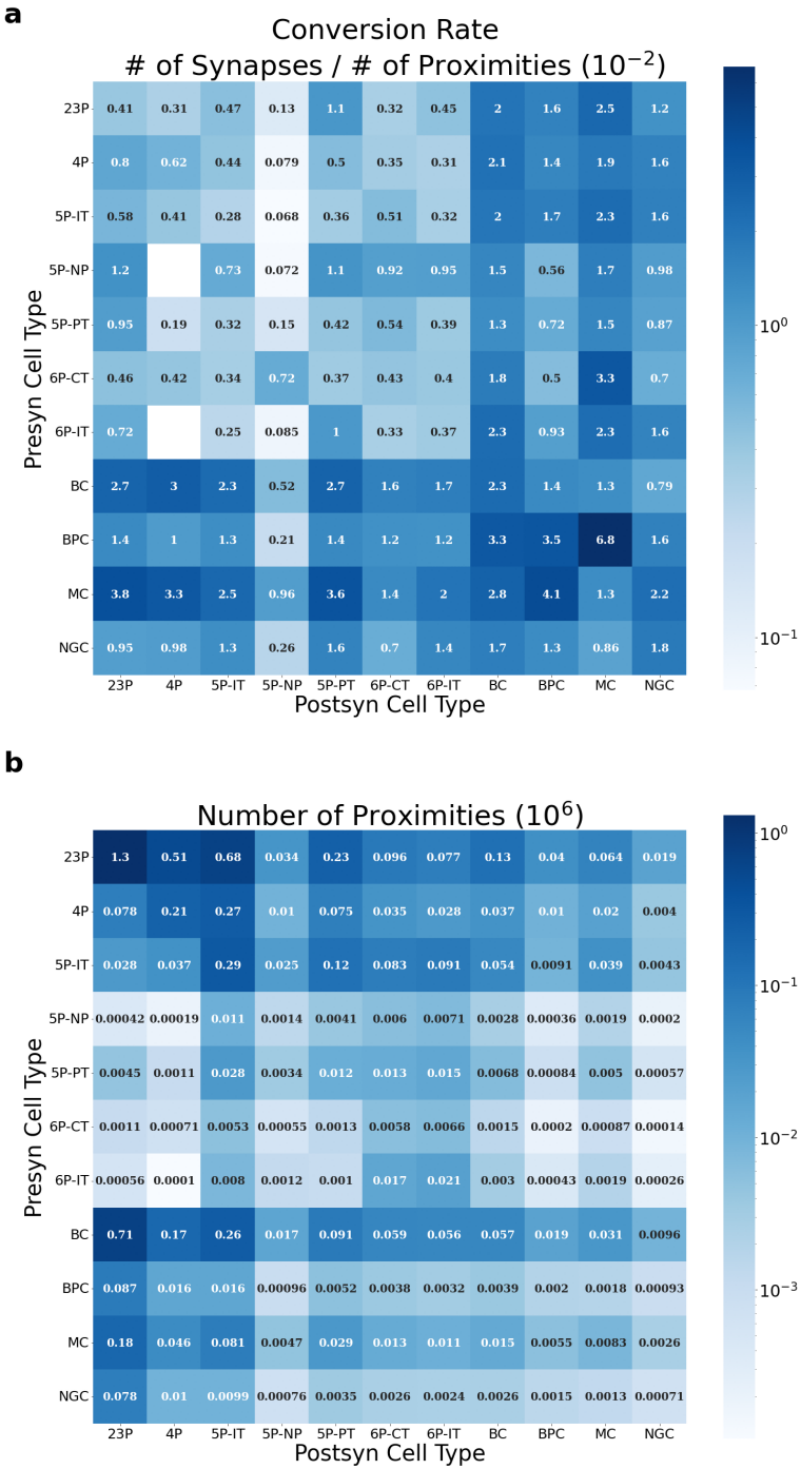


Fig. 27. Conversion Rate Cell Type Matrix The conversion rates (number of synapses / number of proximities) for different presynaptic and postsynaptic cell type pairs. The cell type labels are determined by the GNN whole neuron classifier. Proximities are filtered to only include those with the following features: less than 3 μm proximity distance, dendrite only postsynaptic compartment, presynaptic proximity width less than 130 nm (to exclude myelinated axon), presynaptic and postsynaptic cell type labels with at least a 70% confidence for each from the GNN classifier. **a** Conversion rate for different cell type presynaptic and postsynaptic combinations **b** Number of proximities in dataset used to calculate conversion rate

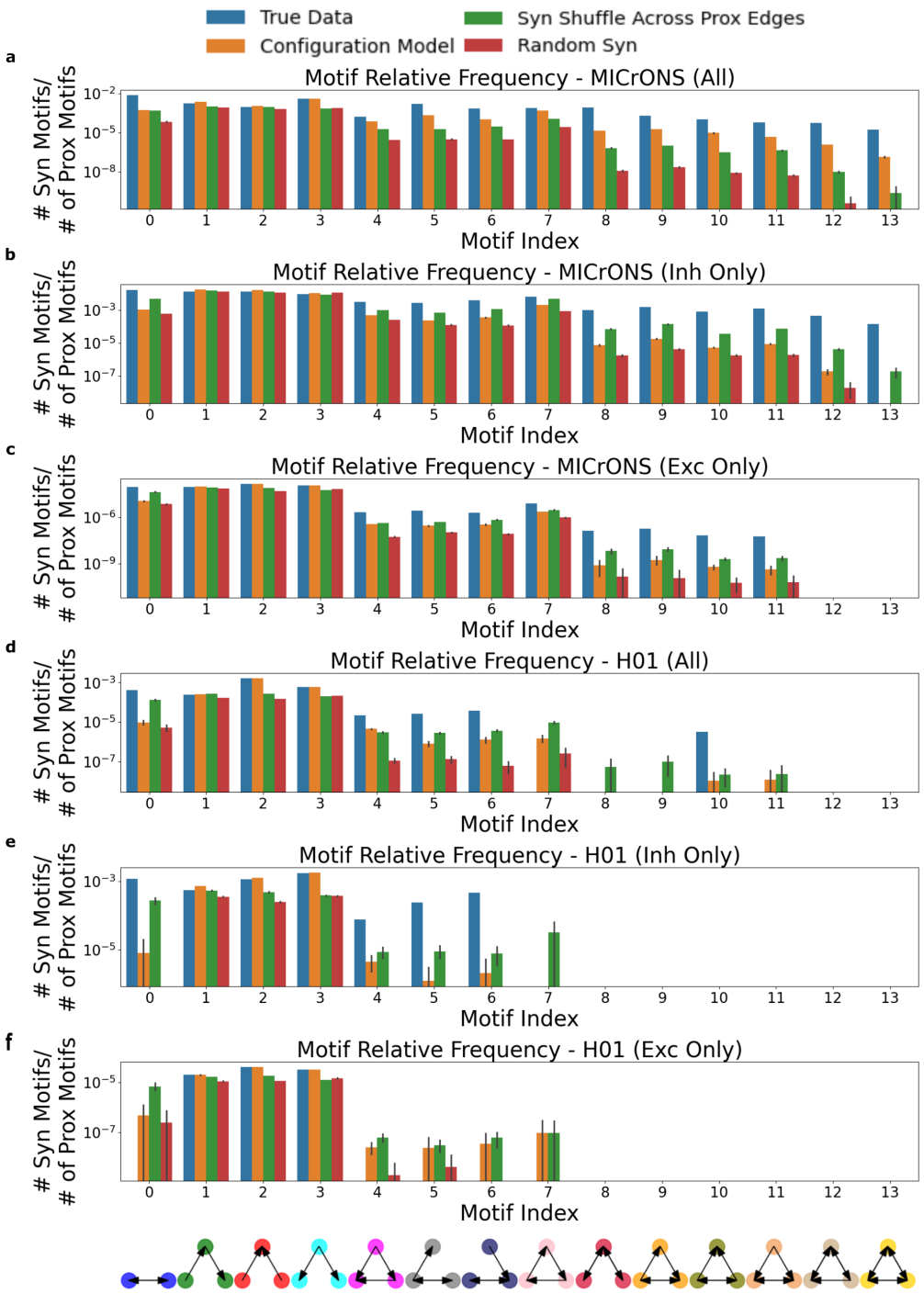


Fig. 28. higher-order Triangle Motif Analysis. Across all automatically-proofread neurons, we count the number of reciprocal connections and directed triangle motifs in the synaptic and proximity connectome and compare the observed ratios to null ratios from three different models: first, a model where synaptic degree distribution is held the same but edges are shuffled (configuration model), second, a model where the synaptic edges are shuffled only between neurons with an existing proximity edge, or third, a model where synapses are randomly shuffled between neurons regardless of proximity. In addition to testing on the entire connectome ("All"), we also performed the same set of comparisons on the inhibitory ("Inh Only") and excitatory ("Exc Only") subgraphs to test whether higher order motif frequencies were consistent in these subpopulations. **a-c** MICrONS dataset relative frequencies (duplicated from Fig. 6j) showing that the relative frequency of higher-order motifs in the synaptic connectome decreases as the number of edges in the motif increases (more higher-order), but are consistently higher than the null model controls (250 random graph samples for each null distribution comparison) for all subgraphs. **d-f** H01 dataset relative frequencies showing that the relative frequency of higher motifs in the synaptic connectome decreases as the number of edges in the motif increases. The observed motif frequencies are again higher than the null model (except in some more edge-dense 3 node motifs in the inhibitory and excitatory only subgraphs), but many of the motifs with more than three directed edges are not observed due to the more incomplete reconstruction of neurons in H01 (400 random graph samples for each null distribution comparison, more samples were computed than MICrONS because computation was faster with a smaller connectome).

