Intra- and Inter-Cellular Awareness for 3D Neuron Tracking and Segmentation in Large-Scale Connectomics

Author name(s) withheld EMAIL(S) WITHHELD
Address withheld

Editors: Under Review for MIDL 2022

Abstract

Currently, most state-of-the-art pipelines for 3D micro-connectomic reconstruction deal with neuron over-segmentation, agglomeration and subcellular compartment (nuclei, mitochondria, synapses, *etc.*) detection separately. Inspired by the proofreading consensus of experts, we established a paradigm to acquire prior knowledge of cellular characteristics and ultrastructures, as well as discriminate the connectivity of neural circuits simultaneously. Following this novel paradigm, we were keen to bring the **I**ntra- and **I**nter-**C**ellular **A**wareness back when **T**racking and **S**egmenting neurons in connectomics. Our proposed method (II-CATS) utilizes few-shot learning techniques to encode the intermediate neuron representation and its key components, which could significantly impact tracings. We further go beyond the original **e**xpected **r**un **l**ength (ERL) metric by focusing on **b**iological constraints (bERL) and spanning from the **n**ucleus to spines (nERL). With the evaluation of these metrics, we perform typical experiments on multiple electron microscopy datasets on diverse animals and scales. In particular, our proposed method is naturally suitable for tracking neurons that have been identified by staining.

Keywords: connectomics, neuron tracking, neuron segmentation, few-shot learning

1. Introduction

Connectomics, which reconstructs the synaptic level connectivity between neurons in electron microscopy, is paving the way toward learning the structure of the brain. With the advancements in high-resolution and high-throughput imaging with various electron microscopy, every interested cellular boundaries and subcellular organelles (nuclei, mitochondria, *etc.*) could be clearly visible. However, to achieve above, the datasets of brain tissue may now exceed a *peta*byte in size (Shapson-Coe et al., 2021; Bae et al., 2021), for which we need feasible automatic object tracking and segmentation methods.

Current state-of-the-art high-precision neuron reconstruction pipelines have already been able to fully-automatically handle the cubic $10\mu m$ scale *saturate* reconstruction (Lee et al., 2017) and cubic $100\mu m$ *dense* reconstruction (Januszewski et al., 2018). These results are evaluated by voxel-wise membrane **boundaries** and instance-wise neuron **objects**, respectively. Due to the huge computation and labor cost, the large-scale *dense* connectome and *sparse* projectome reconstruction of imperfect datasets still remains challenging (Ngai, 2022). Some **boundary-based** approaches (Bailoni et al., 2020; Lee et al., 2021; Sheridan et al., 2022) use U-Nets (Ronneberger et al., 2015; Çiçek et al., 2016) flexibly, predicting intermediate descriptors for segmentation and running the graph-based agglomeration, as depicted in Figure 1(a). Some **object-based** approaches (Januszewski et al., 2018; Meirovitch et al., 2019; Gonda et al., 2021; Liu et al., 2021) execute single-neuron filling

© 2022 Anonymous.

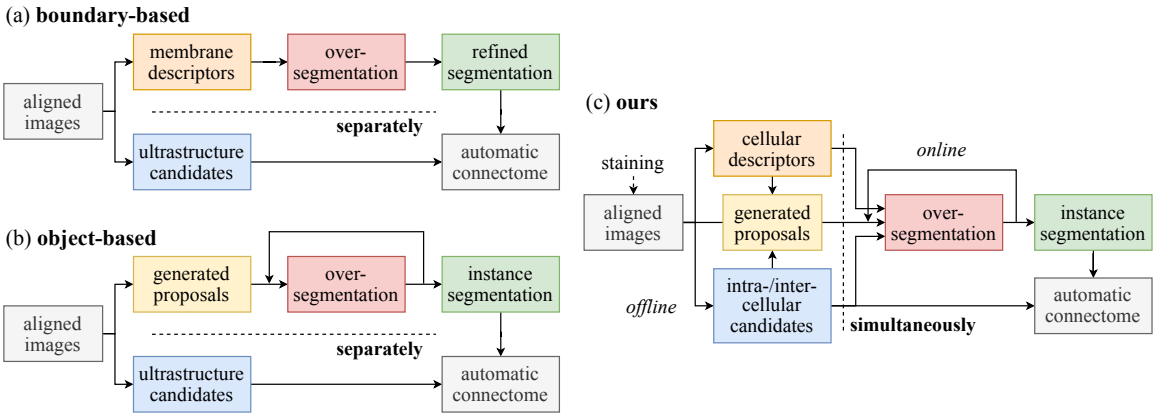

Figure 1: Comparison of current paradigms and our novel paradigm. Both (a) and (b) treat cellular and subcellular structures separately in the process of neural segmentation. In contrast, (c) obtains these information simultaneously.

