Point Cloud Representations of Neurons for Stitching Missing Serial Section

Jules Berman* JBERMAN@FLATIRONINSTITUTE.ORG
Jingpeng Wu JWUI@FLATIRONINSTITUTE.ORG
Dmitri B. Chklovskii DCHKLOVSKII@FLATIRONINSTITUTE.ORG
Center for Computational Neuroscience, Flatiron Institute. New York, NY 10010, U.S.A.

Editors: Under Review for MIDL 2022

Abstract

In the field of Connectomics, a primary problem is that of 3D neuron segmentation. Current methods produce imperfect segmentations which require manual proofreading to achieve high accuracy. Some methods have been explored for automating this proofreading process. To this end, we propose the idea of using point cloud representations of neurons for automated proofreading. We show that this representation allows us to efficiently learn and integrate information about neurons' morphology from the segmentation data alone. We do this by using point clouds to solve a common problem in segmentations, that of split neurons resulting from serial missing sections. We formulate this as a classification problem and train CurveNet, a state-of-the-art point cloud classification model, to identify which neurons should be merged. We show that our method not only performs strongly but scales reasonably to gaps well beyond what other methods have attempted to address. We believe this is a strong indicator of the viability of using point clouds for segmentation proofreading tasks.

Keywords: neuron segmentation, Connectomics, point cloud, proofreading, neuron stitching, deep geometric learning, CurveNet

1. Introduction

In Connectomics a core task is extracting neuron segmentation from 3D volumes of electron microscopy (EM) images (Plaza et al., 2014). The current state-of-the-art method for automating segmentation is based on 3D convolutional UNets (Lee et al., 2017). These produce good but imperfect segmentations, which need to be proofread by trained technicians who correct the errors of the automated methods. Some attempts have been made to provide ways to automate the proofreading process. These are often based around 3D convolutional networks as well (Zung et al., 2017) (Li et al., 2020). Recent work has shown that segmentations can be improved by representing them as graphs and then learning over these graph structures (Matejek et al., 2019). But to our knowledge, no work has been done which attempts to learn over segmentations represented as *point clouds*. To this end, we formulate a new method based on point cloud representations which can be used to solve a common problem in Connectomics pipelines, that of serial missing sections. We show that our method performs strongly and is vastly more data-efficient than equivalent formulations

* Contributed equally

© 2022 J. Berman, J. Wu & D.B. Chklovskii.

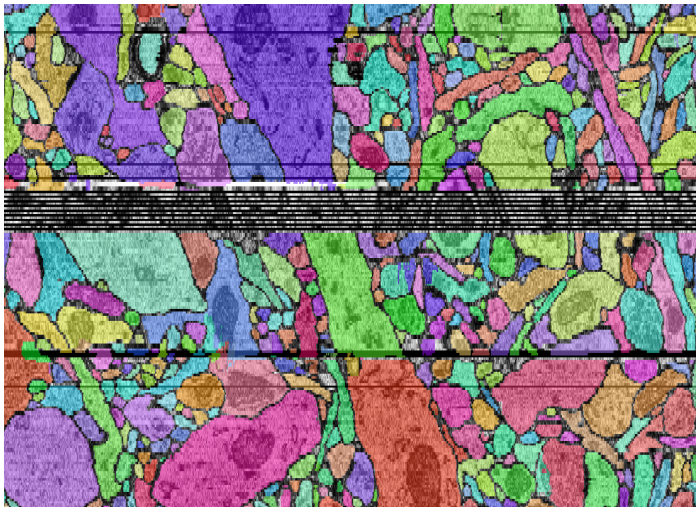

Figure 1: From human brain tissue: 3 serial missing sections then 9 alternating missing cause two major splits in the neuron segmentation (Shapson-Coe et al., 2021).

without point clouds. We believe our success here demonstrates the viability of point cloud representations in automating the proofreading process.