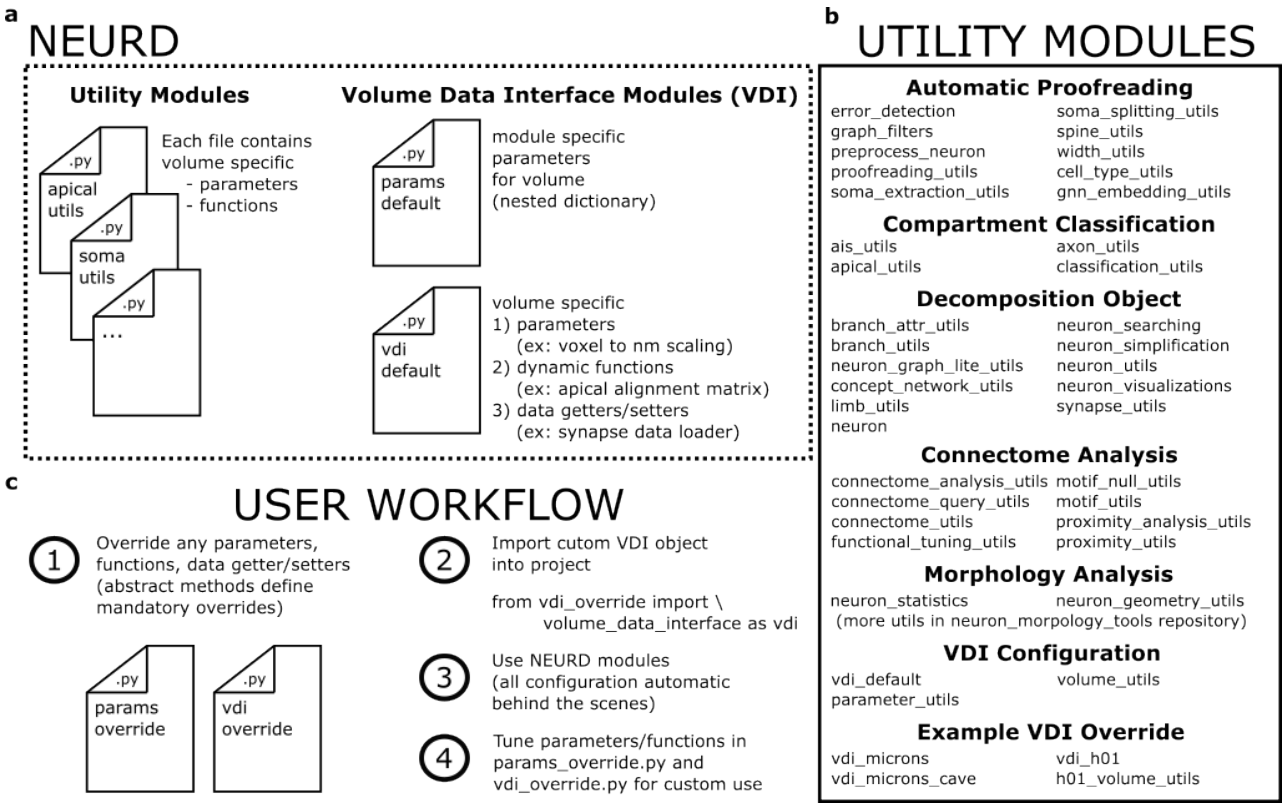


Fig. 29. NEURD Code Package Overview. a The general structure of the code package modules. Most of the modules (.py files) are utility functions that define the functionality pertaining to a specific processing step (ex: *error_dection.py*) or neuron feature (ex: *spine_utils.py*). There are then two .py files that each define a "Volume Data Interface" (VDI). Together they set the user's settings/parameters for NEURD and facilitate data setting and fetching. The VDI modules have default implementations, but NEURD users can override these files for custom implementations (examples of how to override these VDI files are provided in the code documentation). **b** List of utility modules available in NEURD at the time of publication **c** Step by step procedure for how a user would override/configure, import and use the NEURD package

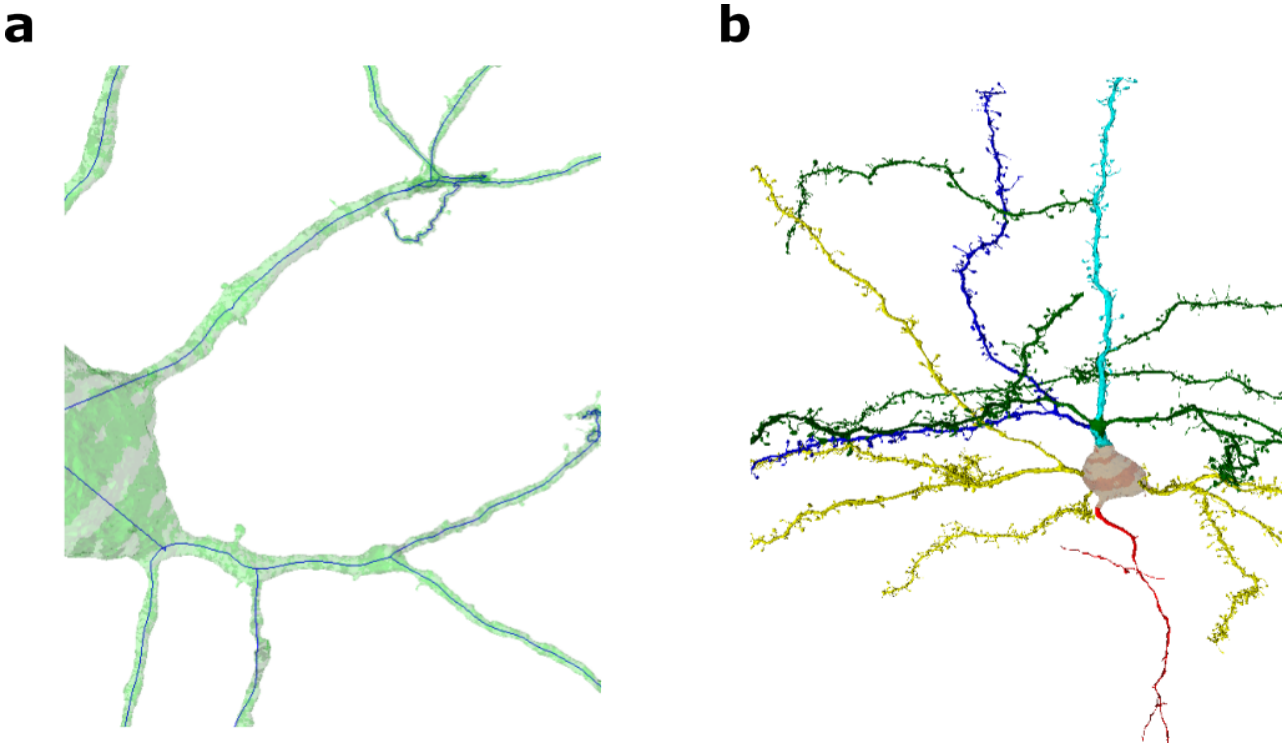


Fig. 30. Skeleton and Compartment Visualization a A general example of a skeleton generated by the NEURD package where most thick dendritic segments are represented by a skeleton that tracks the inside middle of the segment, and thinner segments are represented with surface skeletons. The inhibitory neuron in this example is available for 3-D view and closer inspection using the NEURD Tutorials on the GitHub repository. **b** A general example of compartment labeling of a neurons mesh and skeleton. The excitatory neuron in this example is available for 3-D view and closer inspection using the NEURD Tutorials on the NEURD GitHub repository.

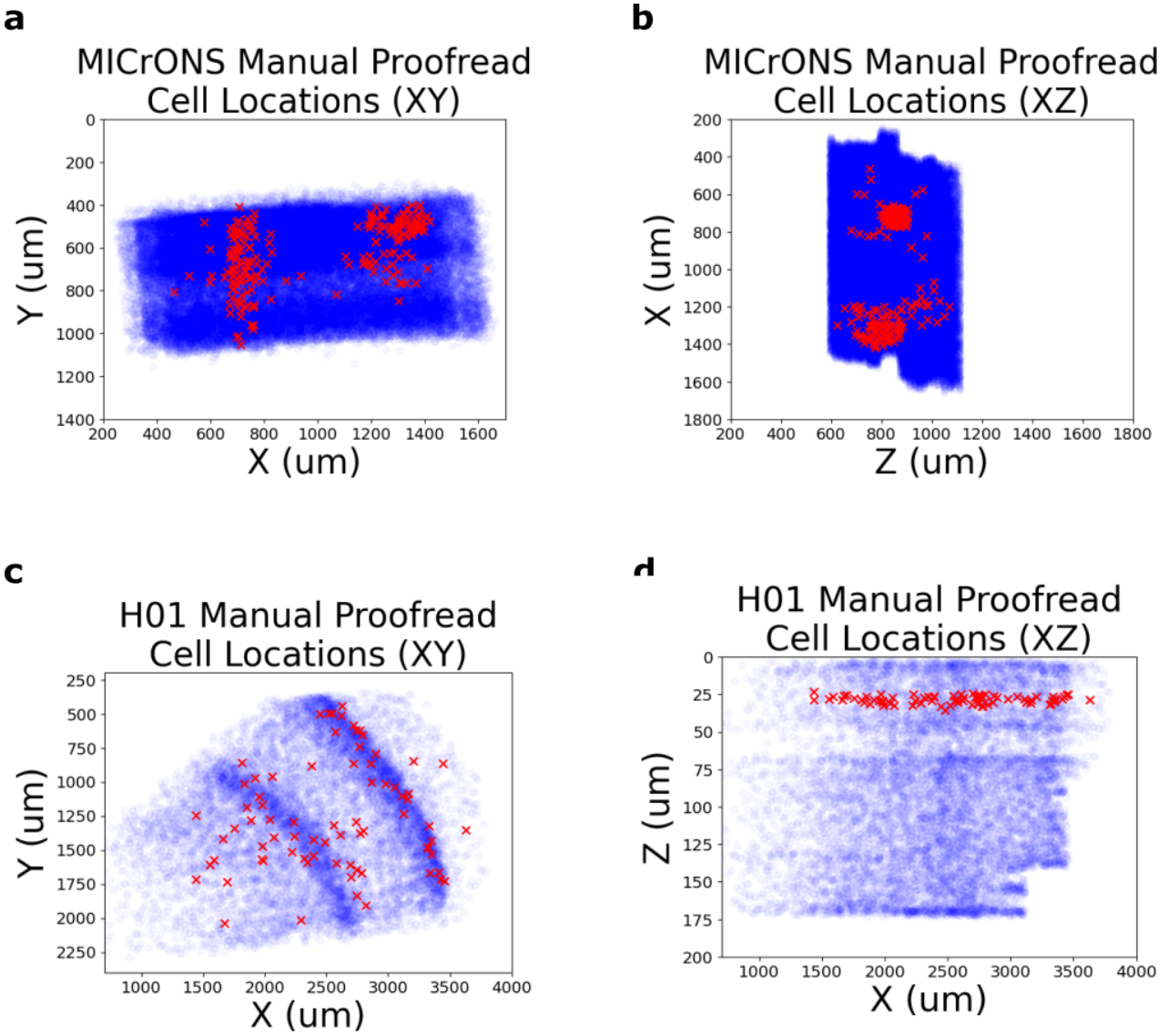


Fig. 31. Manual Proofread Cell Locations a,b Locations in the volume (indicated by red "x" markers) of all manually proofread cells used in the MICrONS validation test set. All cells in the volume are plotted in the background in blue. **c,d** Locations in the volume (indicated by red "x" markers) of all manually proofread cells used in the H01 validation test set. All cells in the volume are plotted in the background in blue.

Appendix Table 1: Cell Type Subclass Glossary

1264

Abbreviation	Cell Type	Excitatory/Inhibitory
23P	Layer 2/3 Pyramidal	Excitatory
4P	Layer 4 Pyramidal	Excitatory
5P IT	Layer 5 Intratelencephalic	Excitatory
5P NP	Layer 5 Near Projecting	Excitatory
5P PT	Layer 5 Pyramidal Track	Excitatory
6CT	Layer 6 Corticothalamic	Excitatory
6IT	Layer 6 Intratelencephalic	Excitatory
BC	Basket Cell	Inhibitory
BPC	Bipolar Cell	Inhibitory
MC	Martinotti Cell	Inhibitory
NGC	Neurogliaform Cell	Inhibitory

Appendix Table 2: Figure Data Size (N)

