# ConnectomeVLM: Human-Level Connectomics Proofreading using Vision Language Models

**Anonymous Authors**[1]

## Abstract

## 1. Introduction

In neuroscience, ground-truth maps of neural connectivity would greatly advance our understanding of how brains produce behavior and cognition. 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 zero-shot performance on tasks including identifying neuronal segments types 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?

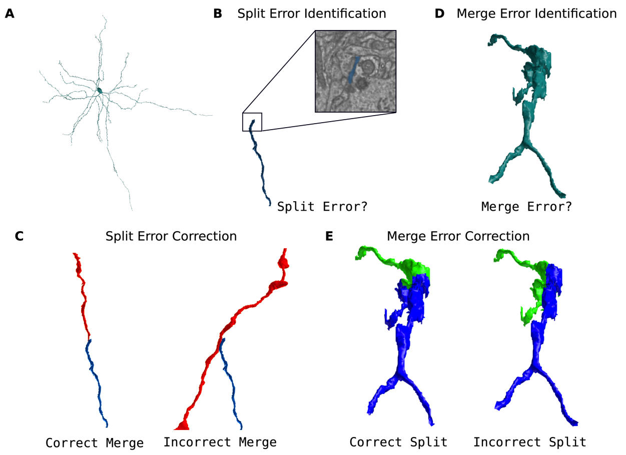

*Figure 1.* A: Mouse Neuron, B: Split Error Identification, endpoint candidate and EM image, C: Split Error Correction, D: Merge Error Identification, E: Merge Error Correction

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 visual reasoning complexity: (1) Vision-only models (CNNs), (2) vision-language models (VLMs) without generation (linear probes on SigLip), and (3) generative VLMs (fine-tuned

[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.

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), and interpretability (whether models learn meaningful heuristics).

Our evaluation reveals task-specific scaling laws: simple 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 finetuning. We demonstrate cross-species generalization, with models trained on mouse cortex transferring to Drosophila, zebrafish, and human datasets with varying degrees of performance degradation. We show 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 3 describes our training methods and data generation pipeline. Section 4 presents results organized by evaluation axis: performance and scaling behavior, cross-species generalization, calibration analysis, 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. Proofreading can be canonically broken down into three stages: (1) error candidate localization (identifying potential error sites), (2) error identification (determining whether a candidate represents a true error), and (3) error correction. 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) 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 datasets across four species were used for evaluating generalization to other species.

| Species | Reference |
|---|---|
| Mouse | The MICrONS Consortium (2025) |
| Fly | FlyWire Consortium (2024) |
| Human | Shapson-Coe et al. (2024) |
| Zebrafish | Petkova et al. (2025) |

*Table 1.* Datasets used in this work. EM = Electron Microscopy

All datasets exposed raw images and their raw and agglomerated segmentation, as well meshes, skeletons, and detailed manual proofreading history, which we accessed via CAVEClient (Dorkenwald et al., 2025), ChunkedGraph (CAVE-connectome contributors, 2026) and CloudVolume (Silversmith, 2021; seung-lab contributors, 2026).

### 3.3. Data Generation

These datasets are used to generate samples for supervised training on four binary choice tasks: Identification and correction — of split errors and merge errors (cf. Figure 1).

**Error Identification Samples:** Neuron corrections in the proofreading history, and morphological differences before and after proofreading are used to extract samples of error locations for the error identification tasks. Neuronal junctions and endpoints without any errors are used as no-samples.

**Error Correction Samples:** Mergers from the edit history and merges of random adjacent roots serve respectively as samples of correct and incorrect split error corrections. The continuous space of possible split corrections makes it more difficult to derive unambiguous yes- and no-examples from only the edit history. Thus, we generate samples by generating plausible sink and source points, manually computing resulting splits, and evaluating them as correct or incorrect by comparing to the final proofread volume.

Finally, three orthogonal renderings (front, side, top) of neuronal segment meshes are generated for all data samples at their respective locations. For split error identification, we also incorporate 3D slices of the EM data for additional context. To preclude contamination, we ensure that the same segments and locations do not appear across different splits. *A full breakdown of the data generation process can be found in Appendix A.*

### 3.4. Model Training and Evaluation

#### 3.4.1. Linear Probe Applied to SigLIP

The linear probe training uses a frozen vision encoder (SigLIP-2 by default, which shares the same architecture as Qwen3-VL's vision encoder) to extract visual features from input images, applies pooling (mean, max, or CLS token) to aggregate patch-level representations into fixed-size feature vectors, and then trains a lightweight classifier on top. The system uses a logistic regression model. Features are cached to disk for efficiency, and the pipeline includes robust diagnostics including group-based train/val splitting to prevent data leakage (ensuring samples from the same neuron do not appear in both splits), 5-fold stratified cross-validation for stable accuracy estimates, and extensive data leakage checks at both the image and sample level.

#### 3.4.2. Vision-Language Model Fine-tuning

We fine-tune Qwen3-VL-32B-Instruct, a 32-billion parameter vision-language model, using Low-Rank Adaptation (LoRA) (?) with rank 16. Training is performed on Modal cloud infrastructure using 2× H100 GPUs with the Unsloth library for memory-efficient optimization. LoRA adapters are applied to the language model's attention layers (query, key, value, and output projections) and MLP layers (gate, up, and down projections), as well as the vision-language merger module that bridges the vision encoder to the language model. The vision encoder itself remains frozen. We use the AdamW optimizer (?) with 8-bit precision, a learning rate of 2e-4 with reduce-on-plateau scheduling, and a maximum of 500 training steps per task.