iteratively, extending the tracing area from seed points, as depicted in Figure 1(b). Other **skeleton-based** approaches (Helmstaedter et al., 2011; Motta et al., 2019; Schmidt et al., 2022) only focus on tracking the neuron skeletons for further connectomic analysis. Both (a) and (b) treat cellular and subcellular structures separately, lacking the procedure of integrating extra biological information.

Aside from these two pipelines, we consider introducing a new paradigm that could take full advantage of both, as depicted in Figure 1(c). The methods related to feature extraction of neurons and their ultrastructures (Lee et al., 2019) have been developed bloomingly. Following our paradigm, one can simultaneously treat various types of cellular descriptors as well as different kinds of intra- and inter-cellular candidates as a priori. With this assistance, one can further handle the over-segmentation from generated instance proposals as a better posteriori estimation. Note that cellular descriptors could be easily generated by existing works (Sheridan et al., 2022), and the so-called candidates could just be predicted *pixel*-level probabilities instead of *instance*-level segmentation.

With the consensus in micro-connectomics, the *voxel*-level accuracy of subcellular mitochondria (Xiao et al., 2018) and synaptic clefts (Dorkenwald et al., 2017; Parag et al., 2018) usually exceeds *instance*-level cellular neuron discrimination. This may be attributed to the homogeneity of appearance among those local compartments. Therefore, these more reliable results can actually be put to good use for more robust neural reconstruction. The ablation study subsection below gives some of the related evidence.

We propose a method to obtain the information from this paradigm in a compact parametric model. To this end, we integrate a differentiable few-shot learner $\mathcal{F}$, which *learns what to learn* (Bhat et al., 2020). In other words, our method enables the segmentation decoder $\mathcal{D}$ to learn a rich internal representation (Figure 2 top-center) instead of directly learning from raw. These crucial aspects of information are encoded by the learner and

predicted by the target model filter ω. The ω is defined as the correlation filters in tracking procedures (Danelljan et al., 2016).

We further determine what the few-shot learner should learn around the structure of neurons. To generate primary proposals, we build a *quasi-dense* tracker $\mathcal{T}$ (Pang et al., 2021) for *box*-level matching on bi-directional local consistency. To make use of multifarious characteristics, we simply implement affinity-based networks with auxiliary learning (Sheridan et al., 2022) for *voxel*-level cellular descriptors and *pixel*-level subcellular candidates (Figure 2 bottom-left).

To evaluate the automated reconstruction method in connectomics, we tried different kinds of metrics to estimate the error from small to large-scale datasets. The *voxel*-level metrics (ARAND, VI (Meilă, 2003)) and *shape*-level metrics (TED (Funke et al., 2017)) require dense instance segmentation ground-truth of neurons, which is impractical for large-scale datasets (Wei et al., 2021). Instead, the expected run length (ERL (Januszewski et al., 2018)) metric measures the expected length of an error-free path along neurons in a volume. This has become commonly used *skeleton*-level metric (also including IED (Berning et al., 2015) and MCM (Sheridan et al., 2022)) for large-scale datasets due to the less associated computational cost than others.

Nuclei are located in the center of neurons. Synapses (and their clefts) extract the inter-cellular connections of neurons (Reilly et al., 2018). Mitochondria (and their unique texture) reflect the intra-cellular stability of neurons. From these perspectives, we think the integrated evaluation of neurons ERL need to embrace a priori of organelles and a structure centered on nucleus. Also, some recent experiments showed similar ERL results (Schmidt et al., 2022). Thus, we extended the original ERL metric by focusing on biological constraints (bERL) and spanning from the nucleus to spines (nERL).

We perform comprehensive experiments on mouse dataset Kasthuri *et al.* (Kasthuri et al., 2015) as well as a song bird dataset J0126 (Januszewski et al., 2018). Our method sets a new state-of-the-art of the ERL-based metrics on certain datasets, comparing to other competitive tracking methods in the field of connectomics. In particular, it is naturally suitable for our method to handle datasets with genetically identified tags, such as *APEX2* (Lam et al., 2015).