Figure	Panel	Title / Concept	N	Subclass N
None	None	NEURD pipeline: mesh objects with merged nuclei detected (MICrONS)	115230	None
None	None	NEURD pipeline: soma detection count (MICrONS)	76894	None
None	None	NEURD pipeline: single soma auto-proofread neurons, after non-neuronal filter (MICrONS)	54802	49073 (Exc), 5729 (Inh)
None	None	NEURD pipeline: mesh objects with merged nuclei detected (H01)	15068	None
None	None	NEURD pipeline: soma detection count (H01)	14303	None
None	None	NEURD pipeline: single soma auto-proofread neurons, after non-neuronal filter (H01)	11444	7357 (Exc), 4087 (Inh)
3, 14	MICrONS			Test Set: 122 (excitatory), 75 (inhibitory) Automatic Proofreading Validation Totals: for excitatory axon synapses: tp = 2216, fp = 324, tn = 24430, fn = 1420 for excitatory dendrite synapses: tp = 525072, fp = 3207, tn = 91423, fn = 4321 for inhibitory axon synapses: tp = 66372, fp = 6097, tn = 57164, fn = 47466 for inhibitory dendrite synapses: tp = 450043, fp = 901, tn = 44940, fn = 4625 for excitatory axon skeleton (um): tp = 85128.69, fp = 3797.15, tn = 87424.84, fn = 19831.9 for excitatory dendrite skeleton (um): tp = 466420.83, fp = 7769.81, tn = 52366.92, fn = 7644.89 for inhibitory axon skeleton (um): tp = 239772.53, fp = 23142.88, tn = 238705.66, fn = 90304.64 for inhibitory dendrite skeleton (um): tp = 184898.31, fp = 2695.24, tn = 140840.23, fn = 5136.95 No Proofreading Validation Totals: for excitatory axon synapses: tp = 3636, fp = 24754, tn = 0, fn = 0 for excitatory dendrite synapses: tp = 529393, fp = 94630, tn = 0, fn = 0 for inhibitory axon synapses: tp = 113838, fp = 63261, tn = 0, fn = 0 for inhibitory dendrite synapses: tp = 454668, fp = 45841, tn = 0, fn = 0 for excitatory axon skeleton (um): tp = 111903.43, fp = 84279.14, tn = 0.0, fn = 3305.89 for excitatory dendrite skeleton (um): tp = 472820.13, fp = 61382.31, tn = 0.0, fn = 6629.94 for inhibitory axon skeleton (um): tp = 354128.6, fp = 237797.11, tn = 0.0, fn = 60760.66 for inhibitory dendrite skeleton (um): tp = 190354.31, fp = 143216.42, tn = 0.0, fn = 4426.28
3, 16	H01			Test Set: 48 (excitatory), 18 (inhibitory) Automatic Proofreading Validation Totals: for excitatory axon synapses: tp = 154, fp = 17, tn = 1263, fn = 140 for excitatory dendrite synapses: tp = 57181, fp = 4582, tn = 2882, fn = 14213 for inhibitory axon synapses: tp = 39, fp = 4, tn = 746, fn = 26 for inhibitory dendrite synapses: tp = 25312, fp = 2773, tn = 363, fn = 2927 for excitatory axon skeleton (um): tp = 8149.14, fp = 269.81, tn = -60.51, fn = 2543.82 for excitatory dendrite skeleton (um): tp = 84499.71, fp = 1700.02, tn = -608.63, fn = 21687.95 for inhibitory axon skeleton (um): tp = 2195.75, fp = 141.39, tn = -153.0, fn = 403.54 for inhibitory dendrite skeleton (um): tp = 13287.63, fp = 423.42, tn = 84.65, fn = 878.56 No Proofreading Validation Totals: for excitatory axon synapses: tp = 294, fp = 1280, tn = 0, fn = 0 for excitatory dendrite synapses: tp = 71394, fp = 7464, tn = 0, fn = 0 for inhibitory axon synapses: tp = 65, fp = 750, tn = 0, fn = 0 for inhibitory dendrite synapses: tp = 28239, fp = 3136, tn = 0, fn = 0 for excitatory axon skeleton (um): tp = 10381.38, fp = 520.88, tn = 0.0, fn = 643.22 for excitatory dendrite skeleton (um): tp = 101312.97, fp = 5966.07, tn = 0.0, fn = 6376.24 for inhibitory axon skeleton (um): tp = 2446.3, fp = 141.39, tn = 0.0, fn = 186.16 for inhibitory dendrite skeleton (um): tp = 13624.78, fp = 1049.47, tn = 0.0, fn = 666.19
4	4a	Logistic Reg - E/I Model (MICrONS)	4986	4024 (excitatory), 961 (inhibitory)
4	4b	Logistic Reg - E/I Model (H01)	7555	5800 (excitatory), 1755 (inhibitory)
4	4d	Distribution of Predicted Exc/Inh Neurons By Layer Depth	49117	23P (18720), 4P (7399), 5P-IT (7437), 5P-NP (514), 5P-PT (1232), 6P-CT (4975), 6P-IT (4944), BC (1425), BPC (946), MC (987), NGC (538)
4	4e	GNN Classifier - Whole Neuron	178	23P (41), 4P (40), 5P-IT (20), 5P-NP (5), 5P-ET (6), 6P-CT (20), 6P-IT (22), BC (11), BPC (9), MC (1), NGC (3)
4	4f	GNN Classifier - Stems	1023	23P (203), 4P (379), 5P-IT (62), 5P-NP (14), 5P-ET (49), 6P-CT (104), 6P-IT (92), BC (56), BPC (33), MC (20), NGC (11)
5	5f	Depth vs Synapses onto AIS (MICrONS)	22955	0.08 (447), 0.16 (2646), 0.24 (3250), 0.31 (4023), 0.39 (3463), 0.46 (2048), 0.54 (2321), 0.61 (2749), 0.69 (1730), 0.77 (278)
5	5g	Depth vs Synapses onto AIS (H01)	3154	0.58 (385), 0.84 (464), 1.09 (355), 1.35 (528), 1.60 (529), 1.86 (426), 2.11 (224), 2.37 (160), 2.62 (83)
5	5h	Angle of Thickest Basal (MICrONS)	41309	None
5	5i	Angle of Thickest Basal (H01)	3445	None
5	5j	Average Segment Length Basal	33049	23P (13516), 5P-NP (398), Other Exc (19135)
5	5k	Shaft Synapses to Spine Ratio	30910	23P (11970), 4P (5443), 5P-ET (462), 5P-NP (216), Other Exc (12819)
5	5l	Spine Neck Length	41898	4P (7034), 5P-NP (461), Other Exc (34403)
5	5m	Spine Syn Size vs. Spine Head Volume	45783	5P-ET (846), 5P-NP (500), MC (973), Other Exc (40594), Other Inh (2870)
6	6c	Cell Type Conversion Rates		23P Onto 5P-ET (225805), Other Exc Onto Exc (3994769), 23P Onto MC (64475), 23P/4P/5P-IT Onto BC (217221), Exc Onto MC (128303), 23P Onto BC Soma (2145), Other Exc Onto Inh (95921), BC Onto 23P/5P-ET Dendr (786459), BC Onto Exc Soma (18696), MC Onto 23P/5P-ET Dendr (201853), Other Inh Onto Exc (947601), BC Onto BC Soma (950), BC Onto Inh Soma (2140), BPC Onto MC (1837), MC Onto BPC (5496), Other Inh Onto Inh (152469)
6	6d	Distance To Soma (H01) Basal Dendrite	756521	Exc Onto Exc (113052), Exc Onto Inh (106834), Inh Onto Exc (481022), Inh Onto Inh (55613)
6	6e	Distance To Soma (H01) Basal Dendrite	6454	Exc Onto Exc (700), Exc Onto Inh (4845), Inh Onto Exc (491), Inh Onto Inh (418)