We train separate LoRA adapters for five proofreading tasks: merge error identification, merge action verification, split action verification, endpoint error identification, and endpoint error identification with EM context. Each task uses a batch size of 4–8 with class balancing to address label imbalance (split action uses undersampling while others use oversampling). All tasks are framed as binary classification problems where the model must output "yes" or "no" within XML answer tags. Generated data (cf. Section 3.3) is stratified into train/validation/test splits (128 validation, 512 test samples) using group-based splitting by spatial location to prevent data leakage between splits.

The training pipeline employs lazy image loading to handle large datasets efficiently, loading images on-the-fly during batch collation rather than upfront. The model is trained only on assistant responses (not user prompts) using Unsloth's vision data collator with Qwen3-VL chat template delimiters. Checkpoints are saved every 100 steps with the best model selected by validation loss. Training configurations, test set indices, and dataset hashes are persisted alongside model weights to ensure reproducibility and enable consistent evaluation across runs.

#### 3.4.3. Vision-Language Model Evaluation

Fine-tuned LoRA adapters are evaluated on held-out test sets using FastVisionModel inference with the base Qwen3-VL-32B-Instruct model. For each test sample, the model receives the same 3-view orthogonal renderings used during training along with the task-specific prompt. The processor applies the Qwen3-VL chat template to format inputs, and the model generates responses with a maximum of 512 new tokens using greedy decoding. Answers are extracted from the generated text by parsing XML answer tags (`<answer>yes</answer>` or `<answer>no</answer>`) and compared against ground truth labels using exact string matching.

Evaluation supports several ablation modes: (1) base model evaluation without adapters to measure zero-shot performance, (2) blank image controls where all images are replaced with uniform gray 1024×1024 canvases to test for prompt exploitation, (3) simple prompt controls using generic "What do you see?" prompts to verify task-specific reasoning, and (4) answer-only mode that removes chain-of-thought analysis instructions to isolate decision-making from reasoning. Test set indices are loaded from `test_indices.json` files saved during training to ensure evaluation uses the exact same held-out samples across runs. Results are saved to parquet files containing predictions, ground truth labels, full prompts, model responses, and sample identifiers for downstream analysis.

#### 3.4.4. CNN Baseline Training

As a non-transformer baseline, we fine-tune pretrained ResNet-50 models (?) (initialized with ImageNet weights) on the same proofreading tasks. For tasks with multiple images per sample, images are arranged into grids using automatic layout detection (e.g., 1×2 for 2 images, 2×2 for 4 images) before being resized to 224×224 pixels (the standard ResNet input size). We apply ImageNet normalization (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]) and data augmentation during training including random horizontal flips, ±10° rotation, and color jittering (brightness and contrast ±20%).

Training uses the AdamW optimizer with a learning rate of 1e-3, weight decay of 1e-4, and cosine annealing learning rate scheduling over 10 epochs. We use large batch sizes (256) to leverage the smaller model size compared to VLMs, with 8 dataloader workers for efficient I/O. The final fully-connected layer is replaced to match the number of task classes, and we support two training modes: (1) full fine-tuning where all parameters are trainable, and (2) linear probe mode where the backbone is frozen and only the classifier head is trained. Early stopping with patience of 5 epochs is applied based on validation loss.

Dataset splits use the same stratified group-based splitting as VLM training, with identical train/val/test indices saved to `test_indices.json` for cross-model comparison. Class balancing is applied via oversampling minority classes to match the majority class count. Models are trained on a single A10G GPU using standard PyTorch training loops with cross-entropy loss. The best model checkpoint (by validation accuracy) is saved and used for final test set evaluation.

#### 3.4.5. CNN Baseline Evaluation

ResNet models are evaluated on held-out test sets by loading the best checkpoint and computing per-class accuracy. For each test sample, images are converted to grid format with the same layout and preprocessing used during training. The model outputs logits for each class, and predictions are obtained via argmax. Final metrics include overall accuracy and per-class accuracy to identify class-specific performance differences. Evaluation supports the same test set filtering as VLM evaluation, loading indices from `test_indices.json` to ensure fair comparison. Cross-dataset evaluation is also supported by specifying alternative dataset paths, enabling assessment of model generalization to different species or imaging conditions.

### 3.5. Calibration Evaluation

We evaluate model calibration using Expected Calibration Error (ECE), which measures the alignment between predicted confidence and actual correctness. For a binary classification task with $N$ predictions, we partition predictions into $M$ bins based on confidence scores and compute:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| \qquad (1)$$

where $B_m$ is the set of predictions in bin $m$, $\text{acc}(B_m)$ is the accuracy within the bin, and $\text{conf}(B_m)$ is the average predicted confidence. We use $M = 10$ uniform bins spanning $[0, 1]$.

**Linear Probe.** The linear probe uses logistic regression, outputting calibrated probabilities $p \in [0, 1]$ through the sigmoid function. For binary tasks, we use $p$ directly as the confidence for the positive class and compute ECE across the test set.

