ConnectomeVLM: Human-Level Connectomics Proofreading using Vision Language Models

Anonymous Authors[1]

Abstract

1. Introduction

Understanding complex systems requires mapping their underlying structure. In neuroscience, this principle is especially acute: to understand how brains produce behavior, disease, and cognition, we need ground-truth maps of neural connectivity. Connectomics, the systematic mapping of neural connectivity at the level of individual synapses, is poised to provide these maps. By imaging entire nervous systems at nanometer resolution using electron microscopy or expansion microscopy, researchers can trace how each neuron connects to others, creating complete wiring diagrams of the brain. These high-resolution maps promise to transform neuroscience by enabling simulation of nervous systems, identification of structural signatures of disease, and reverse-engineering of the computational principles underlying biological intelligence.

However, a critical bottleneck limits progress: while automated segmentation algorithms can partition nanometer-resolution brain images into individual neurons, they make systematic errors that require extensive manual proofreading. These errors fall into two categories (Figure 1): split errors, where a single neuron is incorrectly fragmented into multiple segments, and merge errors, where parts of distinct neurons are incorrectly fused together. The FlyWire Drosophila connectome, for instance, required substantial human effort to correct 139,255 neurons, and scaling this approach to mammalian brains with tens of millions of neurons threatens to make whole-brain reconstruction economically infeasible.

Recent work has shown that large-scale vision-language models (VLMs) can perform proofreading tasks with zero-shot prompting. ConnectomeBench [cite] demonstrated that frontier models like o4-mini and Claude Sonnet 4 achieve competitive performance on tasks including identifying segmentation errors, validating proposed corrections, and determining whether neuron fragments should be merged. Surprisingly, these models solved 3D spatial reasoning problems using only 2D orthogonal projections, mirroring how human proofreaders visually inspect neurons and suggesting that pre-trained visual representations contain sufficient structure to support connectomics workflows. However, these frontier models are large (¿100B parameters), proprietary, and expensive to deploy at scale. This raises fundamental questions about what computational resources are actually necessary: What function does language play versus pure vision? What role do pre-trained representations play versus task-specific fine-tuning? What is the contribution of model capacity versus training data scale? Are frontier-scale models strictly necessary, or can smaller, task-specialized models achieve comparable performance?

This paper provides a systematic scaling analysis to answer these questions. We compare three model classes across four proofreading subtasks that span a range of reasoning complexity: (1) vision-only models (CNNs and ViTs), (2) vision-language models (VLMs) without generation (linear probes on SigLip), and (3) generative VLMs (fine-tuned with LoRA). We evaluate along four deployment-critical axes: performance (data requirements to achieve human-level accuracy), generalization (zero-shot transfer to new species and imaging modalities), calibration (reliability of uncertainty estimates for human-AI collaboration), and interpretability (whether models learn meaningful heuristics). This analysis reveals task-specific scaling laws that inform the design of economically viable automated proofreading systems.

Our systematic evaluation reveals task-specific scaling laws: simple topology-matching tasks achieve human-level performance with vision-only models trained on fewer than 1,000 samples, while complex reasoning tasks benefit from larger vision-language models but still achieve strong performance with task-specific fine-tuning. We demonstrate robust cross-species generalization, with models trained on mouse cortex transferring to Drosophila, zebrafish, and human datasets with modest accuracy degradation. We show

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

that models can be well-calibrated and that they learn biologically meaningful visual heuristics rather than exploiting spurious shortcuts.

The remainder of the paper is organized as follows. Section 2 describes our training methods and data generation pipeline. Section 3 presents results organized by evaluation axis: performance and scaling behavior, cross-species generalization, calibration analysis, cost estimation, and interpretability findings.

2. Background

2.1. The Need for Connectomic Proofreading

Improvements in automated connectomic tracing have historically come from improvements in segmentation and agglomeration (Januszewski et al.; Sheridan et al.) [CITE AGGLOMERATION PAPERS HERE]. However, the sheer scale of scanned nervous systems leads to accumulating errors even with highly accurate segmentation algorithms, and the complex interconnectedness of neurons makes each error compound. This has necessitated substantive concerted manual human proofreading efforts for recent large-scale connectomic efforts (FlyWire Consortium, 2024; MICrONS Explorer, 2025), and limits the scale and accessibility of such endeavors.