2. Methodology

Our proposed method (II-CATS) here has allowed us to track and segment neurons in the procedure of large-scale connectomic reconstruction. It not only can acquire the knowledge of cell structures, but also has the awareness of intra- and inter-cellular compartments.

2.1. Overall Architecture

The integral pipeline is implemented as a multiple object processing framework and a single object end-to-end network architecture, illustrated in Figure 2. Our architecture is composed of the following network modules: the few-shot learner $\mathcal{F}$, feature encoder $\mathcal{E}$, segmentation decoder $\mathcal{D}$, quasi-dense tracker $\mathcal{T}$, internal representation generator $\mathcal{G}$ and internal weight predictor $\mathcal{W}$. Note that every module has the learnable network parameters during the offline training, while the ω is denoted as the target convolutional weights which

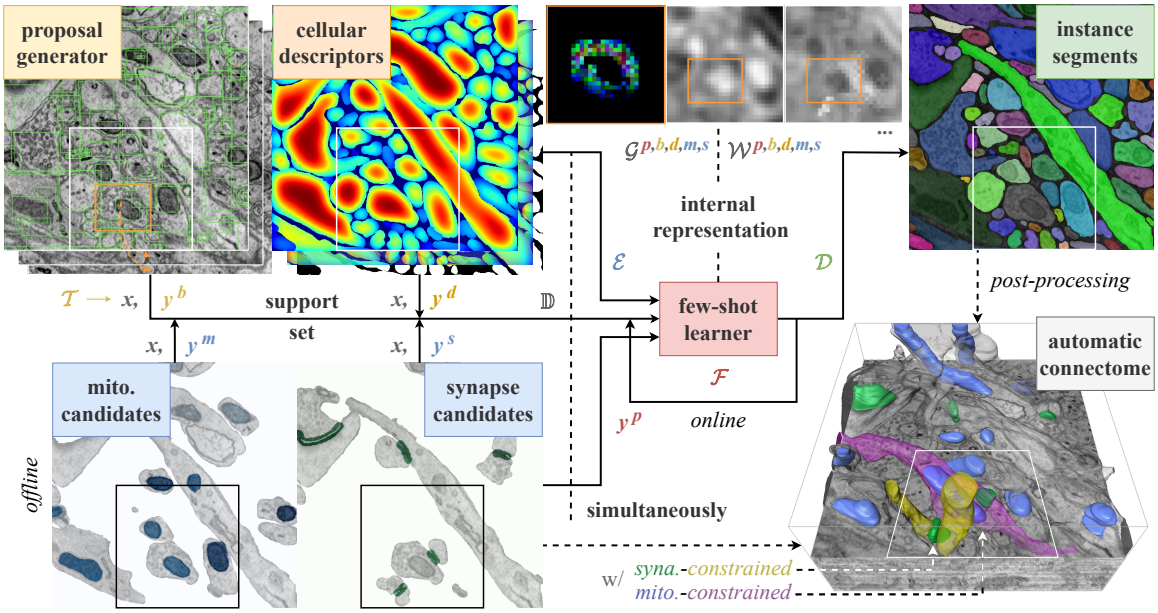

Figure 2: Inputs, outputs and intermediate results of our architecture. They are related to the few-shot learner $\mathcal{F}$, feature encoder $\mathcal{E}$, quasi-dense tracker $\mathcal{T}$, segmentation decoder $\mathcal{D}$, internal representation extractor $\mathcal{G}$ and $\mathcal{W}$. For the volume section x, information $y^{p,b,d,m,s}$ are obtained from previous ***p***redicted segments, generated ***b***ox proposals, cellular ***d***escriptors, ***m***itochondrion, and ***s***ynapse candidates, respectively. (Top-center) Three internal representation of the box area.