The problem of serial missing sections is a result of errors that occur in the imaging process. EM volumes are usually built by capturing parallel 2D cross-sectional images and stacking these into a 3D volume. But some slices are often rendered unusable due to blurring, noises, or some other error that loses 2D slices entirely. Moreover, these losses can happen over multiple consecutive slices (serial slices), creating large sections in the volumes where there is no data connecting neurons (Zheng et al., 2018) (Shapson-Coe et al., 2021) (Consortium et al., 2021). Researchers have attempted to make UNets robust to missing sections through data augmentations techniques (Lee et al., 2017). Here, sections of training data are deliberately replaced with zeros or some form of noise, while the target affinity map remains the same. This requires the UNet to predict the affinity map for the missing slices by interpolating from the surrounding context. While data augmentation has been shown to improve the UNet's ability to predict over missing sections, there are no explicit studies as to the degree to which this method can scale. Additionally, the success of this method is largely dependent on the sophistication of post-UNet processing pipeline. During this step, affinity maps are agglomerated into final segmentations using a variety of methods during which a number of hyperparameters must be tuned for optimal results. Any failures of this method can easily introduce merge errors which can be significantly more difficult for proofreaders to resolve than the split errors, which would occur by simply keeping the segmentations disjoint.

In this paper, we provide a novel method for merging disjoint neurons across large gaps in EM data. Our approach to solving this problem is based purely on the existing segmentations and does not utilize underlying EM data or intermediate affinity maps. Our hypothesis is that the segmentations alone capture the relevant morphological features of neurons necessary to accurately identify which neurons ought to be merged. We do not

represent neurons as dense labels within some larger 3D volume. Instead, we convert these volumetric representations into point cloud representations. Work in the area of deep geometric learning has postulated that non-euclidean representations of complex shapes can better capture the underlying geometric structure contained within data (Bronstein et al., 2017). By using point clouds here, we hope to efficiently represent the morphological features of neurons, using this as a basis to determine which neurons ought to be merged across the gap. To our knowledge, this is the first direct application of point cloud representations in segmentation for Connectomics. Thus, in this paper, we seek to provide a plausible method for merging neurons and show the viability of point cloud representations in the development of automated methods for improving segmentations.

2. Method

We formulate the problem of merging neurons accessing a gap as a binary classification task. Given two neurons within the volume, predict 1 to merge (i.e., assign both neurons the same label) or predict 0 to split (i.e. remain with different labels). For a single example, the direct output of the model is a two dimensional vector whose values are between 0 and 1. Each component can be interpreted as the probability of the label being 0 or 1. We then get a final prediction by thresholding the output by some number $t \in [0, 1]$. If the probability of a merge is greater than t then we predict 1, and we predict 0 otherwise. This thresholding is an important feature as it gives a user direct control over the trade-off between correct and false merges.

2.1. Data Preparation

For training data, we start with a ground truth segmentation and simulate missing sections from volume by simply zeroing out entire slices. In most cases, this is a perfect simulation of missing sections and allows us to generate accurate training data. For test data, we once again simulate missing sections from unseen ground truth in order to measure generalization.

In either case, once there are missing sections, the first step in preparing the data for classification consists in identifying pairs of neurons on either side of the gap to process. We begin by selecting some neuron that exists along a z-slice that borders the missing sections. We may refer to this as the top neuron. We then must select a group of neurons on the other side of the gap as candidates to either merge or remain split. We may refer to these as the bottom neurons. The process of selecting the candidate group is essential to the success of the algorithm. The larger the group of candidates, the more opportunities our model has to make a mistake in prediction. Conversely, the smaller the group of candidates, the higher likelihood that we will not even consider the correct neuron. Therefore it is necessary to construct a heuristic for selecting the group of candidates which is as restrictive as possible while retaining a high likelihood that the top neuron's correct partners are still within the batch. To this end, we examined a couple of possible heuristics and evaluated their performance based on the size of the candidate group they produce relative to how many positive examples are left out of the group. The best performing heuristic is based on the average Euclidean distance of each bottom neuron from the top neuron. G many neurons with the smallest average distance are selected as the candidate group for our algorithm,