2.2. Attempts to automate proofreading

For this reason, various works have investigated the potential of machine learning approaches for proofreading, such as via classic computer-vision models ([CITE] CNN ResNet), more advanced graph-based approaches, and Transformer-based vision-language models (Brown et al., 2025).

Earlier work applied CNN classifiers to perform proofreading-related tasks such as scoring neuron boundaries, predicting error likelihood maps, or classifying neuronal compartments, in mouse EM data. Haehn et al. (2017); Zung et al. (2017); Li et al. (2020)

More recent approaches have utilized graph- and morphology-based heuristics to propose split corrections, in mouse and human EM data (Celii et al., 2025; Joyce et al., 2023). Similarly, Schmidt et al. (2024) employed a CNN-based tracing agent that 'flies' along neurites to trace them and make split and merge corrections; Januszewski et al. (2025) utilize U-Nets to agglomerate fragments based on evaluated shape plausibility.

Finally, Brown et al. (2025) demonstrate neuronal segment classification, merge error identification, and split error correction capabilities in vanilla frontier vision-language models.

But, while a large space of impressive approaches exists for various datasets and sub-problems within the automatic proofreading space, we know of no systems for automatic post-agglomerative proofreading that can comprehensively identify and correct split and merge errors.

2.3. High Specificity of Current Approaches

Humans can be easily trained to proofread and can use their biological knowledge and generalized visuospatial reasoning skills to identify and correct split and merge errors by inspecting 2D projections of neuron meshes and tissue images, in interfaces such as NeuroGlancer (Google Inc., 2016). Furthermore, they generalize easily across imaging modalities (SEM, TEM, ExM), and species, with little to no re-training needed [CITATION NEEDED].

In contrast, current state-of-the-art automated proofreading approaches mostly use error-type specific data representation, models and heuristics, and have mostly not been shown to generalize across modalities and species:

Older CNN-based approaches have focused on specific, hand-crafted proofreading sub-tasks, and their generalization across modalities and species has not been systematically demonstrated.

'Smart agglomerative' approaches like Pathfinder shows impressive results in reducing the overall occurrence of errors during agglomeration, reducing proofreading burden, but cannot proofread remaining errors in and of itself, and its generalization to other data modalities has not been evaluated.

NEURD has been shown to generalize across species and between SEM and TEM, but is specific to merge errors and primarily targets neuron fragments containing cellular somas [DOUBLECHECK THIS]. RoboEM can correct split and merge errors, but does not identify error candidates, and while it has been trained and evaluated on mouse and human data, its ability to generalize across modalities and species without finetuning has not been characterized.

Brown et al. (2025) show vanilla frontier vision-language models (VLMs) generalizing segment identification, merge error detection, and split error correction on mouse and fly EM data, but do not tackle split error detection or merge error correction.

2.4. Toward Unified Systems

The fact that vanilla VLMs show meaningful ... [segue into why we're exploring this more systematically now].

3. Methods

3.1. Proofreading Tasks

Connectomics segmentation errors fall into two categories: split errors, where a single neuron is fragmented into multiple segments, and merge errors, where parts of multiple neurons are incorrectly fused. The proofreading workflow decomposes into three stages: (1) error candidate localization (identifying potential error sites using skeleton-based heuristics), (2) error identification (determining whether a candidate represents a true error), and (3) error correction (validating proposed fixes). Our skeleton-based heuristics achieve approximately 95% recall for split errors and 60% recall for merge errors, though at only 2% precision, necessitating accurate identification models.

We focus on identification and correction tasks across a difficulty spectrum:

Split Error Correction (Low complexity): Given two segment endpoints, determine whether they should be merged. Requires local topology matching by evaluating whether segments exhibit geometric continuity and consistent morphology at the junction point. Success depends on recognizing alignment across multiple 2D views and matching branch patterns.