**ResNet-50.** The ResNet model outputs logits that are passed through a sigmoid activation to produce class probabilities. These probabilities serve as confidence scores for ECE computation. We evaluate calibration on held-out test sets after fine-tuning on task-specific data.

**Vision-Language Model (VLM).** Unlike discriminative models, our VLM (Qwen3-VL-32B) generates natural language responses. To extract calibration information, we prompt the model to verbalize its confidence as part of its response (e.g., "I am 80% confident this is a merge error"). We parse the verbalized confidence from the text output and use this as the predicted probability for ECE computation. For majority voting evaluation (25 samples per question), we compute both individual-level calibration (using confidence from each of 25 votes) and majority-level calibration (using the proportion of votes as confidence).

For all models, we compute ECE on the same held-out test sets to enable direct comparison. Well-calibrated models should have ECE $< 0.05$, indicating predicted confidences closely match empirical accuracies.

## 4. Results

### 4.1. Performance: Task-Specific Models Match or Exceed Frontier Models

Table 2 presents our benchmark results across four proofreading tasks, comparing task-specialized models (linear probes, fine-tuned ResNets, and fine-tuned VLMs) against human annotators and zero-shot frontier models (GPT-5 and Gemini-3-Pro). Our central finding is that **task-specialized models trained on modest amounts of data consistently match or exceed the performance of frontier models**, with different architectures excelling on different tasks.

**Split Error Correction** represents the simplest task in our benchmark, requiring only local topology matching. Here, a linear probe on frozen SigLIP-2 features achieves 97.2±3.4% accuracy, exceeding both human performance (92.3±5.0%) and all other models. The fine-tuned ResNet-50 achieves comparable performance (96.1±2.1%), while the fine-tuned VLM (93.8±2.0%) matches human-level accuracy. Notably, zero-shot frontier models GPT-5 (87.5±2.8%) and Gemini-3-Pro (93.0±2.2%) underperform the specialized models, suggesting that strong visual priors from contrastive pre-training combined with minimal task-specific supervision suffice for this task.

**Split Error Identification** requires multimodal fusion between 3D segmentation and EM image data. Here, the fine-tuned VLM substantially outperforms all other approaches (83.6±3.2%), exceeding human performance (61.3±8.9%) by over 20 percentage points. The linear probe (72.2±1.6%) and ResNet (67.0±3.5%) also surpass human performance, while frontier models again struggle (GPT-5: 57.0%, Gemini-3-Pro: 52.3%). This suggests that explicit multimodal integration during fine-tuning provides advantages that zero-shot prompting cannot replicate.

**Merge Error Identification** demands global morpho-

*Table 2.* Proofreading benchmark results showing balanced accuracy (%) ± error on four tasks. Bold indicates best model per task. Human: standard error; Linear Probe (SigLIP-2): 5-fold CV standard deviation; ResNet-50 Finetuned, Finetuned VLM (Qwen3-VL-32B), GPT-5, Gemini-3-Pro: bootstrap (1000 iter.). All results on MICrONS mouse cortex test sets.

| Task | Human | Linear Probe | ResNet-50 Finetuned | VLM | GPT-5 | Gemini-3-Pro |
|---|---|---|---|---|---|---|
| Split Error Correction | 92.3±5.0 | **97.2±3.4** | 96.1±2.1 | 93.8±2.0 | 87.5±2.8 | 93.0±2.2 |
| Split Error Identification | 61.3±8.9 | 72.2±1.6 | 67.0±3.5 | **83.6±3.2** | 57.0±3.4 | 52.3±2.9 |
| Merge Error Identification | 73.9±8.3 | 64.5±4.3 | 79.2±3.3 | **81.2±3.5** | 58.6±3.6 | 68.8±3.8 |
| Split Action Evaluation | 78.3±6.7 | 60.4±4.4 | **91.6±3.4** | 71.1±3.6 | 62.5±4.3 | 56.2±3.8 |

logical reasoning to detect fused neurons. The fine-tuned VLM (81.2±3.5%) narrowly outperforms the ResNet (79.2±3.3%), with both exceeding human accuracy (73.9±8.3%). The linear probe struggles on this complex task (64.5±4.3%), falling below human performance and suggesting that frozen features lack the capacity for global reasoning required here. Frontier models again underperform (GPT-5: 58.6%, Gemini-3-Pro: 68.8%), with Gemini approaching but not reaching human-level accuracy.

**Split Action Evaluation** requires counterfactual reasoning about proposed corrections. Surprisingly, the fine-tuned ResNet dramatically outperforms all other models (91.6±3.4%), exceeding human accuracy (78.3±6.7%) by 13 percentage points. The VLM (71.1±3.6%) falls below human performance, as do the linear probe (60.4±4.4%) and frontier models (GPT-5: 62.5%, Gemini-3-Pro: 56.2%). This unexpected pattern suggests that the visual features learned by ResNet during fine-tuning may better capture the relevant geometric patterns than language-grounded representations, though further investigation is needed.

**Key insights:** (1) No single architecture dominates across all tasks. Linear probes excel at simple topology matching, VLMs at multimodal fusion and global reasoning, and ResNets at specific correction evaluation tasks. (2) Task-specialized models consistently outperform zero-shot frontier models, often by substantial margins (10–30 percentage points), demonstrating that scale alone does not replace task-specific supervision. (3) All three model classes achieve human-level or superhuman performance on at least some tasks, validating the feasibility of AI-assisted proofreading across the difficulty spectrum.