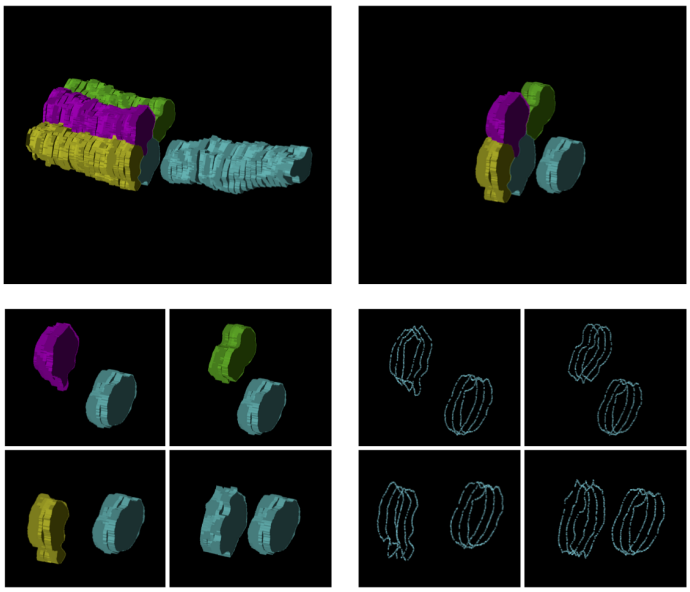

Figure 2: **Top Left**: Top neuron and bottom candidate group are selected.
Top Right: Neurons are truncated to number of context slices.
Bottom Left: The candidate group is separated into four separate examples.
Bottom Right: Volumes are converted to point cloud representations.

where G is a hyperparameter that may be selected. In our case, we found $\mathsf{G} = 4$ to be optimal.

Once the candidate group is selected, we have G many examples. Each example consists of the top neuron and one bottom neuron from the group. It is important to note here that we preserve the relative position of the top and bottom neurons within the entire volume. Each neuron may span many slices, up to the entire volume. But it is clear the most relevant information for merging neurons across a gap is the neuron's shape near that gap. So a choice must be made as to how much context to include. To control this, we introduce the hyperparameter of context slices (CS). This refers to the number of slices parallel to the gap we include to represent each neuron (top or bottom). We truncate each example according to the number of context slices. This means the resulting volume will have Z many slices where $\mathsf{Z} = 2 \times \mathsf{CS} + \mathsf{NS}$.

The next step is to transform the volumetric representation of each example into a point cloud representation. This is done by removing the interiors of each neuron then simply translating each voxel where a neuron exists to an (x, y, z) coordinate based on its relative position in the example. For each example, this will generate a different number of points based on the size of the neurons. To standardize the number of points, we uniformly sample $\mathsf{NP} = 2048$ points from each example. We sample with replacement in the case that the number of voxels is less than NP. Thus the resulting example is an array of shape $(\mathsf{NP} \times 3)$ with a label $y \in [0, 1]$. Lastly, the coordinates of each example are centered and normalized

so coordinates are in $[0, 1]$, but the relative size between examples is maintained. Unless otherwise noted, all our experiments are performed with the $\mathsf{G} = 4$, $\mathsf{NP} = 2048$, and $\mathsf{CS} = 3$.

2.2. Metrics

Within classification problems, there are several metrics used to evaluate the efficacy of a model. Some of the most common are precision, recall, and F1 score. These are useful for comparing models across a fixed dataset, but in our case, the number of negative and positive examples is directly affected by hyperparameters such as the size of the candidate group or the number of missing slices. So instead, we report two other metrics, that of **merge success rate** and **merge error rate**. We define merge error rate as the number of merge errors we create (i.e., False Positives) out of the total number of neurons we attempt to merge (i.e., the total number of top neurons). For each top neuron, it is possible to create arbitrarily many merge errors, so the merge error rate may exceed 1. We define merge success rate as the number of correct connections we make (i.e., True Positives) out of the total number of correct connections there are in the dataset (i.e., True Positives + False Negatives). This metric is equivalent to recall. It is worth noting that some neurons have greater than one correct connection across the gap, so the denominator is not equivalent to the number of top neurons.