Split Error Identification (Medium complexity): Given a segment endpoint, determine whether it represents a true split error requiring correction. Requires multimodal reasoning, fusing information from both 3D segmentation renderings and raw electron microscopy (EM) or expansion microscopy (ExM) images. The model must detect whether the endpoint corresponds to a true neuronal terminus or an artificial break introduced by segmentation errors.

Merge Error Correction Tasks (High complexity):

- Merge Error Identification: Given a segment, determine whether it contains a merge error where parts of multiple distinct neurons have been incorrectly fused. Requires global visual reasoning over the entire segment morphology to detect discontinuities, inconsistent branching patterns, or regions where disparate neuronal structures have been incorrectly joined.

- Split Action Evaluation: Given a proposed split correction, determine whether it successfully separates merged neurons. Requires counterfactual visual reasoning. The model must evaluate whether the split-off component genuinely belongs to a different neuron, often by assessing whether morphological patterns diverge across the proposed boundary.

3.2. Data Sources

Five datasets were used as a source of training and testing data for this project, as shown in Table 1. The MICrONS project's proofread, EM-based mouse connectome was used for training, and EM/ExM datasets across four species were used for evaluation generalization to other species and imaging modalities.

Species	Data Modality	Reference
Mouse	EM	The MICrONS Consortium (2025)
Fly	EM	FlyWire Consortium (2024)
Human	EM	Shapson-Coe et al. (2024)
Zebrafish	EM	Petkova et al. (2025)
Mouse	ExM	Tavakoli et al. (2025)

Table 1. Datasets used in this work. EM = Electron Microscopy, ExM = Expansion Microscopy

All EM-based datasets exposed raw images, meshes, skeletons, supervoxels, L2 nodes, roots, and detailed edit history, which we accessed via CAVEClient (Dorkenwald et al., 2025), ChunkedGraph (CAVEconnectome contributors, 2026) and CloudVolume (Silversmith, 2021a; seung-lab contributors, 2026a).

The ExM-based dataset provided raw image, agglomerated and proofread segmentation and meshes via CloudVolume. Skeletons were manually generated via kimimaro (Silversmith, 2021b; seung-lab contributors, 2026b), which uses a variant of the TEASAR algorithm (Sato et al., 2000); Edit history, and true and false corrections were inferred from the difference between the segmentations.

3.3. Data Generation

We generate samples for supervised training on four binary choice tasks: True and false examples across 1) split error identification, 2) merge error identification, 3) merge correction evaluation, and 4) split correction evaluation.

Junctions and endpoints of the neurons' skeletons without errors are used as negative examples of errors, actual error sites as yes-examples. For evaluation of merge corrections, actual merges from the edit history are used as yes-examples, merges of random adjacent roots at endpoints and actual split error locations as no-examples. The continuous nature of split corrections makes it more difficult to derive clear yes- and no-examples directly from the edit graph, so that we sampled sink and source points and manually computed and evaluated resulting splits.

For each task, we present the model with three views of the visual scene, each from a different orthographic projection, to provide 3D context. For the split error identification problem, we also incorporate 3D slices of the EM or ExM data for additional context. To prevent contamination between

training splits, we ensure that the same segments and locations do not appear in different splits. A full breakdown of the data generation process can be found in Appendix A.