### 4.2. Data Efficiency: Task Complexity Determines Scaling Requirements

Figure **??** presents scaling curves showing how model performance varies with training data across our four proofreading tasks. Our key finding is that **different tasks exhibit dramatically different data requirements**, with simple topology-matching tasks saturating at hundreds of samples while complex reasoning tasks require thousands.

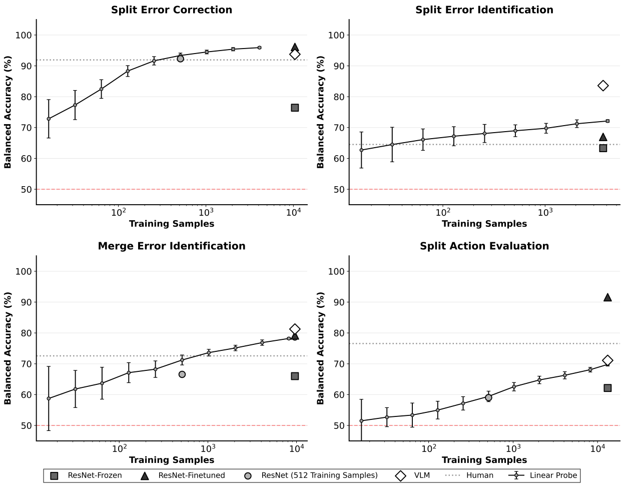

*Figure 2.* Caption

**Split Error Correction** demonstrates the most favorable scaling behavior. The linear probe on frozen SigLIP features achieves human-level performance (92.3%) with fewer than 500 training samples, reaching 97.2% accuracy by 2,000 samples. This rapid saturation suggests that pre-trained contrastive vision encoders already contain the geometric features necessary for local topology matching. In contrast, ResNet-Frozen (training only the classification head while keeping the ImageNet-pretrained backbone frozen) plateaus around 77% accuracy even with all available training data, falling substantially short of human performance. This gap highlights a critical difference in visual priors: SigLIP's contrastive language-image pre-training produces representations better suited to this geometric reasoning task than ResNet's ImageNet classification pre-training. The fine-tuned ResNet (96.1%) and VLM (95.1%) both reach near-human performance, but require full model fine-tuning rather than simple linear probing.

**Split Error Identification** shows more gradual scaling. The linear probe improves from 64% at 32 samples to 72% at 2,000 samples, modestly exceeding human performance (61.3%) but falling well short of the fine-tuned VLM (83.6%). ResNet-Frozen (63%) and ResNet-Finetuned (67%) both struggle with this multimodal fusion task, suggesting that even with full fine-tuning, CNN architectures lack the capacity to effectively integrate segmentation and EM texture information. The VLM's substantial advantage (¿11 percentage points over linear probe, ¿16 points over fine-tuned ResNet) demonstrates the value of architecture explicitly designed for multimodal reasoning.

**Merge Error Identification** exhibits continued improvement even at the largest data scales. The linear probe scales from 59% at 32 samples to 78% at 10,000 samples, approaching but not quite reaching the fine-tuned VLM's performance (81.2%). ResNet-Frozen (66%) again underperforms, while the ResNet trained on 512 samples achieves only 67%, suggesting this task requires substantial training data. The VLM maintains its advantage across all data scales, though the gap narrows as training data increases, suggesting that with sufficient data, simpler models can partially close the performance gap for global reasoning tasks.

**Split Action Evaluation** presents the most surprising scaling pattern. Here, the fine-tuned ResNet dramatically outperforms all other approaches, achieving 91.6% accuracy and substantially exceeding both human performance (78.3%) and the VLM (71.1%). The linear probe shows steady improvement from 51% to 71% but never surpasses human-level performance. ResNet-Frozen (62%) again struggles, reinforcing that ImageNet priors alone are insufficient. The ResNet's exceptional performance on this task, despite underperforming on other complex reasoning tasks, suggests that counterfactual evaluation of proposed splits may rely more heavily on low-level geometric features that CNNs excel at extracting, rather than the global semantic reasoning that benefits VLMs on other tasks.

**Key insights:** (1) Simple tasks (split correction) achieve human-level performance with minimal data (<500 samples) using linear probes on appropriate pre-trained encoders. (2) Complex tasks (merge identification, split action evaluation) require 10× more data and benefit from full model fine-tuning. (3) ResNet models with frozen backbones consistently underperform across all tasks, demonstrating that ImageNet pre-training provides inadequate visual priors for connectomics reasoning. (4) The choice of pre-training objective matters: SigLIP's contrastive language-image training produces more transferable features than ImageNet classification. (5) No single architecture dominates: task structure determines whether simple linear probes, fine-tuned CNNs, or VLMs provide optimal performance.

### 4.3. Generalization: VLMs Transfer Robustly Across Species and Modalities

To evaluate whether models learn general proofreading principles or memorize dataset-specific features, we applied models fine-tuned on mouse cortex data to analogous tasks across four other datasets spanning three additional species (fly, zebrafish, human) and two imaging modalities (EM, ExM). This represents a challenging generalization test: these species exhibit substantial morphological differences (e.g., fly neurons are more heavily branched with soma localized away from dendritic arbors, unlike the soma-proximal dendrites typical of mammalian cortical neurons), and expansion microscopy introduces different artifacts and texture patterns than electron microscopy.