We also report Variation of Information (VI) (Meilă, 2007). VI is a standard metric in Connectomics that is used to evaluate the overall quality of a segmentation in relation to its ground truth. In our experiments, we measure VI after the missing sections are dropped ($\mathsf{VI}_{\mathsf{pre}}$) and then again after we attempt to stitch neurons back together ($\mathsf{VI}_{\mathsf{post}}$). The final number we report is the percent reduction in VI which is simply given by: $\%\mathsf{Reduction} = (\mathsf{VI}_{\mathsf{pre}} - \mathsf{VI}_{\mathsf{post}})/\mathsf{VI}_{\mathsf{pre}}$. The VI generated by dropping slices, given by $\mathsf{VI}_{\mathsf{pre}}$, is dependent on where the gap occurs within the entire volume. The VI is largest when the gap occurs closest to the middle of the volume and decreases as you move towards the edges. Thus to account for this, we measure VI on a given test volume by dropping slices and applying our method at each possible index on the z-axis of the volume. We then average over the results of each iteration.

2.3. Model

We experimented with a variety of point cloud classification models. But ultimately, we found that CurveNet (Xiang et al., 2021), a recently developed model which is a top performer on classification over the ModelNet40 dataset, performed best across all our metrics. The model was training using a learning rate of $\epsilon = 0.001$ using a cross entropy loss and an AdamW optimizer with settings $\beta_1 = 0.9, \beta_2 = 0.999, \alpha = 0.05$.

3. Results

Our initial experiments were run using publicly available training data from the CREMI challenge. While this is technically ground truth data, in practice, there are many flaws and imperfections which make it a reasonable proxy of real-world segmentations. There are three volumes (A, B, C), each of which is of size (given in z, y, x) of $125 \times 1250 \times 1250$ with an anisotropic resolution of $40nm \times 4nm \times 4nm$. This is a relatively large resolution along

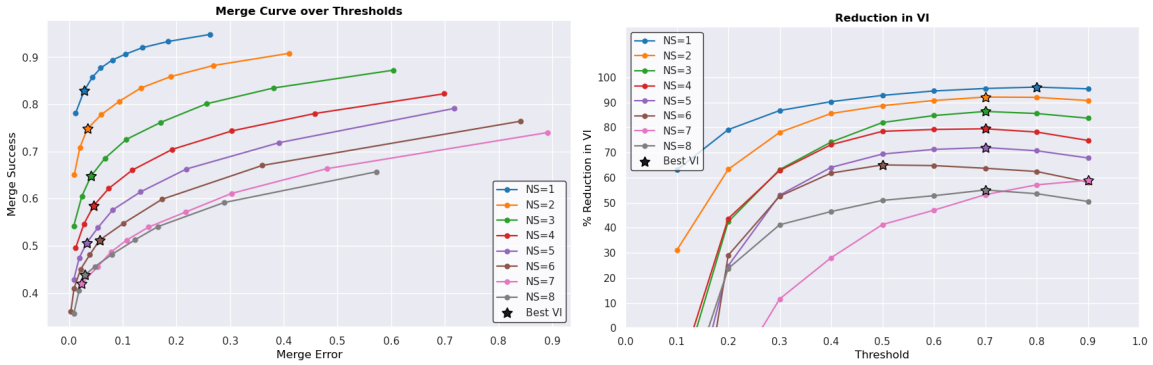

Figure 3: Performance of our method for 1-8 missing sections.
Left: plot of merge error rate vs merge success rate for different thresholds.
Right: plot of percent VI reduction vs threshold.