Figure	Panel	Title / Concept	N	Subclass N
1267				
6	6f	Conversion Rate (MICrONS)		Exc Onto Inh: 0.15 (n_prox=54950, n_syn = 8628), 0.3 (n_prox=123804, n_syn = 17121), 0.45 (n_prox=198634, n_syn = 23931), 0.6 (n_prox=266713, n_syn = 28392), 0.75 (n_prox=328434, n_syn = 31149), 0.9 (n_prox=385586, n_syn = 32852), 1.05 (n_prox=440163, n_syn = 33970), 1.2 (n_prox=492759, n_syn = 34676), 1.35 (n_prox=543632, n_syn = 35143), 1.5 (n_prox=593594, n_syn = 35438), 1.65 (n_prox=643038, n_syn = 35658), 1.8 (n_prox=692573, n_syn = 35814), 1.95 (n_prox=741389, n_syn = 35912), 2.1 (n_prox=790363, n_syn = 36017), 2.25 (n_prox=839090, n_syn = 36086), 2.4 (n_prox=888119, n_syn = 36148), 2.55 (n_prox=937351, n_syn = 36211), 2.7 (n_prox=986303, n_syn = 36264), 2.85 (n_prox=1035527, n_syn = 36300), 3.0 (n_prox=1085128, n_syn = 36322) Inh Onto Inh: 0.15 (n_prox=23894, n_syn = 4791), 0.3 (n_prox=50350, n_syn = 8435), 0.45 (n_prox=77195, n_syn = 11298), 0.6 (n_prox=100582, n_syn = 12893), 0.75 (n_prox=121587, n_syn = 13823), 0.9 (n_prox=140984, n_syn = 14278), 1.05 (n_prox=159432, n_syn = 14545), 1.2 (n_prox=177191, n_syn = 14691), 1.35 (n_prox=194885, n_syn = 14802), 1.5 (n_prox=212290, n_syn = 14878), 1.65 (n_prox=229167, n_syn = 14937), 1.8 (n_prox=245931, n_syn = 14977), 1.95 (n_prox=262711, n_syn = 15015), 2.1 (n_prox=279523, n_syn = 15050), 2.25 (n_prox=296300, n_syn = 15081), 2.4 (n_prox=312730, n_syn = 15104), 2.55 (n_prox=329411, n_syn = 15121), 2.7 (n_prox=346209, n_syn = 15134), 2.85 (n_prox=362732, n_syn = 15144), 3.0 (n_prox=379638, n_syn = 15162) Inh Onto Exc: 0.15 (n_prox=431892, n_syn = 73682), 0.3 (n_prox=811565, n_syn = 121599), 0.45 (n_prox=1170806, n_syn = 157540), 0.6 (n_prox=1479467, n_syn = 180273), 0.75 (n_prox=1757304, n_syn = 193944), 0.9 (n_prox=2016992, n_syn = 202263), 1.05 (n_prox=2266874, n_syn = 208146), 1.2 (n_prox=2509814, n_syn = 212482), 1.35 (n_prox=2747815, n_syn = 215775), 1.5 (n_prox=2980676, n_syn = 218563), 1.65 (n_prox=3210694, n_syn = 220807), 1.8 (n_prox=3439389, n_syn = 222649), 1.95 (n_prox=3666897, n_syn = 224084), 2.1 (n_prox=3893507, n_syn = 225230), 2.25 (n_prox=4119930, n_syn = 226138), 2.4 (n_prox=4346589, n_syn = 226835), 2.55 (n_prox=4572534, n_syn = 227462), 2.7 (n_prox=4800020, n_syn = 227975), 2.85 (n_prox=5028673, n_syn = 228358), 3.0 (n_prox=5257983, n_syn = 228664) Exc Onto Exc: 0.15 (n_prox=770871, n_syn = 8901), 0.3 (n_prox=1543245, n_syn = 16617), 0.45 (n_prox=2324856, n_syn = 23468), 0.6 (n_prox=3025906, n_syn = 28951), 0.75 (n_prox=3668135, n_syn = 34040), 0.9 (n_prox=4272087, n_syn = 38609), 1.05 (n_prox=4851252, n_syn = 42764), 1.2 (n_prox=5412367, n_syn = 46525), 1.35 (n_prox=5960708, n_syn = 49759), 1.5 (n_prox=6500095, n_syn = 52533), 1.65 (n_prox=7033215, n_syn = 54881), 1.8 (n_prox=7562021, n_syn = 56768), 1.95 (n_prox=8086547, n_syn = 58236), 2.1 (n_prox=8609337, n_syn = 59396), 2.25 (n_prox=9133249, n_syn = 60301), 2.4 (n_prox=9654903, n_syn = 60936), 2.55 (n_prox=10178601, n_syn = 61447), 2.7 (n_prox=10702652, n_syn = 61791), 2.85 (n_prox=11230246, n_syn = 62046), 3.0 (n_prox=11760681, n_syn = 62251)
6	6g	Conversion Rate (H01)		Inh Onto Exc: 0.15 (n_prox=5078, n_syn = 129), 0.3 (n_prox=9471, n_syn = 221), 0.45 (n_prox=13806, n_syn = 293), 0.6 (n_prox=17732, n_syn = 353), 0.75 (n_prox=21420, n_syn = 390), 0.9 (n_prox=24885, n_syn = 409), 1.05 (n_prox=28267, n_syn = 429), 1.2 (n_prox=31601, n_syn = 444), 1.35 (n_prox=34865, n_syn = 453), 1.5 (n_prox=38048, n_syn = 460), 1.65 (n_prox=41280, n_syn = 468), 1.8 (n_prox=44520, n_syn = 472), 1.95 (n_prox=47676, n_syn = 478), 2.1 (n_prox=50857, n_syn = 479), 2.25 (n_prox=54149, n_syn = 484), 2.4 (n_prox=57369, n_syn = 488), 2.55 (n_prox=60569, n_syn = 491), 2.7 (n_prox=63750, n_syn = 495), 2.85 (n_prox=67008, n_syn = 499), 3.0 (n_prox=70398, n_syn = 500) Inh Onto Inh: 0.15 (n_prox=720, n_syn = 58), 0.3 (n_prox=1576, n_syn = 116), 0.45 (n_prox=2511, n_syn = 165), 0.6 (n_prox=3417, n_syn = 204), 0.75 (n_prox=4249, n_syn = 226), 0.9 (n_prox=5054, n_syn = 237), 1.05 (n_prox=5857, n_syn = 244), 1.2 (n_prox=6615, n_syn = 256), 1.35 (n_prox=7410, n_syn = 263), 1.5 (n_prox=8243, n_syn = 264), 1.65 (n_prox=9015, n_syn = 266), 1.8 (n_prox=9818, n_syn = 267), 1.95 (n_prox=10556, n_syn = 269), 2.1 (n_prox=11321, n_syn = 271), 2.25 (n_prox=12125, n_syn = 271), 2.4 (n_prox=12912, n_syn = 272), 2.55 (n_prox=13675, n_syn = 273), 2.7 (n_prox=14458, n_syn = 273), 2.85 (n_prox=15234, n_syn = 273), 3.0 (n_prox=16078, n_syn = 273) Exc Onto Inh: 0.15 (n_prox=3234, n_syn = 938), 0.3 (n_prox=6774, n_syn = 1820), 0.45 (n_prox=10224, n_syn = 2403), 0.6 (n_prox=13301, n_syn = 2750), 0.75 (n_prox=16304, n_syn = 3015), 0.9 (n_prox=19103, n_syn = 3180), 1.05 (n_prox=21867, n_syn = 3299), 1.2 (n_prox=24485, n_syn = 3388), 1.35 (n_prox=27127, n_syn = 3449), 1.5 (n_prox=29705, n_syn = 3507), 1.65 (n_prox=32217, n_syn = 3539), 1.8 (n_prox=34790, n_syn = 3567), 1.95 (n_prox=37352, n_syn = 3591), 2.1 (n_prox=39874, n_syn = 3603), 2.25 (n_prox=42505, n_syn = 3618), 2.4 (n_prox=45124, n_syn = 3627), 2.55 (n_prox=47696, n_syn = 3632), 2.7 (n_prox=50393, n_syn = 3635), 2.85 (n_prox=53031, n_syn = 3638), 3.0 (n_prox=55636, n_syn = 3641) Exc Onto Exc: 0.15 (n_prox=19083, n_syn = 114), 0.3 (n_prox=35936, n_syn = 210), 0.45 (n_prox=52776, n_syn = 274), 0.6 (n_prox=67950, n_syn = 332), 0.75 (n_prox=82327, n_syn = 379), 0.9 (n_prox=96308, n_syn = 416), 1.05 (n_prox=109762, n_syn = 444), 1.2 (n_prox=123090, n_syn = 481), 1.35 (n_prox=136144, n_syn = 513), 1.5 (n_prox=148976, n_syn = 540), 1.65 (n_prox=161807, n_syn = 573), 1.8 (n_prox=174565, n_syn = 586), 1.95 (n_prox=187166, n_syn = 608), 2.1 (n_prox=199647, n_syn = 623), 2.25 (n_prox=212400, n_syn = 639), 2.4 (n_prox=224934, n_syn = 645), 2.55 (n_prox=237539, n_syn = 658), 2.7 (n_prox=250354, n_syn = 665), 2.85 (n_prox=263115, n_syn = 675), 3.0 (n_prox=276150, n_syn = 684)
6	6h	Conversion Rate vs. Axon Distance to Soma (MICrONS)		Exc Onto Exc: 37.02 (n_prox = 3626042), 86.29 (n_prox = 3154321), 110.11 (n_prox = 3124147), 133.45 (n_prox = 3101303), 159.94 (n_prox = 3031230), 192.1 (n_prox = 2898453), 232.13 (n_prox = 2632417), 282.1 (n_prox = 2295182), 347.59 (n_prox = 1994158), 480.32 (n_prox = 1860907) Exc Onto Inh: 39.23 (n_prox = 266045), 86.32 (n_prox = 266044), 110.14 (n_prox = 266045), 133.47 (n_prox = 266050), 160.03 (n_prox = 266046), 192.32 (n_prox = 266039), 232.61 (n_prox = 266050), 282.8 (n_prox = 266045), 348.3 (n_prox = 266048), 478.41 (n_prox = 256818)
6	6i	Conversion Rate vs. Axon Distance to Soma (H01)		Exc Onto Exc: 16.2 (n_prox = 56117), 52.59 (n_prox = 51054), 82.33 (n_prox = 48430), 105.72 (n_prox = 45839), 127.5 (n_prox = 45548), 150.26 (n_prox = 44634), 176.13 (n_prox = 42697), 208.56 (n_prox = 42372), 259.9 (n_prox = 40808), 406.54 (n_prox = 43756) Exc Onto Inh: 16.29 (n_prox = 9986), 53.1 (n_prox = 9985), 82.37 (n_prox = 9986), 105.74 (n_prox = 9985), 127.37 (n_prox = 9986), 150.25 (n_prox = 9986), 176.28 (n_prox = 9984), 208.69 (n_prox = 9986), 260.43 (n_prox = 9985), 400.13 (n_prox = 9987)
6	6j	Motif index		True Data Motif Counts: 1 (mean = 7377.00, std = None), 2 (mean = 10953501.00, std = None), 3 (mean = 3930648.00, std = None), 4 (mean = 57770467.00, std = None), 5 (mean = 390627.00, std = None), 6 (mean = 3350392.00, std = None), 7 (mean = 383004.00, std = None), 8 (mean = 15873.00, std = None), 9 (mean = 66407.00, std = None), 10 (mean = 16156.00, std = None), 11 (mean = 48173.00, std = None), 12 (mean = 9957.00, std = None), 13 (mean = 5003.00, std = None), 14 (mean = 243.00, std = None) Configuration Model Motif Counts: 1 (mean = 542.35, std = 22.87), 2 (mean = 14299374.71, std = 27655.61), 3 (mean = 4518749.52, std = 3704.19), 4 (mean = 59388828.02, std = 41381.50), 5 (mean = 166468.05, std = 1221.97), 6 (mean = 457613.39, std = 21497.53), 7 (mean = 54988.81, std = 2388.76), 8 (mean = 10084.60, std = 167.97), 9 (mean = 1126.76, std = 122.16), 10 (mean = 1601.45, std = 91.37), 11 (mean = 4437.23, std = 291.20), 12 (mean = 825.09, std = 54.10), 13 (mean = 113.36, std = 18.21), 14 (mean = 1.89, std = 1.40) Syn Shuffle Across Prox Edges Motif Counts: 1 (mean = 480.44, std = 22.40), 2 (mean = 6279304.00, std = 14980.70), 3 (mean = 3868010.65, std = 6644.99), 4 (mean = 9647512.96, std = 18046.76), 5 (mean = 42992.30, std = 248.85), 6 (mean = 38400.50, std = 2007.61), 7 (mean = 15842.36, std = 810.50), 8 (mean = 2393.19, std = 58.46), 9 (mean = 51.05, std = 9.23), 10 (mean = 85.18, std = 10.63), 11 (mean = 146.84, std = 14.67), 12 (mean = 73.36, std = 9.71), 13 (mean = 0.91, std = 0.89), 14 (mean = 0.00, std = 0.06) Random Syn Motif Counts: 1 (mean = 68.34, std = 8.16), 2 (mean = 5312021.94, std = 13718.80), 3 (mean = 2656056.72, std = 1670.41), 4 (mean = 10637406.97, std = 12420.92), 5 (mean = 6338.19, std = 86.42), 6 (mean = 6363.58, std = 765.06), 7 (mean = 1589.55, std = 193.60), 8 (mean = 528.37, std = 22.31), 9 (mean = 0.90, std = 0.91), 10 (mean = 1.87, std = 1.45), 11 (mean = 3.64, std = 2.12), 12 (mean = 0.87, std = 0.94), 13 (mean = 0.00, std = 0.06), 14 (mean = 0.00, std = 0.00) Proximities Motif Counts: 1 (mean = 970833.00, std = None), 2 (mean = 5902464653.00, std = None), 3 (mean = 3956073180.00, std = None), 4 (mean = 13851915701.00, std = None), 5 (mean = 2218025598.00, std = None), 6 (mean = 1999665781.00, std = None), 7 (mean = 534201392.00, std = None), 8 (mean = 20112550.00, std = None), 9 (mean = 75994000.00, std = None), 10 (mean = 83402913.00, std = None), 11 (mean = 461402658.00, std = None), 12 (mean = 167761891.00, std = None), 13 (mean = 92029299.00, std = None), 14 (mean = 13916043.00, std = None)
7	7d	Response Correlation vs Synapses Between Neurons		1 (n = 5350, err = 0.11797110607469749) 2 (n = 280, err = 0.13994874211810762) 3 (n = 34, err = 0.13324103791020708) 4+ (n = 11, err = 0.25867609439400974)
9	9a	Compartment Skeletal Length Validation (MICrONS)	158 Neurons	Axon (67.01 mm), Basal (245.11 mm), Apical Trunk (52.17 mm), Oblique (59.51 mm), Apical (43.95 mm), Apical Tuft (24.20 mm)'
9	9b	Compartment Skeletal Length Validation (H01)	89 Neurons	Axon (21.99 mm), Basal (131.60 mm), Apical Trunk (19.12 mm), Oblique (13.61 mm), Apical (34.29 mm), Apical Tuft (24.36 mm)
10	10b	Classification Percentage vs. Double Back Rule Threshold	1991	None
10	10bc	Classification Percentage vs. Width Jump Ratio	468	None