Figure **??** reveals a striking pattern: **vision-language models generalize far more robustly than CNNs, particularly on complex reasoning tasks**, while simple topology-matching tasks show strong generalization across all architectures.

**Split Error Correction** demonstrates strong cross-species transfer for all models. The ResNet-Finetuned maintains 83.1–88.3% accuracy across fly, zebrafish, and human (compared to 96.1% on mouse), degrading by only 8–13 percentage points. The linear probe shows similar robustness (80.4–83.1% across species vs. 94.4% on mouse), as does the VLM (83.1–89.4% vs. 95.9% on mouse). This consistent performance across architectures suggests that local topology matching relies on universal geometric features that transfer readily across species and modalities.

**Split Error Identification** reveals moderate generalization with VLM advantages. The VLM maintains 66.0–72.7% accuracy on zebrafish and human (vs. 83.6% on mouse), degrading by 11–18 percentage points. The linear probe drops to 66.0–69.1% (vs. 72.1% on mouse), while humans achieve 55.6–61.3% on the new species. The VLM's ability to maintain performance above human-level on zebrafish (72.7% vs. 61.3% human) despite training only on mouse data demonstrates effective transfer of multimodal fusion capabilities.

**Merge Error Identification** exposes catastrophic ResNet failure on new species. While the ResNet achieved 79.2% on mouse, it collapses to 53.1% on fly, 22.9% on zebrafish, and 10.4% on human cortex. This dramatic degradation (up to 69 percentage points) suggests the ResNet overfits to species-specific morphological patterns during training and fails to learn transferable principles of merge error detection. In stark contrast, the VLM maintains 78.1–81.2% accuracy across all species (vs. 81.2% on mouse), degrading by at most 3 percentage points. Even the linear probe (46.9–59.4% on new species vs. modest performance on mouse) substantially outperforms the fine-tuned ResNet

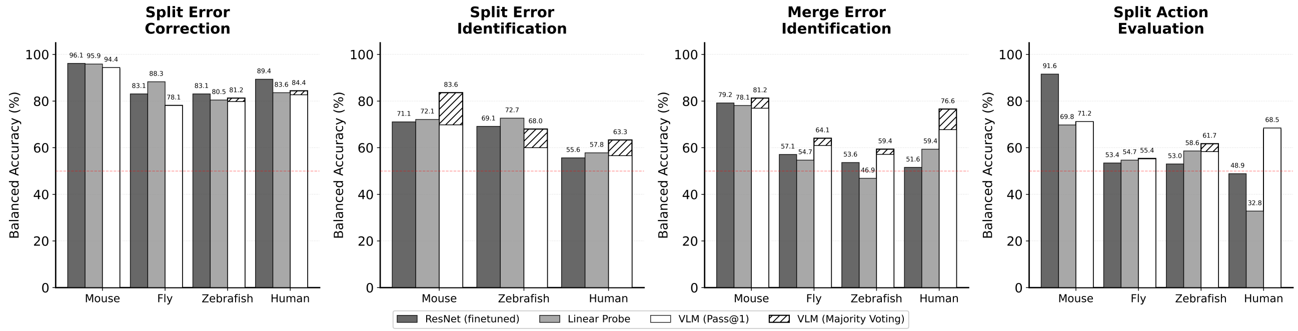

*Figure 3.* Caption

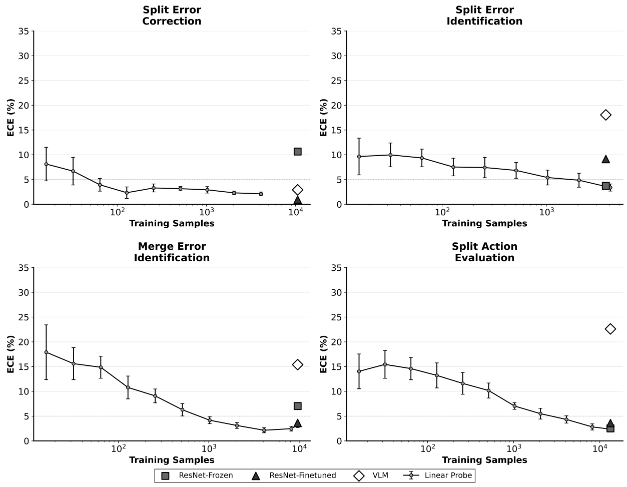

*Figure 4.* Caption

on zebrafish and human, highlighting how catastrophically CNN fine-tuning can fail at cross-species transfer for complex reasoning tasks.

**Split Action Evaluation** shows moderate generalization with architectural tradeoffs. The ResNet, which dominated on mouse data (91.6%), drops to 51.0–69.6% on new species, a degradation of 22–41 percentage points. The VLM (Pass@1) shows more graceful degradation from 71.2% on mouse to 54.7–69.6% on new species (6–16 point drop). Notably, VLM majority voting (generating multiple responses and taking the consensus) consistently improves performance by 1–7 percentage points across species, achieving 55.4–71.2% and matching or exceeding ResNet performance on all non-mouse datasets despite ResNet's initial advantage on mouse.