the z-axis, making this a particularly challenging data set for merging neurons across gaps from missing z-slices. From these volumes, we take 16 slices each for the test and validation datasets. Our main results concern how the method performs as we increase the number of missing sections (NS) from 1 to 8. We report these results in Figure 3. The first plot we show we refer to as the merge curve. This shows the merge error rate on the x-axis and the merge success rate on the y-axis. Each point is the performance at a given threshold, starting at 0.1 and going to 0.9. The stared points are the optimal thresholds for that model in terms of the best reduction in VI. As expected, we perform best for one missing slice, and performance decreases as more slices are removed. But there is still a meaningful amount of success even in the most difficult case. At 8 slices, we are able to merge a little above 40% of neurons while creating merge errors in less than 5% of cases. One interesting point to note is that for most runs, the optimal VI is achieved at a threshold of 0.7 or 0.8. These correspond to error rates of which are less than 5% . This suggests that, in terms of the VI metric, we should strongly prefer to avoid merge errors over increasing the possible number of merge successes. This is aligned with most proofreaders' preferences, as merge errors are much more difficult to fix than missed merge successes.

3.1. Parallel is Easy

A natural question is: how do the underlying arrangement of neurons affect relative to the EM images affect the performance of our method. By design, the CREMI volumes provide a perfect opportunity to investigate this question. In Volume A, most of the neurons run parallel to the z-axis. In Volume B and Volume C, the neurons run through the volume at many different angles, at times almost completely perpendicular to the z-axis. The effect of this difference is made clear in Figure 4. Here we see the two border slices adjacent to either side of the gap. The images show merge errors and merge successes for an example of 5 missing sections. Neurons colored green were merged with their partner correctly, red indicates there was some merge error, and blue indicates that no attempt to merge was made at all due to thresholding. The images make clear that we can successfully merge many

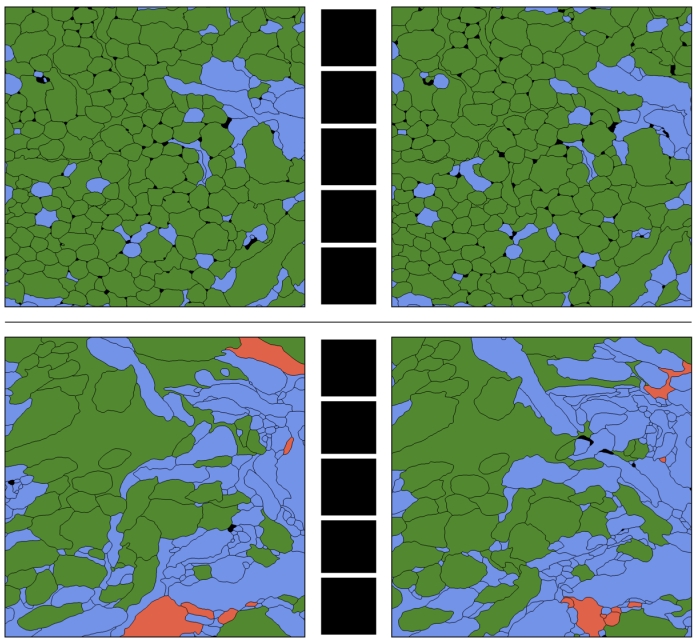

Figure 4: Performance of our method for an example of 5 missing sections. Green indicates merge success, red indicate merge error, blue indicates no merge attempted.
Top: Volume A where neurons run parallel to z-axis.
Bottom: Volume B where neurons run at many different angles.

more neurons in volume A relative to volume B. Additionally, in both volumes, the neurons that run parallel to the z-axis are successfully merged at a much higher rate than those that cross the z-axis. A possible reason for this stark difference is that when the neurons run parallel to the z-axis they have much less variation in terms of the relative displacement between slices and in terms of the variation in their cross-sectional shape along slices.