4. Results

4.1. Performance Scaling Laws

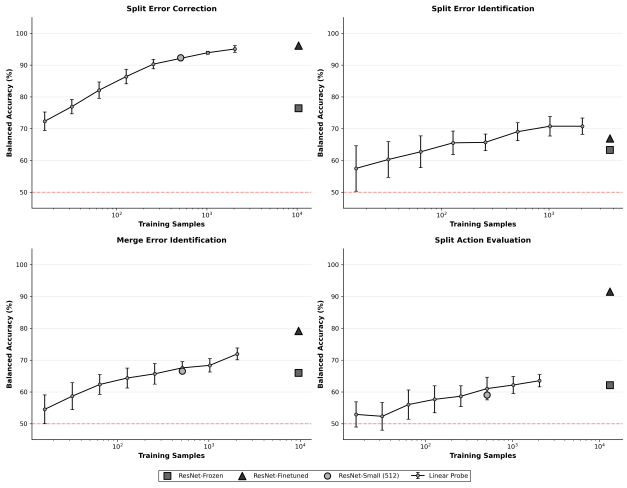

Figure 1. Caption

4.2. Generalization

4.3. Calibration

4.4. Computational Cost Estimation for Automated Proofreading

Connectome proofreading represents a major practical bottleneck in connectomics. FlyWire required 30 human-years to fully proofread the *Drosophila* brain's 139,255 neurons; such timescales prohibit rapid iteration on segmentation algorithms and limit deployment to new datasets. To enable efficient AI-based proofreading at scale, we must understand the computational cost landscape. We analyzed edit histories from two major connectomics datasets: the MICrONS mouse cortex (2,314 proofreading-accessible neurons in a 1 mm^3 volume, representing a partially-proofread mammalian circuit), and FlyWire's *Drosophila* brain (139,255 neurons, fully proofread, representing a large-scale completed dataset).