**Key insights:** (1) Simple geometric tasks generalize robustly across all architectures, with 8–15% accuracy degradation. (2) VLMs show dramatically superior cross-species transfer on complex reasoning tasks, maintaining near-training performance while ResNets catastrophically fail. (3) Fine-tuned ResNets overfit to species-specific morphological patterns, particularly for global reasoning tasks, limiting their practical deployability. (4) The VLM's consistent performance across species (within 3–18 percentage points of training accuracy) validates that it learns transferable visual principles rather than memorizing mouse-specific features. (5) Ensemble methods (majority voting) provide additional robustness gains for VLMs at minimal computational cost.

### 4.4. Calibration: Data Scale Improves Discriminative Models but Not VLMs

Calibration is critical for human-AI collaboration: well-calibrated models enable human reviewers to trust high-confidence predictions while scrutinizing uncertain cases. We evaluate calibration using Expected Calibration Error (ECE), which measures the alignment between predicted confidence and actual correctness. Well-calibrated models should have ECE $< 5\%$, indicating that when a model reports 80% confidence, it is correct approximately 80% of the time.

Figure **??** reveals a striking divergence: **discriminative models (linear probes and ResNets) improve calibration with training data scale, while VLMs remain poorly calibrated even at large data scales**, particularly on complex reasoning tasks.

**Split Error Correction** demonstrates strong calibration across all discriminative models. The linear probe achieves excellent calibration (ECE ≈ 2–3%) at large data scales, improving from 9% ECE at 32 samples. The ResNet-Finetuned achieves near-perfect calibration (ECE $< 1\%$) even at moderate data scales. In contrast, ResNet-Frozen shows poor

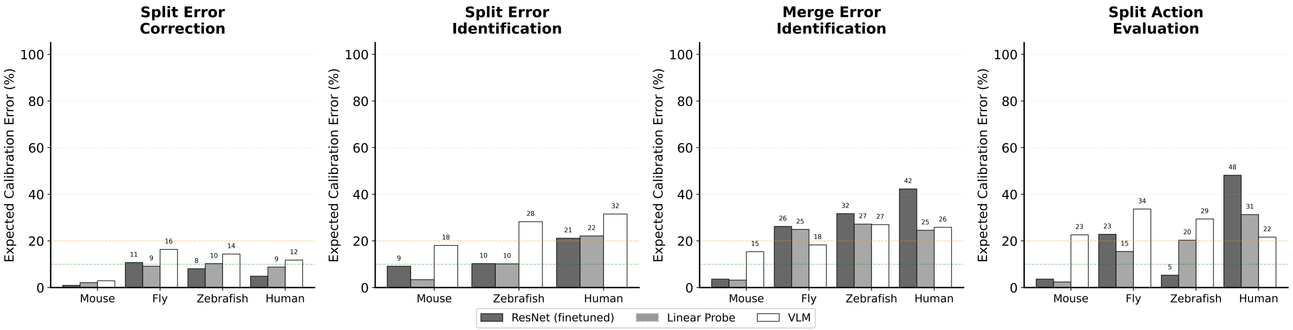

*Figure 5.* Caption

calibration (ECE ≈ 11%), consistent with its performance plateau in Figure **??**. The VLM achieves reasonable calibration (ECE ≈ 3%), suggesting that verbalized confidence aligns moderately well with actual accuracy for this simple task.

**Split Error Identification** exposes VLM calibration failures. While the linear probe improves from 10% to 5% ECE with increasing data, and both ResNet configurations achieve strong calibration (ECE ≈ 3–4%), the VLM shows catastrophically poor calibration (ECE ≈ 18%). This 18% miscalibration indicates that the VLM's verbalized confidence bears little relationship to its actual correctness on this multimodal fusion task. Despite achieving the highest accuracy (83.6%, Table 2), the VLM cannot reliably communicate when it is uncertain, severely limiting its utility for human-AI collaboration.

**Merge Error Identification** shows dramatic linear probe improvement. The linear probe's ECE decreases from 18% at 32 samples to just 2–3% at 10,000 samples, demonstrating that calibration can be learned from data even for complex reasoning tasks. ResNet-Finetuned achieves similarly strong calibration (ECE ≈ 3%), while ResNet-Frozen shows moderate miscalibration (ECE ≈ 7%). The VLM again exhibits substantial miscalibration (ECE ≈ 15%), indicating that its confidence reports are systematically unreliable for global morphological reasoning.

**Split Action Evaluation** reveals the worst VLM miscalibration. The linear probe achieves excellent calibration at scale (ECE ≈ 2–3%), improving from 15% at small data scales. ResNet-Finetuned shows strong calibration (ECE ≈ 3%). However, the VLM suffers from severe miscalibration (ECE ≈ 23%), the highest across all tasks. This 23% ECE means that the VLM's verbalized confidence is essentially uninformative about actual correctness, potentially leading human reviewers to trust incorrect predictions or unnecessarily scrutinize correct ones.