Figure	Panel	Title / Concept	N	Subclass N
20	20a	Spine Density	33653	23P (11694), 4P (5409), MC (682), Other Exc (13781), Other Inh (2087)
20	20b	Soma Volume	33823	4P (5445), 5P-ET (384), 5P-NP (388), BC (1012), BPC (693), Other Exc (24856), Other Inh (1045)
20	20c	Apical Trunk Starting Width	27577	4P (5525), 5P-ET (503), 5P-NP (363), Other Exc (21186)
20	20d	Synapses Onto AIS vs. Cell Type	25060	23P (8787), 4P (4011), 5P-IT (4839), 5P-NP (251), 5P-PT (738), 6P-CT (2320), 6P-IT (2233), BC (817), BPC (355), MC (490), NGC (219)
20	20e	Average Segment Length (Dendrite)	2768	BC (1045), BPC (697), MC (673), NGC (353)
20	20f	Soma Synapses	33655	23P (11976), 5P-ET (506), 5P-NP (388), BC (709), NGC (364), Other Exc (18339), Other Inh (1373)
20	20g	Axon Starting Width	39051	5P-ET (1273), 5P-NP (365), BPC (648), Other Exc (34664), Other Inh (2101)
21	21c	Synapses Onto AIS (Excitatory Cells)	26327	H01 (3208), MICrONS (23119)
21	21d	Synapses Onto Soma (Excitatory Cells)	26327	H01 (3208), MICrONS (23119)
21	21e	Depth Bin (MICrONS)	22955	0.08 (447), 0.16 (2646), 0.24 (3250), 0.31 (4023), 0.39 (3463), 0.46 (2048), 0.54 (2321), 0.61 (2749), 0.69 (1730), 0.77 (278)
21	21f	Depth Bin (H01)	3154	0.58 (385), 0.84 (464), 1.09 (355), 1.35 (528), 1.60 (529), 1.86 (426), 2.11 (224), 2.37 (160), 2.62 (83)
21	21g	Depth vs Synapses onto Soma (MICrONS)	22955	0.08 (447), 0.16 (2646), 0.24 (3250), 0.31 (4023), 0.39 (3463), 0.46 (2048), 0.54 (2321), 0.61 (2749), 0.69 (1730), 0.77 (278)
21	21h	Depth vs Synapses onto Soma (H01)	3154	0.58 (385), 0.84 (464), 1.09 (355), 1.35 (528), 1.60 (529), 1.86 (426), 2.11 (224), 2.37 (160), 2.62 (83)
22	22a	Number of Leaf Nodes Dendrite	2721	BPC (654), NGC (342), Other Inh (1725)
22	22b	# Leaf Nodes vs. Initial Width Basal/Dendrite Stems (MICrONS)	197860	None
22	22c	# Leaf Nodes vs. Initial Width Basal/Dendrite Stems (H01)	30058	None
23	23a	Spine Head Syn Size vs. Spine Head Volume (MICrONS)	240246	excitatory (222009), inhibitory (18237)
23	23b	Spine Head Syn Size vs. Spine Neck Width (MICrONS)	240246	excitatory (222009), inhibitory (18237)
23	23c	Spine Neck Syn Size vs. Spine Neck Width (MICrONS)	240246	excitatory (222009), inhibitory (18237)
23	23d	Spine Head Syn Size vs. Spine Neck Syn Size (MICrONS)	240246	excitatory (222009), inhibitory (18237)
23	23e	Spine Head Syn Size vs. Spine Head Volume (MICrONS)	240246	excitatory (222009), inhibitory (18237)
23	23f	Cell Type vs. Average Number of Syn on Spine (MICrONS)	233981	excitatory spines (n = 219476) inhibitory spines (n = 14505)
23	23g	Spine Head Syn Size vs. Spine Head Volume (H01)	73084	excitatory (71402), inhibitory (1682)
23	23h	Spine Head Syn Size vs. Spine Neck Width (H01)	73084	excitatory (71402), inhibitory (1682)
23	23i	Spine Neck Syn Size vs. Spine Neck Width (H01)	73084	excitatory (71402), inhibitory (1682)
23	23j	Spine Head Syn Size vs. Spine Neck Syn Size (H01)	73084	excitatory (71402), inhibitory (1682)
23	23k	Spine Head Syn Size vs. Spine Head Volume (H01)	73084	excitatory (71402), inhibitory (1682)
23	23l	Cell Type vs. Average Number of Syn on Spine (H01)	58678	excitatory spines (n = 57645) inhibitory spines (n = 1033)
24	24a	Syn Size vs Spine Head Volume (MICrONS)	159552	Exc Onto Exc (76295), Inh Onto Exc (83257)
24	24b	Syn Size vs Spine Head Volume (H01)	159312	Exc Onto Exc (76248), Inh Onto Exc (83064)
24	24c	Spine Head Volume (MICrONS)	152942	Exc Onto Exc (73980), Inh Onto Exc (78962)
24	24d	Spine Head Volume (H01)	63440	Exc Onto Exc (6855), Inh Onto Exc (56585)
24	24e	# of Syn on Spine Head (MICrONS)	426	Exc Onto Exc (346), Inh Onto Exc (80)
24	24f	# of Syn on Spine Head (H01)	425	Exc Onto Exc (345), Inh Onto Exc (80)
24	24g	Max Syn vs Syn Size Ratio on Multi Synapse Spine Head (MICrONS)	419	Exc Onto Exc (343), Inh Onto Exc (76)
24	24h	Max Syn vs Syn Size Ratio on Multi Synapse Spine Head (H01])	83	Exc Onto Exc (17), Inh Onto Exc (66)
24	24i	Synapse Size (MICrONS)	1011266	Exc Onto Exc (164046), Exc Onto Inh (113229), Inh Onto Exc (675088), Inh Onto Inh (58903)
24	24j	Synapse Size (H01)	7343	Exc Onto Exc (1055), Exc Onto Inh (4954), Inh Onto Exc (899), Inh Onto Inh (435)
26	26a	Conversion Rate vs. Depth Diff (MICrONS)		Exc Onto Exc: -93.5 (n_prox = 708773), -16.61 (n_prox = 1076784), 15.58 (n_prox = 2987066), 39.74 (n_prox = 4235702), 59.36 (n_prox = 3999294), 80.32 (n_prox = 3945396), 115.64 (n_prox = 3548662), 170.01 (n_prox = 2960345), 231.13 (n_prox = 2408377), 306.08 (n_prox = 1827653) Exc Onto Inh: -89.23 (n_prox = 39518), -16.13 (n_prox = 67956), 15.8 (n_prox = 210778), 39.94 (n_prox = 339614), 59.37 (n_prox = 340694), 80.38 (n_prox = 341392), 116.18 (n_prox = 339496), 171.15 (n_prox = 337738), 231.95 (n_prox = 332885), 306.83 (n_prox = 289258)
26	26b	Conversion Rate vs. Depth Diff (h01)		Exc Onto Exc: -103.97 (n_prox = 11034), -27.39 (n_prox = 14694), -4.95 (n_prox = 8450), 5.51 (n_prox = 16023), 14.89 (n_prox = 40616), 26.62 (n_prox = 43181), 38.69 (n_prox = 47070), 51.54 (n_prox = 53530), 67.14 (n_prox = 55479), 89.52 (n_prox = 57422), 126.91 (n_prox = 50020), 242.6 (n_prox = 59649) Exc Onto Inh: -104.09 (n_prox = 2286), -27.54 (n_prox = 3149), -4.91 (n_prox = 1748), 5.38 (n_prox = 3026), 14.85 (n_prox = 7583), 26.67 (n_prox = 8376), 38.77 (n_prox = 9764), 51.66 (n_prox = 11456), 67.1 (n_prox = 12138), 89.58 (n_prox = 12742), 127.41 (n_prox = 11994), 239.65 (n_prox = 14731)

Figure	Panel	Title / Concept	N	Subclass N
1269 28	28a	Motif Relative Frequency - MICrONS (All)		True Data Motif Counts: 1 (mean = 7377.00, std = None), 2 (mean = 10953501.00, std = None), 3 (mean = 3930648.00, std = None), 4 (mean = 57770467.00, std = None), 5 (mean = 390627.00, std = None), 6 (mean = 3350392.00, std = None), 7 (mean = 383004.00, std = None), 8 (mean = 15873.00, std = None), 9 (mean = 66407.00, std = None), 10 (mean = 16156.00, std = None), 11 (mean = 48173.00, std = None), 12 (mean = 9957.00, std = None), 13 (mean = 5003.00, std = None), 14 (mean = 243.00, std = None) Configuration Model Motif Counts: 1 (mean = 542.35, std = 22.87), 2 (mean = 14299374.71, std = 27655.61), 3 (mean = 4518749.52, std = 3704.19), 4 (mean = 59388828.02, std = 41381.50), 5 (mean = 166468.05, std = 1221.97), 6 (mean = 457613.39, std = 21497.53), 7 (mean = 54988.81, std = 2388.76), 8 (mean = 10084.60, std = 167.97), 9 (mean = 1126.76, std = 122.16), 10 (mean = 1601.45, std = 91.37), 11 (mean = 4437.23, std = 291.20), 12 (mean = 825.09, std = 54.10), 13 (mean = 113.36, std = 18.21), 14 (mean = 1.89, std = 1.40) Syn Shuffle Across Prox Edges Motif Counts: 1 (mean = 480.44, std = 22.40), 2 (mean = 6279304.00, std = 14980.70), 3 (mean = 3868010.65, std = 6644.99), 4 (mean = 9647512.96, std = 18046.76), 5 (mean = 42992.30, std = 248.85), 6 (mean = 38400.50, std = 2007.61), 7 (mean = 15842.36, std = 810.50), 8 (mean = 2393.19, std = 58.46), 9 (mean = 51.05, std = 9.23), 10 (mean = 85.18, std = 10.63), 11 (mean = 146.84, std = 14.67), 12 (mean = 73.36, std = 9.71), 13 (mean = 0.91, std = 0.89), 14 (mean = 0.00, std = 0.06) Random Syn Motif Counts: 1 (mean = 68.34, std = 8.16), 2 (mean = 5312021.94, std = 13718.80), 3 (mean = 2656056.72, std = 1670.41), 4 (mean = 10637406.97, std = 12420.92), 5 (mean = 6338.19, std = 86.42), 6 (mean = 6363.58, std = 765.06), 7 (mean = 1589.55, std = 193.60), 8 (mean = 528.37, std = 22.31), 9 (mean = 0.90, std = 0.91), 10 (mean = 1.87, std = 1.45), 11 (mean = 3.64, std = 2.12), 12 (mean = 0.87, std = 0.94), 13 (mean = 0.00, std = 0.06), 14 (mean = 0.00, std = 0.00) Proximities Motif Counts: 1 (mean = 970833.00, std = None), 2 (mean = 5902464653.00, std = None), 3 (mean = 3956073180.00, std = None), 4 (mean = 13851915701.00, std = None), 5 (mean = 2218025598.00, std = None), 6 (mean = 1999665781.00, std = None), 7 (mean = 534201392.00, std = None), 8 (mean = 20112550.00, std = None), 9 (mean = 75994000.00, std = None), 10 (mean = 83402913.00, std = None), 11 (mean = 461402658.00, std = None), 12 (mean = 167761891.00, std = None), 13 (mean = 92029299.00, std = None), 14 (mean = 13916043.00, std = None)
28	28b	Motif Relative Frequency - MICrONS (Inh Only)		True Data Motif Counts: 1 (mean = 620.00, std = None), 2 (mean = 153546.00, std = None), 3 (mean = 92977.00, std = None), 4 (mean = 248328.00, std = None), 5 (mean = 13517.00, std = None), 6 (mean = 24619.00, std = None), 7 (mean = 9023.00, std = None), 8 (mean = 686.00, std = None), 9 (mean = 621.00, std = None), 10 (mean = 769.00, std = None), 11 (mean = 1860.00, std = None), 12 (mean = 694.00, std = None), 13 (mean = 321.00, std = None), 14 (mean = 29.00, std = None) Configuration Model Motif Counts: 1 (mean = 39.06, std = 5.93), 2 (mean = 205483.82, std = 581.96), 3 (mean = 116115.15, std = 209.37), 4 (mean = 284227.47, std = 572.75), 5 (mean = 2069.34, std = 51.22), 6 (mean = 2105.25, std = 356.40), 7 (mean = 843.10, std = 135.95), 8 (mean = 218.74, std = 15.63), 9 (mean = 4.56, std = 2.35), 10 (mean = 8.75, std = 3.32), 11 (mean = 12.07, std = 4.11), 12 (mean = 4.85, std = 2.49), 13 (mean = 0.13, std = 0.34), 14 (mean = 0.00, std = 0.00) Syn Shuffle Across Prox Edges Motif Counts: 1 (mean = 173.79, std = 14.30), 2 (mean = 180258.18, std = 1674.63), 3 (mean = 90356.21, std = 609.90), 4 (mean = 223690.59, std = 1702.49), 5 (mean = 4441.91, std = 84.33), 6 (mean = 6021.68, std = 553.27), 7 (mean = 2684.63, std = 237.40), 8 (mean = 518.17, std = 26.97), 9 (mean = 43.93, std = 9.60), 10 (mean = 69.06, std = 9.87), 11 (mean = 83.00, std = 12.20), 12 (mean = 40.96, std = 6.96), 13 (mean = 3.09, std = 1.95), 14 (mean = 0.04, std = 0.19) Random Syn Motif Counts: 1 (mean = 21.80, std = 4.84), 2 (mean = 155168.95, std = 1720.36), 3 (mean = 77526.32, std = 268.59), 4 (mean = 312209.08, std = 1758.50), 5 (mean = 1112.85, std = 33.99), 6 (mean = 1129.44, std = 254.99), 7 (mean = 277.83, std = 65.05), 8 (mean = 91.41, std = 9.20), 9 (mean = 1.07, std = 1.12), 10 (mean = 2.00, std = 1.44), 11 (mean = 4.13, std = 2.44), 12 (mean = 1.00, std = 0.99), 13 (mean = 0.01, std = 0.11), 14 (mean = 0.00, std = 0.00) Proximities Motif Counts: 1 (mean = 35519.00, std = None), 2 (mean = 11523474.00, std = None), 3 (mean = 6692764.00, std = None), 4 (mean = 26319627.00, std = None), 5 (mean = 4298050.00, std = None), 6 (mean = 8808184.00, std = None), 7 (mean = 2352861.00, std = None), 8 (mean = 105187.00, std = None), 9 (mean = 625365.00, std = None), 10 (mean = 492474.00, std = None), 11 (mean = 2315658.00, std = None), 12 (mean = 553182.00, std = None), 13 (mean = 729469.00, std = None), 14 (mean = 196848.00, std = None)
28	28c	Motif Relative Frequency - MICrONS (Exc Only)		True Data Motif Counts: 1 (mean = 51.00, std = None), 2 (mean = 349641.00, std = None), 3 (mean = 374553.00, std = None), 4 (mean = 934337.00, std = None), 5 (mean = 3004.00, std = None), 6 (mean = 2158.00, std = None), 7 (mean = 531.00, std = None), 8 (mean = 55.00, std = None), 9 (mean = 3.00, std = None), 10 (mean = 5.00, std = None), 11 (mean = 15.00, std = None), 12 (mean = 6.00, std = None), 13 (mean = 0.00, std = None), 14 (mean = 0.00, std = None) Configuration Model Motif Counts: 1 (mean = 6.29, std = 2.49), 2 (mean = 354143.78, std = 192.71), 3 (mean = 375972.81, std = 249.71), 4 (mean = 933938.36, std = 970.55), 5 (mean = 542.67, std = 22.76), 6 (mean = 230.08, std = 124.94), 7 (mean = 89.97, std = 42.33), 8 (mean = 15.87, std = 3.63), 9 (mean = 0.02, std = 0.13), 10 (mean = 0.05, std = 0.23), 11 (mean = 0.14, std = 0.38), 12 (mean = 0.04, std = 0.22), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Syn Shuffle Across Prox Edges Motif Counts: 1 (mean = 23.94, std = 4.68), 2 (mean = 305884.63, std = 1550.10), 3 (mean = 210043.87, std = 892.90), 4 (mean = 467195.54, std = 1778.53), 5 (mean = 622.03, std = 25.46), 6 (mean = 408.27, std = 89.54), 7 (mean = 183.76, std = 41.90), 8 (mean = 20.45, std = 4.57), 9 (mean = 0.14, std = 0.36), 10 (mean = 0.24, std = 0.49), 11 (mean = 0.45, std = 0.67), 12 (mean = 0.24, std = 0.54), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Random Syn Motif Counts: 1 (mean = 3.92, std = 1.92), 2 (mean = 266889.39, std = 1546.80), 3 (mean = 133404.87, std = 397.34), 4 (mean = 534010.83, std = 1699.14), 5 (mean = 82.06, std = 9.40), 6 (mean = 86.25, std = 42.83), 7 (mean = 21.74, std = 11.83), 8 (mean = 6.75, std = 2.74), 9 (mean = 0.00, std = 0.06), 10 (mean = 0.00, std = 0.06), 11 (mean = 0.01, std = 0.12), 12 (mean = 0.01, std = 0.08), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Proximities Motif Counts: 1 (mean = 557816.00, std = None), 2 (mean = 3766065797.00, std = None), 3 (mean = 2705586217.00, std = None), 4 (mean = 8114160435.00, std = None), 5 (mean = 1469402757.00, std = None), 6 (mean = 821620414.00, std = None), 7 (mean = 262765795.00, std = None), 8 (mean = 6914301.00, std = None), 9 (mean = 21414903.00, std = None), 10 (mean = 27175777.00, std = None), 11 (mean = 222547569.00, std = None), 12 (mean = 100503502.00, std = None), 13 (mean = 32338897.00, std = None), 14 (mean = 4909292.00, std = None)