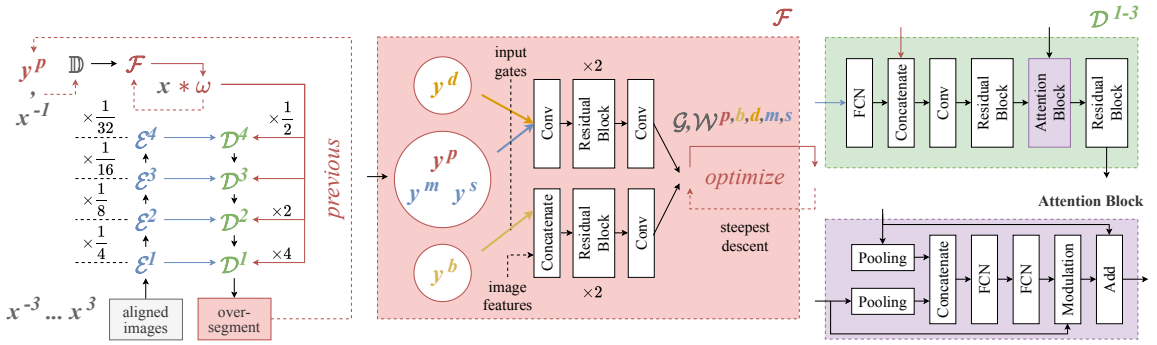

Figure 3: Details about our encoder-decoder architecture, the key few-shot learner $\mathcal{F}$ and segmentation decoder $\mathcal{D}$. (Middle) The inputs of $\mathcal{F}$ are memorized as support set $\mathbb{D}$, in which we treat different information y with different $\mathcal{G}, \mathcal{W}$ networks. For instance, the bounding-box input y^b use the extra image features. To deal with the problem of partially missing the information during tracking, we design the input gates g. (Right) The basic components of $\mathcal{D}$ and its attention block.

predicted by the few-shot learner online. The following subsections detail the key modules in our architecture, illustrated in Figure 3.

2.2. Few-Shot Learner

Different cutting-edge learning strategies have been implemented to microscopy image analysis due to the special characteristics of biomedical images. Especially, learning hyper-representations in few-shot supervised tasks (Franceschi et al., 2018) is suitable for resolving neural connectivity locally in large-scale connectomic reconstruction. Inspired from (Bhat et al., 2020), we would like to learn what to learn at a glance of existing neural candidates.

We push forward our few-shot learner to minimize the squared error between the output of ω convolution to x and the various internal representations $\mathcal{G}(y)$, which weighted by the element-wise attention weights $\mathcal{W}(y)$:

$$\mathcal{L}_{\text{shot}}(\omega) = \frac{1}{2} \sum_{(x,y)\in\mathbb{D}} \|\mathcal{W}(y) \circ (x * \omega - \mathcal{G}(y))\|^2 + \frac{\lambda}{2}\|\omega\|^2, \tag{1}$$

where the $\mathbb{D}$ is the support set for the few-shot online training. Operator $*$ is the convolution and $\circ$ is the *Hadamard* product. The scalar λ is the learnable regularization parameter.

Since the $\mathbb{D}$ receives various support pair (x, y^i) in practice, we introduce a series of gate function g and g^i to control whether we get the information and back-propagate the loss of corresponding ω^i, $i \in \{p, b, d, m, s\}$. In practice, we implement g^i by linear function and g by fully-connected networks, in order to learn the importance of each and project the sum to a proper shape for following decoder $\mathcal{D}$, respectively (Figure 3 middle).

$$x * \omega = g\left(\sum_{i\in\{p,b,d,m,s\}} g^i(x * \omega^i)\right). \tag{2}$$

After that, the differentiable few-shot learner optimizes the target $\omega = \mathcal{F}(\mathbb{D})$, which is $\arg\min_{\omega'} \mathcal{L}_{\text{shot}}(\omega')$. The well-known closed-form solution of this problem requires extensive matrix multiplications. Following (Bhat et al., 2020), we apply an approximate solution of several iterations of steepest descent, which can be expressed as $\omega^{i+1} = \omega^i - \alpha^i\beta^i$ and:

$$\begin{aligned} \alpha^i &= \frac{\|\beta^i\|^2}{\sum_t \|\mathcal{W}(y) \circ (x * \beta^i)\|^2 + \lambda\|\beta^i\|^2}, \\ \beta^i &= \sum_t x *^T \left(\mathcal{W}^2(y) \circ (x * \omega^i - \mathcal{G}(y)\right) + \lambda\omega^i, \end{aligned} \tag{3}$$

where $\beta^i = \nabla\mathcal{L}_{\text{shot}}$ is the gradient and $\alpha^i = \arg\min_{\alpha'} \mathcal{L}_{\text{shot}}(\omega - \alpha'\beta^i)$ is the found step-length on the i-th step. Operator $*^T$ is the transposed convolution.