**Key insights:** (1) Training data scale consistently improves calibration for discriminative models, with linear probes achieving ECE $< 5\%$ at 1,000–10,000 samples across all tasks. (2) ResNets with frozen backbones show poor calibration, consistent with their limited representational capacity. (3) VLMs exhibit severe miscalibration (ECE 15–23%) on complex reasoning tasks despite achieving high accuracy, indicating that verbalized confidence does not reflect internal model uncertainty. (4) The VLM's calibration failure is worst precisely on the tasks where it achieves highest performance (Split Error Identification: 83.6% accuracy, 18% ECE), suggesting that accuracy gains do not automatically translate to reliable uncertainty quantification. (5) For deployment in human-AI collaborative systems, discriminative models provide more trustworthy confidence estimates than generative VLMs, even when VLMs achieve superior raw accuracy.

**Cross-species calibration.** Figure **??** (Appendix) extends this calibration analysis to zero-shot transfer across species. The key finding is that **calibration properties largely transfer with performance**: models that are well-calibrated on mouse data remain well-calibrated on new species, while poorly calibrated models remain poorly calibrated.

For Split Error Correction, all models maintain good calibration across species (ECE $< 16\%$). For Split Error Identification, the VLM remains poorly calibrated across all species (ECE 28–32%), while discriminative models maintain ECE around 10–20%. Most dramatically, for Merge Error Identification, the VLM shows catastrophic miscalibration on zebrafish (ECE 82%) and human data, far exceeding its already-poor calibration on mouse (ECE 15%). This suggests that as VLM performance degrades on out-of-distribution data, calibration degrades even more severely. In contrast, ResNet and linear probe calibration remains relatively stable across species (ECE 15–30%), even when accuracy drops substantially.

This cross-species analysis reinforces that discriminative models provide more reliable uncertainty estimates not only on training distributions but also when deployed to new species and imaging modalities, making them more suitable for production deployment where models must handle diverse datasets.

### 4.5. Interpretation

## 5. Discussion

### 5.1. Estimating the Cost of 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).

We subsampled the proofread neurons from each dataset—mouse: $n = 500$, fly: $n = 1{,}000$ —and retrieved complete edit histories, categorizing each operation as merge (consolidating over-split fragments) or split (separating under-segmented regions). For each sampled neuron, we computed the edit count distribution and identified heavy-tail concentration (95th percentile threshold). To project costs to full datasets, we applied linear extrapolation validated via cross-sample statistical consistency (differences between $n = 100$ and $n = 500$ mouse samples were $< 7\%$). Computational costs were modeled as:

$$\text{GPU Cost} = \frac{(\text{Merge operations}) + (\text{Split operations}) \times (\text{inference time per operation})}{3600 \text{ sec/hr}} \times \$2/\text{GPU-hour} \qquad (2)$$

where inference time ranges 1–5 seconds per operation (Qwen-32B model on dual H100 GPUs), and GPU rate is $2/hour.

**Edit distributions reveal species-level segmentation differences.** Mouse neurons require dramatically more proofreading: $411 \pm 288$ edits per neuron (median=335) versus $17.5 \pm 32$ edits per neuron (median=8) for fly—a 23.4× intensity difference. This reflects distinct anatomical challenges: mammalian cortex has higher neuronal density and more complex morphology, leading to more substantial segmentation errors requiring correction (Figure 6A).

**Operation ratios expose segmentation biases.** Mouse proofreading exhibits balanced merge-to-split ratios (46.3% merge, 53.7% split), indicating that the initial segmentation contains comparable over- and under-segmentation errors. Fly proofreading is merge-dominated (74.0% merge, 26.0% split), revealing systematic oversegmentation in the FAFB volume where numerous small fragments must be consolidated. These divergent patterns suggest that cost-effective proofreading systems must handle both error modes (Figure 6B).

**Heavy-tail distribution concentrates computational effort.** Approximately 5% of neurons exceed the 95th percentile threshold, but their contribution differs markedly. Mouse heavy-tail neurons (¿971 edits) account for only 14.1% of total edits, indicating relatively uniform workload distribution. In contrast, fly heavy-tail neurons (¿58 edits) concentrate 33.3% of total edits despite 23.4× lower per-neuron baseline, revealing that fly proofreading is dominated by a small number of structurally complex outliers.

**Cost projections: Full-dataset extrapolations.** Linear extrapolation to the full MICrONS dataset (2,314 neurons in 1 mm$^3$) yields 951,387 total edits for mouse, and to the full FlyWire dataset (139,255 neurons) yields 2,442,671 total edits for fly. Using the realistic per-operation inference time of 2.25 seconds (mean of 2.0–2.5 s range), the GPU-hour costs at dual H100 pricing ($2/hour) are:

- **Mouse (MICrONS 1 mm$^3$)**: 595 GPU-hours, $1,189 cost
- **Fly (FlyWire, full brain)**: 1,527 GPU-hours, $3,053 cost
- **Mouse (full cortex, estimated)**: 2,572,500 GPU-hours, $5,145,000 cost

The dramatic difference in the third estimate reflects extrapolation to the full mouse neocortex (112 mm$^3$, approximately 10 million neurons). Despite 23.4× lower per-neuron effort than mouse, fly's 60× larger dataset makes it 2.6× more expensive than the MICrONS sample. However, projecting to the full mammalian cortex reveals why connectome proofreading remains challenging: GPU costs would scale to over $5 million for complete mouse cortex coverage, underscoring the computational bottleneck in large-scale connectomics.