Figure	Panel	Title / Concept	N	Subclass N
1270				
28	28d	Motif Relative Frequency - H01 (All)		True Data Motif Counts: 1 (mean = 9.00, std = None), 2 (mean = 2641.00, std = None), 3 (mean = 9916.00, std = None), 4 (mean = 10139.00, std = None), 5 (mean = 65.00, std = None), 6 (mean = 79.00, std = None), 7 (mean = 49.00, std = None), 8 (mean = 0.00, std = None), 9 (mean = 0.00, std = None), 10 (mean = 0.00, std = None), 11 (mean = 2.00, std = None), 12 (mean = 0.00, std = None), 13 (mean = 0.00, std = None), 14 (mean = 0.00, std = None) Configuration Model Motif Counts: 1 (mean = 0.20, std = 0.44), 2 (mean = 2815.22, std = 17.66), 3 (mean = 9884.73, std = 46.55), 4 (mean = 10177.57, std = 33.03), 5 (mean = 13.26, std = 3.94), 6 (mean = 2.47, std = 5.97), 7 (mean = 1.62, std = 4.67), 8 (mean = 0.09, std = 0.30), 9 (mean = 0.00, std = 0.00), 10 (mean = 0.00, std = 0.00), 11 (mean = 0.01, std = 0.08), 12 (mean = 0.00, std = 0.06), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Syn Shuffle Across Prox Edges Motif Counts: 1 (mean = 2.87, std = 1.69), 2 (mean = 2986.72, std = 80.70), 3 (mean = 1669.05, std = 51.16), 4 (mean = 3435.57, std = 94.71), 5 (mean = 8.83, std = 3.27), 6 (mean = 8.34, std = 6.05), 7 (mean = 4.67, std = 3.63), 8 (mean = 0.62, std = 0.77), 9 (mean = 0.01, std = 0.08), 10 (mean = 0.02, std = 0.13), 11 (mean = 0.01, std = 0.12), 12 (mean = 0.01, std = 0.08), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Random Syn Motif Counts: 1 (mean = 0.11, std = 0.32), 2 (mean = 1833.82, std = 60.04), 3 (mean = 911.83, std = 30.86), 4 (mean = 3662.59, std = 78.88), 5 (mean = 0.33, std = 0.62), 6 (mean = 0.40, std = 1.28), 7 (mean = 0.08, std = 0.40), 8 (mean = 0.02, std = 0.13), 9 (mean = 0.00, std = 0.00), 10 (mean = 0.00, std = 0.00), 11 (mean = 0.00, std = 0.00), 12 (mean = 0.00, std = 0.00), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Proximities Motif Counts: 1 (mean = 21240.00, std = None), 2 (mean = 10969177.00, std = None), 3 (mean = 6071205.00, std = None), 4 (mean = 16743583.00, std = None), 5 (mean = 2913183.00, std = None), 6 (mean = 3002536.00, std = None), 7 (mean = 1278298.00, std = None), 8 (mean = 64293.00, std = None), 9 (mean = 126906.00, std = None), 10 (mean = 175963.00, std = None), 11 (mean = 621990.00, std = None), 12 (mean = 281840.00, std = None), 13 (mean = 154254.00, std = None), 14 (mean = 21604.00, std = None)
28	28e	Motif Relative Frequency - H01 (Inh Only)		True Data Motif Counts: 1 (mean = 1.00, std = None), 2 (mean = 27.00, std = None), 3 (mean = 40.00, std = None), 4 (mean = 163.00, std = None), 5 (mean = 1.00, std = None), 6 (mean = 4.00, std = None), 7 (mean = 3.00, std = None), 8 (mean = 0.00, std = None), 9 (mean = 0.00, std = None), 10 (mean = 0.00, std = None), 11 (mean = 0.00, std = None), 12 (mean = 0.00, std = None), 13 (mean = 0.00, std = None), 14 (mean = 0.00, std = None) Configuration Model Motif Counts: 1 (mean = 0.01, std = 0.08), 2 (mean = 36.15, std = 1.79), 3 (mean = 43.17, std = 1.26), 4 (mean = 166.47, std = 2.56), 5 (mean = 0.06, std = 0.25), 6 (mean = 0.02, std = 0.25), 7 (mean = 0.01, std = 0.16), 8 (mean = 0.00, std = 0.00), 9 (mean = 0.00, std = 0.00), 10 (mean = 0.00, std = 0.00), 11 (mean = 0.00, std = 0.00), 12 (mean = 0.00, std = 0.00), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Syn Shuffle Across Prox Edges Motif Counts: 1 (mean = 0.24, std = 0.46), 2 (mean = 26.57, std = 6.10), 3 (mean = 16.70, std = 4.37), 4 (mean = 36.09, std = 6.73), 5 (mean = 0.11, std = 0.34), 6 (mean = 0.15, std = 0.53), 7 (mean = 0.05, std = 0.25), 8 (mean = 0.01, std = 0.10), 9 (mean = 0.00, std = 0.00), 10 (mean = 0.00, std = 0.00), 11 (mean = 0.00, std = 0.00), 12 (mean = 0.00, std = 0.00), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Random Syn Motif Counts: 1 (mean = 0.00, std = 0.00), 2 (mean = 17.56, std = 4.43), 3 (mean = 8.63, std = 3.09), 4 (mean = 35.02, std = 6.35), 5 (mean = 0.00, std = 0.00), 6 (mean = 0.00, std = 0.00), 7 (mean = 0.00, std = 0.00), 8 (mean = 0.00, std = 0.00), 9 (mean = 0.00, std = 0.00), 10 (mean = 0.00, std = 0.00), 11 (mean = 0.00, std = 0.00), 12 (mean = 0.00, std = 0.00), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Proximities Motif Counts: 1 (mean = 857.00, std = None), 2 (mean = 49602.00, std = None), 3 (mean = 34848.00, std = None), 4 (mean = 94884.00, std = None), 5 (mean = 13094.00, std = None), 6 (mean = 16507.00, std = None), 7 (mean = 6447.00, std = None), 8 (mean = 315.00, std = None), 9 (mean = 743.00, std = None), 10 (mean = 967.00, std = None), 11 (mean = 3032.00, std = None), 12 (mean = 1171.00, std = None), 13 (mean = 795.00, std = None), 14 (mean = 115.00, std = None)
28	28f	Motif Relative Frequency - H01 (Exc Only)		True Data Motif Counts: 1 (mean = 0.00, std = None), 2 (mean = 125.00, std = None), 3 (mean = 127.00, std = None), 4 (mean = 316.00, std = None), 5 (mean = 0.00, std = None), 6 (mean = 0.00, std = None), 7 (mean = 0.00, std = None), 8 (mean = 0.00, std = None), 9 (mean = 0.00, std = None), 10 (mean = 0.00, std = None), 11 (mean = 0.00, std = None), 12 (mean = 0.00, std = None), 13 (mean = 0.00, std = None), 14 (mean = 0.00, std = None) Configuration Model Motif Counts: 1 (mean = 0.01, std = 0.08), 2 (mean = 124.43, std = 1.84), 3 (mean = 126.63, std = 0.84), 4 (mean = 315.37, std = 1.54), 5 (mean = 0.04, std = 0.21), 6 (mean = 0.04, std = 0.49), 7 (mean = 0.02, std = 0.29), 8 (mean = 0.00, std = 0.06), 9 (mean = 0.00, std = 0.00), 10 (mean = 0.00, std = 0.00), 11 (mean = 0.00, std = 0.00), 12 (mean = 0.00, std = 0.00), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Syn Shuffle Across Prox Edges Motif Counts: 1 (mean = 0.10, std = 0.31), 2 (mean = 104.83, std = 11.31), 3 (mean = 56.74, std = 8.09), 4 (mean = 119.75, std = 12.60), 5 (mean = 0.11, std = 0.33), 6 (mean = 0.05, std = 0.25), 7 (mean = 0.04, std = 0.21), 8 (mean = 0.00, std = 0.06), 9 (mean = 0.00, std = 0.00), 10 (mean = 0.00, std = 0.00), 11 (mean = 0.00, std = 0.00), 12 (mean = 0.00, std = 0.00), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Random Syn Motif Counts: 1 (mean = 0.00, std = 0.06), 2 (mean = 70.12, std = 9.48), 3 (mean = 35.57, std = 6.33), 4 (mean = 141.68, std = 12.69), 5 (mean = 0.00, std = 0.06), 6 (mean = 0.01, std = 0.12), 7 (mean = 0.00, std = 0.00), 8 (mean = 0.00, std = 0.00), 9 (mean = 0.00, std = 0.00), 10 (mean = 0.00, std = 0.00), 11 (mean = 0.00, std = 0.00), 12 (mean = 0.00, std = 0.00), 13 (mean = 0.00, std = 0.00), 14 (mean = 0.00, std = 0.00) Proximities Motif Counts: 1 (mean = 13917.00, std = None), 2 (mean = 6157073.00, std = None), 3 (mean = 3034464.00, std = None), 4 (mean = 9384467.00, std = None), 5 (mean = 1804679.00, std = None), 6 (mean = 1677490.00, std = None), 7 (mean = 669589.00, std = None), 8 (mean = 34993.00, std = None), 9 (mean = 70275.00, std = None), 10 (mean = 98293.00, std = None), 11 (mean = 378087.00, std = None), 12 (mean = 177686.00, std = None), 13 (mean = 93192.00, std = None), 14 (mean = 13418.00, std = None)