The implemented few-shot learner module $\mathcal{F}(\mathbb{D}, \omega^0) = \omega^N$ predict satisfactory filter ω by a handful of N iterations. In practice, we would tried various meta learning-based steepest descent methods to decide which one to use, including Newton's method when minimizing the KL-divergence loss (Battiti, 1992).

3. Experiments

We evaluate our method on multiple connectomic datasets across various animals and different scales. The main challenges of 3D reconstruction of connectomic datasets could be briefly described as the large scale of data, the huge variance of different areas (Figure 4 C), and other issues introducing the merged or split errors. We employ the VI metric to distinguish the integrity of segments among different areas, the ERL-based metrics to check tracking errors, and the FLOPs to examine the feasibility in practice.

We use the widely-used *PyTorch Connectomics* (Lin et al., 2021) repository with default configurations from the official document, which could be easy to reproduce related experiments on the ultrastructure segmentation, long-range affinities and watershed. For experiments on Kasthuri *et al.*, we primitively borrow the network parameters trained on SNEMI and fine-tune them. Training and inference codes will be released upon publication.

3.1. Datasets

The open datasets we mainly use are listed following:

SNEMI (Arganda-Carreras et al., 2015) provides a labeled image stack of 6x6x3 μm^3 size in a mouse neocortex for training and validation, with about 6x6x30 nm^3 resolution. As shown in Figure 2, this dataset only contains neurites without cell bodies.

Kasthuri *et al.* (Kasthuri et al., 2015) is the superset of full SNEMI dataset, which is about 10^6 μm^3 and 473 *giga*bytes. It has 1849 sections in total, and the partial manually labeled *Neuroglancer*[1] ground truth.

J0126 (Januszewski et al., 2018) consists of an about 10^6 μm^3 region of a songbird brain in 9x9x20 nm^3. We use 33 labeled stacks for training and one of skeletons (the skeleton of $\mathbf{N}_3$ in Figure 4) for testing, which are kindly provided by (Sheridan et al., 2022).

Joesch *et al.* (Joesch et al., 2016) is partially used for the *Starburst Amacrine* cell (SAC) plexus reconstruction. The region of interest includes 300 slices (93x270x18 μm^3) selected from 680x680x38 μm^3, catalyzed by *APEX2* tags.

3.2. Ablation Studies

Here, we analyze the impact of the various awareness introduced into proposed II-CATS architecture. Our ablation studies are performed on the validation set, *i.e.*, a 6x3x3 μm^3 stack in SNEMI. For simplicity, we use the same pretrained backbone ResNet-50 for our models in Table 1.

Baseline model FFN-a constitutes the open source version FFN (Januszewski et al., 2018), where the FCNs could be simply plugged in the color encoding of transfer mechanism from (Meirovitch et al., 2019) (FFN-3C), or the mitochondrion and synapse information (FFN w/ *mito.*, *syna.*). To achieve this, we put parallel input channels for these subcellular *pixel*-level candidates and slightly increase the number of residual modules and dilate the RoI size in the FFN.

TS model directly explores the *quasi-dense* matching proposal of neurons. Our few-shot learner $\mathcal{F}$ only receives the output of the tracker $\mathcal{T}$, *i.e.*, the support set $\mathbb{D} = \{(x, y^b)\}$.

1. https://storage.googleapis.com/neuroglancer-public-data

CATS model further exploits the cellular description y^d (long-range affinities, shape direction vectors and shape sizes). We also use the differentiable $\mathcal{F}$ to train the underlying representation for target filter ω in an end-to-end manner.

II-CATS models are verified separately (w/ *mito.*, *syna.*) and simultaneously. The information of interposition mitochondrion provides a substantial decrease of VI, which is similar with previous reports on FFN. Besides, the information of synaptic cleft partition contributes to *merged* errors, extending the ERL by additional $y^{m,s}$ samples in $\mathbb{D}$.

The results of bERL and the internal representation (Figure 2 top-center) indicates that we do better job on $\mathcal{L}_{\text{intra}}$ than $\mathcal{L}_{\text{inter}}$, which means we make capital of the w/ *mito.* module. It suggests that we could adding explicit modeling of these losses to the training procedure of $\mathcal{F}$ in the future work.