**Cost sensitivity and realistic ranges.** Across the 1–5 second per-operation range, costs scale proportionally: mouse MICrONS ranges $264–$1,320, fly ranges $678–$3,390. The realistic 2.0–2.5 second window (shown in Figure **??**A) encompasses costs of approximately $1,000–$1,200 (mouse) and $2,700–$3,400 (fly). These costs are highly sensitive to model latency; improving inference speed from 2.5 to 2.0 seconds saves 20% on GPU hours.

**Statistical validation.** Cross-sample consistency between

$n = 100$ and $n = 500$ mouse samples validates our extrapolation approach: mean edits differ by 2.5%, projected totals differ by 2.5%, and heavy-tail contribution differs by 32.6%. Similar consistency for fly ($n = 100$ vs $n = 1{,}000$: 6.6% and 6.5% differences) demonstrates robust sampling methodology.

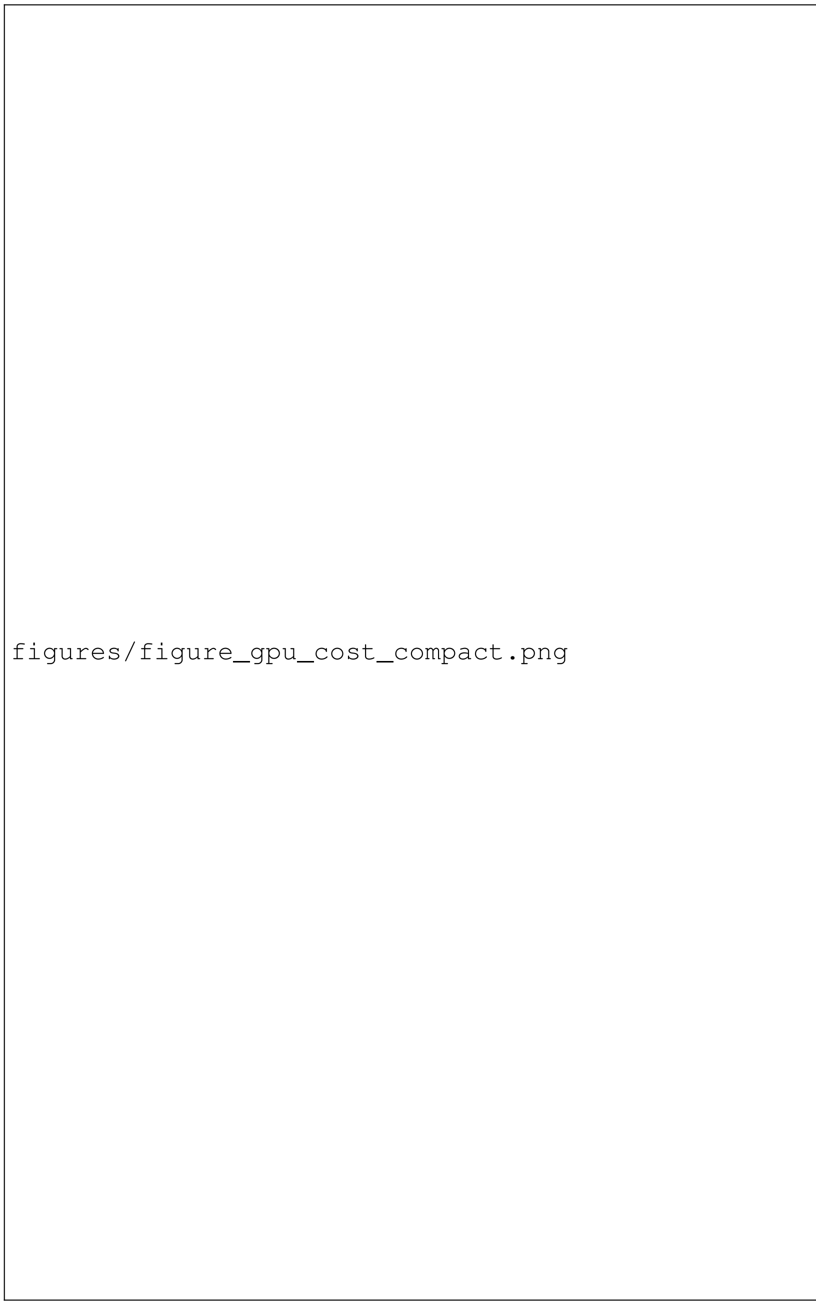

*Figure 6.* **GPU Computational Cost Analysis Across Species. (A) Heavy-Tail Edit Distribution.** Rank-ordered edit counts for mouse (n=500, blue) and fly (n=1,000, red) reveal concentrated proofreading effort on log scale. Dashed lines mark 95th percentile thresholds. Mouse heavy-tail neurons contribute 14.1% of total edits; fly heavy-tail neurons concentrate 33.3% of edits despite 23.4× lower per-neuron baseline effort, indicating distinct workload concentration patterns. **(B) Operation Type Distribution.** Stacked bars show merge vs. split operation percentages. Mouse exhibits balanced correction (46.3% merge, 53.7% split) indicating comparable under- and over-segmentation errors in initial segmentation. Fly is merge-dominated (74.0% merge, 26.0% split), revealing systematic oversegmentation requiring fragment consolidation. Mean edits per neuron below each bar highlight the 23.4× intensity difference between species.

## 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.

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, 2026. URL https://github.com/seung-lab/cloud-volume.

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, 2021. URL https://doi.org/10.5281/zenodo.5671443.

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.