Appendix Table 3: Neuron Feature Documentation

Pipeline Stage	Feature Name	Resolution Available At	Access Point	Description (if needed)
2. Soma Identification	glia_meshes	neuron	pipeline_products	mesh fragments from large glia merge errors
2. Soma Identification	nuclei_meshes	neuron	pipeline_products	the meshes that deemed inside/touching/proximial to soma that were not part of the soma mesh and not part of a neurite
2. Soma Identification	soma_meshes	neuron	pipeline_products	separated meshes for all separate somas detected
3. Decomposition	limbs	neuron	attribute	a list of the limb objects in the neuron
3. Decomposition	specific limb	neuron	attribute	access a particular limb using the name or index of limb similar to dictionary key indexing (ex: "L4" or 4)
3. Decomposition	branches	limb	attribute	a list of the branch objects in the limb object
3. Decomposition	specific branch	limb	attribute	access a particular branch using the index of lthe branch
3. Decomposition	concept_network	neuron, limb	attribute	graph connectivity of limbs and soma (for neuron) or branches (for limbs)
3. Decomposition	concept_network_directed	limb	attribute	directed graph connectivity of branches on a limb where directionality is the skeletal walk direction away from the soma
3. Decomposition	mesh	neuron, limb, branch, soma	attribute	the triangular mesh representation of the object
3. Decomposition	[mean/max/sum/std]_node_[feature]	limb, neuron	custom function	can iterate through each limb and then each node and aggregate/summarize the value of an attribute
3. Decomposition	[parent/children/siblings]_[feature]_[max/min]	limb	limb_utils functions	custom functions that can get the relative feature for a parent node and children node on a limb
3. Decomposition	area	neuron, limb, branch, soma	attribute	area (um^2) of the respective mesh (neuron area does not include somas)
3. Decomposition	soma_area	neuron	attribute	area (um^2) of all the somas in a neuron (could be a multi-soma neuron)
3. Decomposition	bbox_[x/y/z]_[min/max](_soma_relative)	neuron	pipeline_products	the bounding box aligned with the xyz axies that surround the neuron (soma relative is a bounding box center using the soma center coordinate as the origin). The bounding box can be used for a quick estimation if proximities should be checked, etc.
3. Decomposition	best node feature match in: - siblings - children - ancestors - k-hop neighborhood	limb	custom function	Could iterate through the an relational nodes and compare to a feature/value/product of a certain node to find best match (example in limb_utils.best_feature_match_in_descendents)
3. Decomposition	node feature diff vs: - parent - sibling -child	limb	custom function	implement just the same as "best node feature match"
3. Decomposition	branch_length_[mean/median]	neuron	pipeline_products	mean/median branch skeletal length
3. Decomposition	endpoint_downstream	branch	attribute	3D coordinate of the branch downpoint skeleton endpoint
3. Decomposition	endpoint_upstream	branch	attribute	3D coordinate of the branch upstream skeleton endpoint
3. Decomposition	insignificant_limbs	neuron	attribute	list of mesh objects touching a soma mesh but was too small for decomposition (in order to avoid decompositng small mesh fragment merges)
3. Decomposition	n_multi_soma_touching_limbs	neuron	attribute	the number of limbs that are adjacent to more than 1 soma (products of merge errors)
3. Decomposition	multi_soma_touching_limbs	neuron	attribute	the indexes of the limbs that are adjacent to more than 1 soma (usually product of merge error)
3. Decomposition	n_same_soma_multi_touching_limbs	neuron	attribute	the number of limbs that are adjacent to at least 1 soma in multiple separate locations (products of merge errors)
3. Decomposition	touching_somas	limb	attribute	The soma identifiers that a current limb is adjacent two (useful for finding paths to cut for multi-soma or multi-touch limbs
3. Decomposition	n_somas	neuron	attribute	the number of separate somas currently in the neuron object (after multi-soma splitting this should always be 1)
3. Decomposition	n_branches	neuron, limb	attribute	the number of non-branching segments composing the neuron or object (aka, number of nodes in the graph)
3. Decomposition	n_branches_per_limb	neuron	attribute	list of number of branches for each limb where index of list matches the limb index
3. Decomposition	n_limbs	neruon	attribute	number of neurite projections
3. Decomposition	mesh_volume	neuron, limb, branch, soma	attribute	volume (um^3) of the respective mesh (neuron volume does not include somas)
3. Decomposition	mesh_volume_with_somas	neuron	attribute	volume (um^3) of neuron mesh includeing the somas
3. Decomposition	skeleton	neuron, limb, branch	attribute	Nx2x3 numpy array where each row (a 2x3 array) are two 3D coordinates that define one segment of the skeleton
3. Decomposition	skeletal_length	neuron, limb, branch	attribute	sum of skeletal distance (nm) of each non-branching segement (typically using 1000 nm approximate spacing between each skeletal point)
3. Decomposition	skeleton_bounding_box	custom function	mesh_tools.skeleton_utils	bounding box 3D corner coordinates (2x3) for entire skeleton. Useful for approximating mesh bounding box for applications like proximity calculations
3. Decomposition	skeletal_vector_[upstream/downstream]	branch	attribute	the skelelton vector near either the upstream coordinate or downstream coordinate where the vectors in both cases are oriented in the skeletal walk direction away from the soma
3. Decomposition	spines	neuron, limb, branch	attribute	list of meshes of the extracted spines
3. Decomposition	n_spines	neuron,limb.branch	attribute	number of spines on that entity
3. Decomposition	spine_density	neuron	attribute	number of spines / skeletal length (# / nm)
3. Decomposition	spine_volume_[density/median]	neuron	attribute	spine_volume / skeletal length (# / nm)
3. Decomposition	total_spine_volume	neuron, limb, branch	neuron_utils function	sum of all volume of spine meshes (um**3)
3. Decomposition	width_overall	branch	attribute	the mediabn radial width (nm) of the non-branching segment. The current width is generated by first removing any spines detected in order to avoid those protrusions from affecting value.
3. Decomposition	width_new	branch	attribute	dictionary storing the median radial width (nm) of the non-branching segment.for both spines included ("median_mesh_center") and spines not included "no_spine_median_mesh_center"
3. Decomposition	width_array	branch	attribute	dictionary storing the median radial width (nm) for 1000 nm segments allong the skeleton of the non-branching segment.for both spines included ("median_mesh_center") and spines not included "no_spine_median_mesh_center"
3. Decomposition	width_array_upstream_to_downstream	branch	attribute	same as width_array but arrays are returned for the 1000 nm skeletal segment points oriented from upstream to downstream
3. Decomposition	width_[upstream/downstream]	branch	attribute	approximiation of radial width (nm) near [upstream/downstream] skeleton point by averaging the 3 closest width measurements in width array (of "no_spine_median_mesh_center")
3. Decomposition	width(_no_spine)_[90_perc/median]	neuron	pipeline_products	width summary statistics for neuron
3. Decomposition	root_skeleton_vector_from_soma	limb	custom function	the vector from the soma mesh center to the root starting coordinate of the limb's protrusion from the soma
3. Decomposition	root_width	limb	limb_utils	the limb width at the start of the limb's protrusion from the soma
4. Multi-Soma Split	n_error_limbs	neuron	attribute	the number of limbs whose meshes were eliminated as a last resort to separate multi-soma merges because no reasonable split location could be determined
4. Multi-Soma Split	n_paths_cut	neuron	pipeline_products	the number of skeletal paths that were cut in order to eliminate all somas to soma paths
4. Multi-Soma Split	valid_error_split_points_by_limb	neuron	attribute	a dictionary data structure that stores for each limb - valid points: coordinates that should belong to the existing neuronal process (a marker of where the valid mesh is). - error points: coordinates that should belong to incorrect neuronal process resulting from merge errors (a marker of where the error mesh starts) - coordinate: locations of split points used in the elimination of soma to soma paths The valid and error points can be used as inputs for automatic mesh splitting algorithms in other pipelines (ex: Neuroglancer)
5. Cell Typing	axon_starting_coordinate	neuron	attribute	The upstream coordinate of the first branch labeled as axon (this may be different from the root coordinate of. the limb containing the axon because the axon can start downstream of the soma off a dendritic process)