Table 1: Ablative analysis of cellular, intra- and inter-cellular awareness modules on SNEMI dataset. The unit of all kinds of ERL metrics is *nm*. The top-2 ranking places are in bold.

Method	VI ↓	ERL ↑	bERL ↑
FFN-a	0.938	2771	3313
FFN-3C	1.152	**2925**	3464
FFN w/ *mito.*	0.920	2905	3472
FFN w/ *syna.*	0.991	2766	3249
TS	1.003	2088	2191
CATS	0.917	2267	2506
CATS w/ *mito.*	**0.761**	2332	**3489**
CATS w/ *syna.*	0.890	2827	3025
II-CATS	**0.709**	**3056**	**3690**

Table 2: Comparison on the Kasthuri *et al.* and J0126 dataset.

	Kasthuri *et al.*		J0126 ($\mathbf{N_3}$)	
Method	VI ↓	tERL ↑	tERL ↑	nERL ↑
FFN-a	1.898	**8964**	**16886**	8209
CATS	1.835	8561	14942	**9361**
II-CATS	**1.725**	8819	-	-

3.3. Comparison

In this subsection, we extend our comparison from the small-scale SNEMI to rather large-scale datasets, which could be more meaningful in practice, shown in Table 2. As for the decreased VI on Kasthuri *et al.* dataset, II-CATS successfully obtains the predicted subcellular results, which only trained on small-scale SNEMI dataset. As for the two ERL results on J0126, CATS also reaches comparable results to the original FFN method.

Table 3: Comparison on the Joesch *et al.* staining dataset.

Method	VI ↓	tERL ↑	FLOPs ↓
Watershed	1.505	3264	$\sim 10^{17}$
FFN-a	0.922	5931	$> 10^{19}$
CATS	**0.820**	**5956**	$\sim 10^{18}$

The results of nERL metric considered the scale factor among skeletons, which should be important for whole cell proofreading. The visualization on a part of J0126 dataset (Figure 4) demonstrates that our method almost gets rid of small fragments and tunnels which the FFN results have.

In particular, we also do our experiments on the target cells possessed typical staining characteristics, illustrated in Table 3. Due to the interference between the darkened areas and original membrane boundaries, methods tracking from proposals (Figure 1b) usually excel the other methods segmenting from membrane descriptors (Figure 1a). However, our proposed method outperform both of them on these cells of Joesch *et al.* dataset. We also estimate qualitative computational complexity analysis, which infers that our method (CATS) is at least 10 times faster compared to FFN-a.

Note that the nERL metric need somas or nuclei, which is unavailable for original SNEMI stack (Table 1). We also do not have the ground truth of mitochondria and synapses in Kasthuri *et al.*, J0126 and Joesch *et al.*, so we can not test the bERL metric in these datasets. Next step, we are looking forward to building our own dataset, containing all things we will need.

4. Conclusion

We present II-CATS, a series of connectomic few-shot learning methods based on our proposed novel paradigm. First and foremost, this new way of thinking guides us to exploit the intra- and inter-cellular priori for a more robust neuron connectomic posteriori. After that flash of insight, we implement a multiple object tracker with *quasi-dense* anchors and a single object decoder with *learning to learn* parameters in order to achieve that paradigm in practice. Further more, we use the VI and ERL metric along with the biologically-implied bERL and nERL metric to evaluate our methods in a more meaningful situation. To sum up, our methods have the advantages on small, large and staining datasets, meanwhile have the potential to generalize to more complex 3D reconstruction.

Acknowledgments

Acknowledgments withheld.

References

Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireşan, Alessandro Giusti, Luca M Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, 9:142, 2015.

J Alexander Bae, Mahaly Baptiste, Agnes L Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J Bumbarger, Manuel A Castro, Brendan Celii, Erick Cobos, Forrest Collman, et al. Functional connectomics spanning multiple areas of mouse visual cortex, 2021.

Alberto Bailoni, Constantin Pape, Steffen Wolf, Anna Kreshuk, and Fred A Hamprecht. Proposal-free volumetric instance segmentation from latent single-instance masks. In *DAGM German Conference on Pattern Recognition*, pages 331–344. Springer, 2020.