4.5. Interpretation

5. Discussion

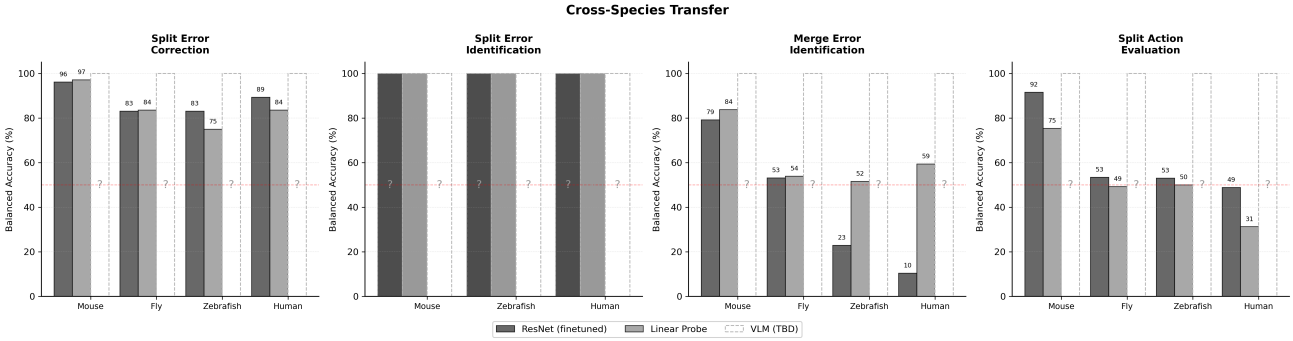

Figure 2. Caption

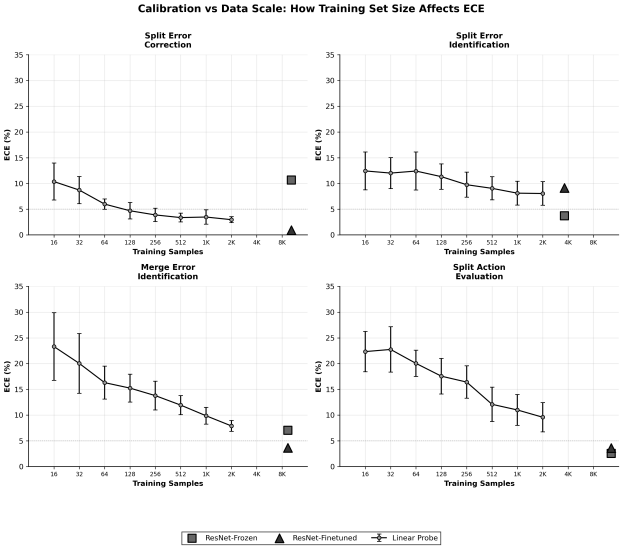

Figure 3. Caption

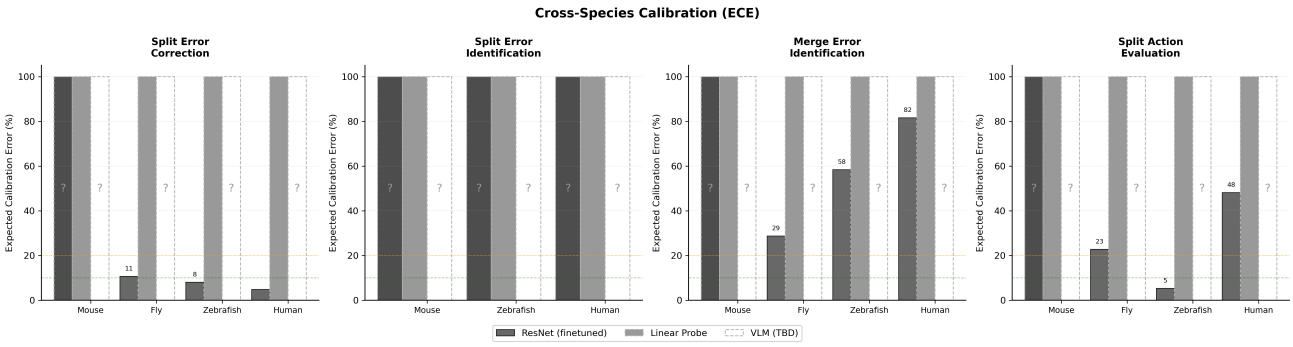

Figure 4. Caption

Impact Statement

Authors are **required** to include a statement of the potential broader impact of their work, including its ethical aspects and future societal consequences. This statement should be in an unnumbered section at the end of the paper (co-located with Acknowledgements – the two may appear in either order, but both must be before References), and does not count toward the paper page limit. In many cases, where the ethical impacts and expected societal implications are those that are well established when advancing the field of Machine Learning, substantial discussion is not required, and a simple statement such as the following will suffice:

"This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here."

The above statement can be used verbatim in such cases, but we encourage authors to think about whether there is content which does warrant further discussion, as this statement will be apparent if the paper is later flagged for ethics review.

References

Brown, J., Kirjner, A., Vivekananthan, A., Boyden, E., et al. Connectomebench: Can LLMs proofread the connectome? arXiv preprint, 2025. URL https://arxiv.org/pdf/2511.05542.

CAVEconnectome contributors. PyChunkedGraph: proofreading and segmentation data management backend, 2026. URL https://github.com/CAVEconnectome/PyChunkedGraph.

Celii, B., Papadopoulos, S., Ding, Z., Fahey, P. G., Wang, E., Papadopoulos, C., Kunin, A. B., Patel, S., Bae, J. A., Bodor, A. L., Brittain, D., Buchanan, J., Bumbarger, D. J., Castro, M. A., Cobos, E., Dorkenwald, S., Elabbady, L., Halageri, A., Jia, Z., Jordan, C., Kapner, D., Kemnitz, N., Kinn, S., Lee, K., Li, K., Lu, R., Macrina, T., Mahalingam, G., Mitchell, E., Mondal, S. S., Mu, S., Nehoran, B., Popovych, S., Schneider-Mizell, C. M., Silversmith, W., Takeno, M., Torres, R., Turner, N. L., Wong, W., Wu, J., Yu, S.-C., Yin, W., Xenes, D., Kitchell, L. M., Rivlin, P. K., Rose, V. A., Bishop, C. A., Wester, B., Froudarakis, E., Walker, E. Y., Sinz, F., Seung, H. S., Collman, F., da Costa, N. M., Reid, R. C., Pitkow, X., Tolias, A. S., and Reimer, J. NEURD offers automated proofreading and feature extraction for connectomics. *Nature*, 640(8058):487–496, April 2025.

Dorkenwald, S., Schneider-Mizell, C. M., Brittain, D., Halageri, A., Jordan, C., Kemnitz, N., Castro, M. A., Silversmith, W., Maitin-Shephard, J., Troidl, J., Pfister, H., Gillet, V., Xenes, D., Bae, J. A., Bodor, A. L., Buchanan, J., Bumbarger, D. J., Elabbady, L., Jia, Z., Kapner, D., Kinn, S., Lee, K., Li, K., Lu, R., Macrina, T., Mahalingam, G., Mitchell, E., Mondal, S. S., Mu, S., Nehoran, B., Popovych, S., Takeno, M., Torres, R., Turner, N. L., Wong, W., Wu, J., Yin, W., Yu, S.-C., Reid, R. C., da Costa, N. M., Seung, H. S., and Collman, F. CAVE: Connectome annotation versioning engine. *Nature Methods*, pp. 1–9, April 2025.