3.2. Context Slices

As mentioned before, an important hyperparameter for the method is that of context slices. This refers to the number of slices parallel to the missing sections used to represent the top and bottom neurons. To understand the influence of this choice in representation on the method's performance, we perform an ablation study on this parameter. The results are given in Figure 5. The clear trend is that it is optimal to have more than one context slice, but the choice of 2, 3, or 4 slices does not have a large impact. One interpretation of this result is that the single context slice does not allow the network to understand the direction that the neuron is "moving" along the z-axis. The addition of just one more slice allows the network to compute some form of a derivative that indicates whether the top neuron is moving towards or away from the bottom neuron.

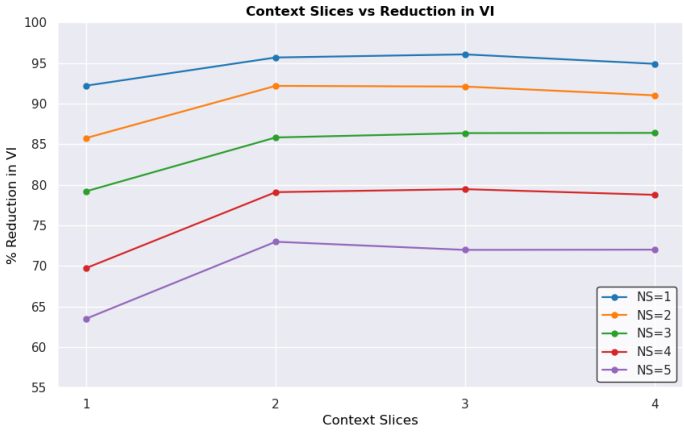

Figure 5: Ablation study on context slices. Figure shows how the optimal VI reduction changes as we vary the number of context slices in the neuron representation.

3.3. Data Efficiency

It is clear that point clouds representations are much more efficient in terms of data than volumetric representations of a segmentation neuron. This is because the volumetric representation of a neuron is data inefficient in two ways. The interior of the neuron must be represented, and the neuron must be padded into a rectangular volume. It is easy to see that with no information loss, one could represent a single neuron with an array of x, y, z coordinates for each voxel on the exterior surface of the neuron. In almost all cases, this will use significantly less data. We can also look at robustness to downsampling. That is, of the points on the exterior, how many are necessary to capture the relevant morphological features in order to successfully merge neurons.

We study this explicitly by varying the number of points with which we sample the volumetric representation of each example. The results are shown in Figure **??**. There is optimal performance at 2048 points. But the loss in VI decreases very slowly until 128 points, after which there is a steep drop off. With 3 spatial dimensions and 128 points, the representation uses 384 total floating point numbers. It is important to emphasize how small this is in comparison to volumetric representations. To represent an example as a volume with the same amount of data, one would have to use a volume of size roughly $7 \times 7 \times 8$. This is completely unfeasible no matter how one attempts to formulate the problem.

4. Conclusion

We have presented a novel method for stitching neurons across serial missing sections in EM volumes. We showed our method is a viable option for solving this problem across gaps of up to 8 at successive slices, more than any other method as attempted to address. Our approach was formulated around point cloud representations which were generated from the segmentation information alone. By limiting our data to only this information, we showed that our network solves this problem by learning the underlying structure of