Pipeline Stage	Feature Name	Resolution Available At	Access Point	Description (if needed)
1273 5. Cell Typing	spines_obj	neuron, limb, branch	attribute	list of synapse objects where each object stores the following information about the synapse: mesh_face_idx: a list of face indices of the branch that belong to the spine mesh mesh: the submesh of the branch that represent the spine (mesh_face_idx indexed into the branch mesh neck_face_idx: a list of face indices of the spine's mesh that were classified as the neck (can be empty if not detected) head_face_idx: list of face indices of the spine's mesh that were classified as the head (can be empty if not detected) neck_sdf: the sdf value of the neck submesh from the clustering algorithm used to segment the head from the neck head_sdf: the sdf value of the head submesh from the clustering algorithm used to segment the head from the neck head_width: a width approximation using ray tracing of the head submesh neck_width: a width approximation using ray tracing of the head submesh volume: volume of entire mesh spine_id: unique identifier for spine sdf: the sdf value of the spine submesh from the clustering algorithm used to segment the spine from the branch mesh endpoints_dist: skeletal walk distance of the skeletal point closest to the start of the spine protrusion to the branch skeletal endpoints upstream_dist: skeletal walk distance of the skeletal point closest to the start of the spine protrusion to the upstream branch skeletal endpoint downstream_dist: skeletal walk distance of the skeletal point closest to the start of the spine protrusion to the downstream branch skeletal endpoint coordinate_border_verts: coordinate: one coordinate of the border vertices to be used for spine locations bbox_oriented_side_lengths head_bbox_oriented_side_lengths neck_bbox_oriented_side_lengths head_mesh_splits head_mesh_splits_face_idx branch_width_overall branch_skeletal_length branch_width_at_base skeleton: surface skeleton over the spine mesh skeletal_length: length of spine skeleton # -- attributes similar to those of spine attribute closest_branch_face_idx closest_sk_coordinate: 3D location in space of closest skeletal point on branch for which spine is located closest_face_coordinate: center coordinate of closest mesh face on branch for which spine is located closest_face_dist: distance from synapse coordinate to closest_face_coordinate soma_distance: skeletal walk distance from synapse to soma soma_distance_euclidean: straight path distance from synapse to soma center compartment: the compartment of the branch that the spine is located on limb_idx: the limb identifier that the spine is located on branch_idx: the branch identifier that the spine is located on
5. Cell Typing	head_neck_shaft_idx	branch	attribute	an array the same size as number of faces with indices indicating whether that mesh face was classified as a spine head, spine neck or part of neurite shaft
5. Cell Typing	spine_density	neuron, limb, branch	attribute	number of spines / skeletal length (# / um) for the dendritic-like (above certain thickness) the branches longer than 5 um skeletal length near the soma (within 110 um). This number is used for the excitatory/inhibitory logistic regression classifier. Calculated near the soma to hopefully mitigate affects of merge errors that are more likely farther away from soma.
5. Cell Typing	spine_volume_density	neuron, limb, branch	attribute	sum spine volume (um**3) / skeletal length (um)
5. Cell Typing	synapses	neuron, limb, branch	attribute	list of synapse objects where each object stores the following information about the synapse: synapse_id: synapse volume upstream_dist: skeletal distance from the closest upstream branch point downstream_dist: skeletal distance from the closest downstream branch point or endpoint coordinate: 3D location in space: closest_sk_coordinate: 3D location in space of closest skeletal point on branch for which synapse is located closest_face_coordinate: center coordinate of closest mesh face on branch for which synapse is located closest_face_dist: distance from synapse coordinate to closest_face_coordinate soma_distance: skeletal walk distance from synapse to soma soma_distance_euclidean: straight path distance from synapse to soma center head_neck_shaft: whether the synapse is located on a spine head, spine neck or neurite shaft (decoding of integer label is in spine_utils) compartment: the compartment of the branch that the synapse is located on limb_idx: the limb identifier that the synapse is located on branch_idx: the branch identifier that the synapse is located on Note: features like head_neck_shaft, compartment are not populated until later stages (cell typing, autoproofreading) when that information is available for the branches
5. Cell Typing	synapses_df	neuron, limb, branch	custom function	summary dataframe of all properties of all synapses in object
5. Cell Typing	n_synapses_[head/neck/no_head/shaft/spine]	branch	custom function	synapses on a branch mapped to a mesh face with a certain spine label. The "spine" category refers to submeshes that were identified as a spine but could not be further split into a spine head or neck
5. Cell Typing	[axon/dendrite]_[mesh/skeleton/limb_branch_dict]	neuron	attribute	mesh, skeleton or limb branch information of all of the branches classified as axon or dendrite
5. Cell Typing	axon_angle_max	neuron	pipeline_products	the angle between the upstream skeleton vector and the vector pointing upwards in the volume (the typical direction apicals point). An axon pointing straight down would have a value close to 180
5. Cell Typing	[axon/dendrite]_bbox_[x/y/z]_[min/max](_soma_relative)	neuron	pipeline_products	the bounding box aligned with the xyz axies that surround the neuron's compartment (soma relative is a bounding box center using the soma center coordinate as the origin). The bounding box can be used for a quick estimation if proximities should be checked, etc.
5. Cell Typing	baylor_cell_type	neuron	pipeline_products	the excitatory/inhibitory classification using the Baylor College of Medicine tuned logistic regression model that only uses two cell type features: spine density and postsynaptic synapse density (information in cell_type_utils)
5. Cell Typing	baylor_cell_type_exc_probability	neuron	pipeline_products	probability output of E/I Baylor College of Medicine tuned logistic regression model
5. Cell Typing	boutons	neuron	attributes	list of meshes of boutons detected during axon identification
5. Cell Typing	n_syn_axon_ais_postsyn	neuron	pipeline_products	number of postsyaptic connections onto the axon intial segment (50 um downstream from the start of the axon)
5. Cell Typing	n_syn_axon_[ais/bouton/non_bouton]_[presyn/postsyn]	neuron	pipeline_products	number of synapses on the specified locations
5. Cell Typing	n_synapses_[pre/post]	neuron,limb,branch,soma	attribute	number of presynaptic/postsynaptic connections
5. Cell Typing	n_synapses_[somas/error]	neuron	attribute	Number of synapses on the soma or not attributed to the neuron (error), usually due to mutli-soma splitting at this stage
5. Cell Typing	neuron_graph_axon_labeled	neuron	pipeline_products	neuron object lite of the neuron after axon identification but before the axon branches are converted to a finer grain decomposition where smaller protrusions are separated (but before any automatic proofreading is applied)
5. Cell Typing	neuron_graph_high_fid_axon	neuron	pipeline_products	neuron object lite of the neuron after both axon identification and the axon branches are converted to a finer grain decomposition where smaller protrusions are separated (but before any automatic proofreading is applied)
5. Cell Typing	syn_density_[head/neck/shaft/no_head]	neuron	pipeline_products	number of postsynaptic connections with a specific spine label / skeletal length (# / um) for the dendritic-like (above certain thickness) the branches longer than 5 um skeletal length near the soma (within 110 um). This number is used for the excitatory/inhibitory logistic regression classifier. Calculated near the soma to hopefully mitigate affects of merge errors that are more likely farther away from soma.
5. Cell Typing	synapses_[pre/post]	neuron,limb,branch,soma	attribute	presynaptic/postsynaptic connections
5. Cell Typing	synapses_[somas/error]	neuron	attribute	soma or error synapses

Pipeline Stage	Feature Name	Resolution Available At	Access Point	Description (if needed)
1274 6. Auto Proofreading	compartment	branch	attribute	the compartment label for that branch. The possible compartments are axon: soma: apical_shaft (only excitatory): the first protrusion of the apical neurite from the soma (the shaft generally projects upward towards the top of the volume as a single lined entity apical (only excitatory): all branches downstream of the first apical shaft protrusion (includes apical_shaft,apical_tuft,oblique). This compartment includes offshoots of the apical shaft that were not close to a 90 degree protrusion angle with respect to the the apical shaft trajectory apical_tuft (only excitatory): the branches downstream of the point at which the apical shaft terminates its striaght upward trajectory and a branching point of 2 or more offshoots with a non straight upward trajectory oblique (only excitatory): offshoots of the apical shaft (before the apical_tuft section) that were close to a 90 degree protrusion angle with respect to the the apical shaft trajectory dendrite (only inhibitory): any non-axon neurites on an inhibitory cell basal (only excitatory): any non-apical neurites on an excitatory cell
6. Auto Proofreading	distance_errored_synapses(_[pre/post])	neuron	attribute	synapses whose closest mesh face was beyong a threshold distance andt thus was not considered
6. Auto Proofreading	mesh_errored_synapses(_[pre/post])	neuron	attribute	synapses mapped to mesh locations that were classified as mesh fragments from merge errors
6. Auto Proofreading	[compartment]_[limb_branch_dict]	neuron	attribute	the dictionary mapping of limbs and branches that were classified with that compartment label
6. Auto Proofreading	[compartment]_[mesh/skeleton]	neuron	custom function (neuron_utils)	can create the mesh or skeleton of a certain compartment using the limb branch dict attribute and neuron_utils module
6. Auto Proofreading	[comp]_skeletal_length	neuron	custom function (skeleton utils)	use skeleton_utils from [compartment]_[mesh/skeleton]
6. Auto Proofreading	[compartment]_bbox_[x/y/z]_[min/max](_soma_relative)	neuron	pipeline_products	the bounding box aligned with the xyz axies that surround the neuron's compartment (soma relative is a bounding box center using the soma center coordinate as the origin). The bounding box can be used for a quick estimation if proximities should be checked, etc.
6. Auto Proofreading	[compartment]_branch_length_[mean/median]	neuron	pipeline_productxs	mean/median branch length of all nodes with a given compartment label
6. Auto Proofreading	[compartment]_n_limbs	neuron	pipeline_products	number of limbs with at least one node with a given compartment label
6. Auto Proofreading	[compartment]_soma_angle_[max/min]	neuron	pipeline_products	the maximum of all the connected subgraphs of the compartment for the angle between the most upstream skeleton vector in the subgraph and the vector pointing upwards in the volume (the typical direction apicals point). An axon pointing straight down would have a value close to 180
6. Auto Proofreading	[error_filter]_error_[length/area]	neuron	pipeline_products	the area (in um ^ 2) of the faces / length of skeleton distance (in um) canceled out by filter (filters defined in proofreading_utils)
6. Auto Proofreading	[compartment]_root_skeletal_angle_[max/min]	neuron	custom function (neuron_utils)	the maximum or minimum angle between the top of volume vector and the root node (most upstream node) skeleton vector angle of all connected component subgrahps of that compartment label
6. Auto Proofreading	[compartment]_root_width_[max/min]	neuron	custom function (neuron_utils)	the maximum or minimum of the root node (most upstream node) of all connected component subgrahps of that compartment label
6. Auto Proofreading	n_[comp]_neurites	neuron	pipeline_products	The number of limbs with at least one branch with the given conmpartment
6. Auto Proofreading	n_syn_[compartment](_[head/neck/shaft/no_head)(_[postsyn/presyn])	neuron	pipeline_products	Number of synapses with any of the requirements for a specified compartment, spine classification (head/neck/shaft/no_head) or connection type (presyn/postsyn)
6. Auto Proofreading	syn_density_[head/neck/shaft/no_head]_after_proof	neuron	pipeline_products	number of postsynaptic connections with a specific spine label / skeletal length (# / um) for the branches longer than 5 um skeletal length near the soma (within 110 um). This number is used for the excitatory/inhibitory logistic regression classifier. Calculated near the soma to hopefully mitigate affects of merge errors that are more likely farther away from soma.
6. Auto Proofreading	spine_density_after_proof	neuron	pipeline_products	number of spines / skeletal length (# / um) for the dendritic-like (above certain thickness) branches longer than 5 um skeletal length near the soma (within 110 um). This number is used for the excitatory/inhibitory logistic regression classifier. Calculated near the soma to hopefully mitigate affects of merge errors that are more likely farther away from soma.
6. Auto Proofreading	split_locations_before_filter	neuron	pipeline_products	a nested dictionary datastructure storing the information on where the merge error filters were trigger. The order that the merge error filters was applied matters because the branches that triggered the filter are only those that had not triggered an early applied filter, and thus it was not already filtered away. Note: this product includes all branches that triggered the filter at this stage, regardless if they were downstream of another The datastructure is organized in the following way: merge error filter name: --> dict limb name: list of 2x3 arrays that stores coordinates of the endpoints the skeleton of the branch that triggered the merge error filter (1st coordinate is upstream branch)
6. Auto Proofreading	merge_valid_error_suggestions	neuron	attribute	a dictionary data structure that stores a list for each merge error filter of merge filter suggestions that would be the minimal cuts that would eliminate all merge errors of that type . Each filter detection is a dictionary storing meta data about the suggestion, with the following as some of the keys: - valid points: coordinates that should belong to the existing neuronal process (a marker of where the valid mesh is). - error points: coordinates that should belong to incorrect neuronal process resulting from merge errors (a marker of where the error mesh starts) - coordinate: locations of split points used in the elimination of soma to soma paths The valid and error points can be used as inputs for automatic mesh splitting algorithms in other pipelines (ex: Neuroglancer)
6. Auto Proofreading	neuron_graph_after_proof	neuron	pipeline_products	neuron object lite of the neuron after autoproofreading is applied (networkx graph object, so NEURD environment is not needed)