FlyWire Consortium. Neuronal wiring diagram of an adult brain. 634(8032):124–138, 2024. doi: 10.1038/s41586-024-07558-y. URL https://www.nature.com/articles/s41586-024-07558-y.

Google Inc. Neuroglancer: Webgl-based viewer for volumetric data. 2016. URL https://github.com/google/neuroglancer.

Haehn, D., Kaynig, V., Tompkin, J., Lichtman, J. W., and Pfister, H. Guided proofreading of automatic segmentations for connectomics. *arXiv*, April 2017.

Januszewski, M., Kornfeld, J., Li, P. H., Pope, A., Blakely, T., Lindsey, L., Maitin-Shepard, J., Tyka, M., Denk, W., and Jain, V. High-precision automated reconstruction of neurons with flood-filling networks. 15(8):605–610. ISSN 1548-7105. doi: 10.1038/s41592-018-0049-4.

Januszewski, M. et al. Accelerating neuron reconstruction with PATHFINDER. bioRxiv preprint, 2025. bioRxiv:2025.05.16.654254.

Joyce, J., Chalavadi, R., Chan, J., Tanna, S., Xenes, D., Kuo, N., Rose, V., Matelsky, J., Kitchell, L., Bishop, C., Rivlin, P. K., Villafañe-Delgado, M., and Wester, B. A novel semi-automated proofreading and mesh error detection pipeline for neuron extension. *bioRxiv*, October 2023.

Li, H., Januszewski, M., Jain, V., and Li, P. H. Neuronal subcompartment classification and merge error correction. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, pp. 88–98, 2020.

MICrONS Explorer. MICrONS Explorer dataset portal: Cortical mm3 (minnie65_public), 2025. URL https://www.microns-explorer.org/cortical-mm3.

Petkova, M. D., Januszewski, M., Blakely, T., Herrera, K. J., Schuhknecht, G. F. P., et al. A connectomic resource for neural cataloguing and circuit dissection of the larval zebrafish brain. 2025. doi: 10.1101/2025.06.10.658982. URL https://doi.org/10.1101/2025.06.10.658982. Preprint.

Sato, M., Bitter, I., Bender, M. A., Kaufman, A. E., and Nakajima, M. TEASAR: Tree-structure extraction algorithm for accurate and robust skeletons. In *Proceedings of the Eighth Pacific Conference on Computer*

Graphics and Applications, pp. 281–289, 2000. doi: 10.1109/PCCGA.2000.883951.

Schmidt, M., Motta, A., Sievers, M., and Helmstaedter, M. RoboEM: automated 3D flight tracing for synaptic-resolution connectomics. *Nat Methods*, 21(5):908–913, May 2024.

seung-lab contributors. cloud-volume: serverless python client for neuroglancer precomputed volumes, 2026a. URL https://github.com/seung-lab/cloud-volume.

seung-lab contributors. kimimaro: skeletonize densely labeled 3D segmentations with a TEASAR-derived method, 2026b. URL https://github.com/seung-lab/kimimaro.

Shapson-Coe, A., Januszewski, M., Berger, D. R., Pope, A., Wu, Y., Blakely, T., and et al. A petavoxel fragment of human cerebral cortex reconstructed at nanoscale resolution. 384(6696):eadk4858, 2024. doi: 10.1126/science.adk4858. URL https://www.science.org/doi/10.1126/science.adk4858.

Sheridan, A., Nguyen, T. M., Deb, D., Lee, W.-C. A., Saalfeld, S., Turaga, S. C., Manor, U., and Funke, J. Local shape descriptors for neuron segmentation. 20(2):295–303. ISSN 1548-7105. doi: 10.1038/s41592-022-01711-z. URL https://www.nature.com/articles/s41592-022-01711-z.

Silversmith, W. cloud-volume, 2021a. URL https://doi.org/10.5281/zenodo.5671443.