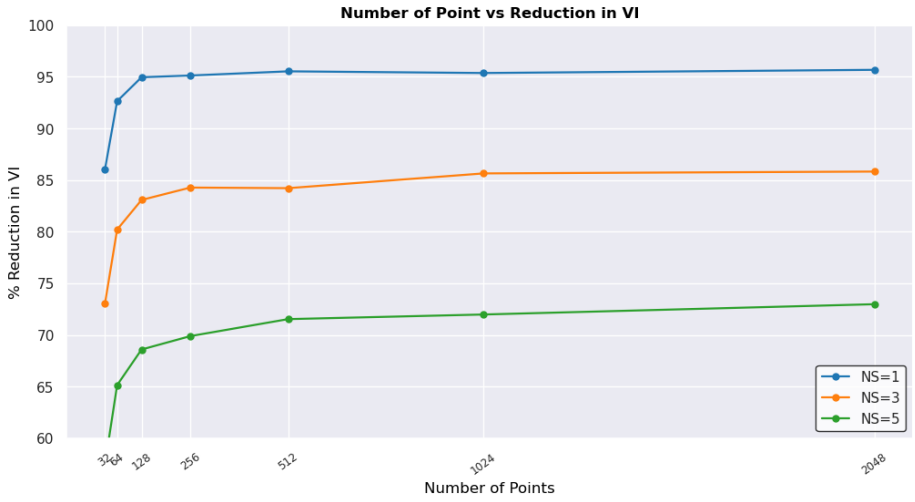

Figure 6: Figure shows how the optimal VI reduction changes as we vary the number point in the point cloud representation.

neuron morphology. Additionally, our success in using point clouds shows that it is a viable representation of segmentations for other automated proofreading tasks. We hope that this work not only provides researchers with another tool to improve neuron segmentations, but also is a first step in using geometric representations of Connectomics data.

Acknowledgments

We would like to thank the Simons Foundation for their generous sponsorship of this research.

References

Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, Jul 2017. ISSN 1558-0792. doi: 10.1109/msp.2017.2693418. URL http://dx.doi.org/10.1109/MSP.2017.2693418.

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. *bioRxiv*, 2021. doi: 10.1101/2021.07.28.454025. URL https://www.biorxiv.org/content/early/2021/08/09/2021.07.28.454025.

Kisuk Lee, Jonathan Zung, Peter Li, Viren Jain, and H. Sebastian Seung. Superhuman accuracy on the SNEMI3D connectomics challenge. *CoRR*, abs/1706.00120, 2017. URL http://arxiv.org/abs/1706.00120.

Hanyu Li, Michał Januszewski, Viren Jain, and Peter H. Li. Neuronal subcompartment classification and merge error correction. *bioRxiv*, 2020. doi: 10.1101/2020.04.16.043398. URL https://www.biorxiv.org/content/early/2020/07/16/2020.04.16.043398.

Brian Matejek, Daniel Haehn, Haidong Zhu, Donglai Wei, Toufiq Parag, and Hanspeter Pfister. Biologically-constrained graphs for global connectomics reconstruction. 07 2019. doi: 10.1109/CVPR.2019.00219.

Marina Meilă. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2006.11.013. URL https://www.sciencedirect.com/science/article/pii/S0047259X06002016.

Stephen M Plaza, Louis K Scheffer, and Dmitri B Chklovskii. Toward large-scale connectome reconstructions. *Current Opinion in Neurobiology*, 25:201–210, 2014. ISSN 0959-4388. doi: https://doi.org/10.1016/j.conb.2014.01.019. URL https://www.sciencedirect.com/science/article/pii/S095943881400035X. Theoretical and computational neuroscience.

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. *bioRxiv*, 2021. doi: 10.1101/2021.05.29.446289. URL https://www.biorxiv.org/content/early/2021/05/30/2021.05.29.446289.

Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 915–924, October 2021.

Zhihao Zheng, J. Scott Lauritzen, Eric Perlman, Camenzind G. Robinson, Matthew Nichols, Daniel Milkie, Omar Torrens, John Price, Corey B. Fisher, Nadiya Sharifi,

Steven A. Calle-Schuler, Lucia Kmecova, Iqbal J. Ali, Bill Karsh, Eric T. Trautman, John A. Bogovic, Philipp Hanslovsky, Gregory S.X.E. Jefferis, Michael Kazhdan, Khaled Khairy, Stephan Saalfeld, Richard D. Fetter, and Davi D. Bock. A complete electron microscopy volume of the brain of adult drosophila melanogaster. *Cell*, 174(3):730–743.e22, 2018. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2018.06.019. URL https://www.sciencedirect.com/science/article/pii/S0092867418307876.

Jonathan Zung, Ignacio Tartavull, Kisuk Lee, and H. Sebastian Seung. An error detection and correction framework for connectomics. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6821–6832, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.