Silversmith, W. kimimaro, 2021b. URL https://doi.org/10.5281/zenodo.5539913.

Tavakoli, M. R., Lyudchik, J., Januszewski, M., Vistunou, V., Agudelo Dueñas, N., Vorlaufer, J., Sommer, C., Kreuzinger, C., Oliveira, B., Novarino, G., Jain, V., and Danzl, J. G. Light-microscopy-based connectomic reconstruction of mammalian brain tissue. *Nature*, pp. 1–13, May 2025.

The MICrONS Consortium. Functional connectomics spanning multiple areas of mouse visual cortex. *Nature*, 640 (8058):435–447, April 2025.

Zung, J. et al. An error detection and correction framework for connectomics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://proceedings.neurips.cc/paper/2017/file/4500e4037738e13c0c18db508e18d483-Paper.pdf.

A. Appendix A - Data Generation

A.1. Training/Testing Data Sample Generation

Collecting errors and ground truth corrections To extract 'ground truth' merge corrections of split errors and split corrections of merge errors, as well as 'false' corrections at control sites, root supervoxel sets from ChunkedGraph, and skeletons from CloudVolume were utilized. Set differences between proofread roots and their ancestors allow comprehensively determining locations that involved errors and corrections for any given root, and the respective ground truth corrections. Specifically, a proofread root's gain of a spatially contiguous set supervoxels implies a merge correction of a split error there, with the original root containing these supervoxels constituting a ground truth merge partner. Vice versa, a proofread root's loss of supervoxels at a given site implies a split correction of a merge errror. The same principle was applied for extracting errors and corrections from ExM segmentation data, but on the level of segmentation voxels. Furthermore, 'complex' roots that showed multiple transitive corrections (e.g. root A merging in root B, while root B itself was split into C and D) were excluded, as the lack of availability of an edit graph and intermediate roots made accurate rendering of errors and corrections intractable.

A.1.1. SPLIT ERRORS

Due to the continuous nature of splits, it is not trivial to determine whether a model-generated split (defined by a set of source and sink points) is valid, nor whether it is correct relative to the ground truth. Therefore, each split induced by model-selected source and sink points was first computed using ChunkedGraph's multi-cut algorithm and subsequently evaluated using the following heuristic.

A split is considered *good* if and only if

$$SV_{ig} > k \cdot SV_{ib} \quad \forall i \in \mathcal{R},$$

where $\mathcal{R}$ denotes the set of relevant ground truth roots with size greater than threshold t, evaluated within a local cutout of spatial extent e.

Relevant roots are defined as all ground truth roots R_i that overlap the root being split. For each such root, the split component with the larger overlap is designated as the base component, while the other component is treated as the split-off component. Good supervoxels SV_{ig} correspond to supervoxels in the split-off component that do not overlap with root R_i, whereas bad supervoxels SV_{ib} correspond to supervoxels in the split-off component that do overlap with R_i and are therefore erroneously separated. The precision factor k controls how many times more good supervoxels than bad supervoxels must be present for each relevant root in order for the split to be considered correct.

These parameters were set empirically based on author judgment to $k = 20$, $t = 1000$, and $e = 7500\,\mathrm{nm}$.

A.1.2. COLLECTING ERROR CANDIDATES

Using root skeletons fetched from the server, or generated via kimimaro, error candidates were generated heuristically: Skeleton nodes of degree 1 were treated as endpoints and split error candidates, skeleton nodes of degree ¿ 2 were treated as junctions and merge error candidates. To determine precision and recall of error candidates, actual errors and candidates were matched with [THRESHOLD] distance threshold.

A.2. Rendering

All samples were rendered using a unified mesh rendering pipeline relying on octarine; participating meshes were downloaded from CloudVolume. All renderings were centered on coordinates of errors and error candidates, with a view extent of 5000-8000nm.

A.2.1. VISUALIZATION OF SPLITS

To visualize splits, split components were matched to low-level meshes (L2 meshes) based on ChunkedGraph, which were rendered in distinct colors. Low-level meshes that were ambiguously associated with both components were colored by bisecting them with a plane orthogonal to the vector between the two sides' outer vertices that touched each colored mesh.