Roberto Battiti. First-and second-order methods for learning: between steepest descent and newton's method. *Neural computation*, 4(2):141–166, 1992.

Manuel Berning, Kevin M Boergens, and Moritz Helmstaedter. Segem: efficient image analysis for high-resolution connectomics. *Neuron*, 87(6):1193–1206, 2015.

Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *European Conference on Computer Vision*, pages 777–794. Springer, 2020.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European conference on computer vision*, pages 472–488. Springer, 2016.

Sven Dorkenwald, Philipp J Schubert, Marius F Killinger, Gregor Urban, Shawn Mikula, Fabian Svara, and Joergen Kornfeld. Automated synaptic connectivity inference for volume electron microscopy. *Nature methods*, 14(4):435–442, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

Jan Funke, Jonas Klein, Francesc Moreno-Noguer, Albert Cardona, and Matthew Cook. Ted: A tolerant edit distance for segmentation evaluation. *Methods*, 115:119–127, 2017.

Felix Gonda, Donglai Wei, and Hanspeter Pfister. Consistent recurrent neural networks for 3d neuron segmentation. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1012–1016. IEEE, 2021.

Moritz Helmstaedter, Kevin L Briggman, and Winfried Denk. High-accuracy neurite reconstruction for high-throughput neuroanatomy. *Nature neuroscience*, 14(8):1081–1088, 2011.

Michał Januszewski, Jörgen Kornfeld, Peter H Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. High-precision automated reconstruction of neurons with flood-filling networks. *Nature methods*, 15(8): 605–610, 2018.

Maximilian Joesch, David Mankus, Masahito Yamagata, Ali Shahbazi, Richard Schalek, Adi Suissa-Peleg, Markus Meister, Jeff W Lichtman, Walter J Scheirer, and Joshua R Sanes. Reconstruction of genetically identified neurons imaged by serial-section electron microscopy. *elife*, 5:e15015, 2016.

Narayanan Kasthuri, Kenneth Jeffrey Hayworth, Daniel Raimund Berger, Richard Lee Schalek, José Angel Conchello, Seymour Knowles-Barley, Dongil Lee, Amelio Vázquez-Reina, Verena Kaynig, Thouis Raymond Jones, et al. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015.

Stephanie S Lam, Jeffrey D Martell, Kimberli J Kamer, Thomas J Deerinck, Mark H Ellisman, Vamsi K Mootha, and Alice Y Ting. Directed evolution of apex2 for electron microscopy and proximity labeling. *Nature methods*, 12(1):51–54, 2015.

Kisuk Lee, Jonathan Zung, Peter Li, Viren Jain, and H Sebastian Seung. Superhuman accuracy on the snemi3d connectomics challenge, 2017.

Kisuk Lee, Nicholas Turner, Thomas Macrina, Jingpeng Wu, Ran Lu, and H Sebastian Seung. Convolutional nets for reconstructing neural circuits from brain images acquired by serial section electron microscopy. *Current opinion in neurobiology*, 55:188–198, 2019.

Kisuk Lee, Ran Lu, Kyle Luther, and H Sebastian Seung. Learning and segmenting dense voxel embeddings for 3d neuron reconstruction. *IEEE Transactions on Medical Imaging*, 40(12):3801–3811, 2021.

Zudi Lin, Donglai Wei, Jeff Lichtman, and Hanspeter Pfister. Pytorch connectomics: a scalable and flexible segmentation framework for em connectomics, 2021.

Jing Liu, Bei Hong, Xi Chen, Qiwei Xie, Yuanyan Tang, and Hua Han. An effective ai integrated system for neuron tracing on anisotropic electron microscopy volume. *Biomedical Signal Processing and Control*, 69:102829, 2021.

Marina Meilă. Comparing clusterings by the variation of information. In *Learning theory and kernel machines*, pages 173–187. Springer, 2003.

Yaron Meirovitch, Lu Mi, Hayk Saribekyan, Alexander Matveev, David Rolnick, and Nir Shavit. Cross-classification clustering: An efficient multi-object tracking technique for 3-d instance segmentation in connectomics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8425–8435, 2019.

Alessandro Motta, Manuel Berning, Kevin M Boergens, Benedikt Staffler, Marcel Beining, Sahil Loomba, Philipp Hennig, Heiko Wissler, and Moritz Helmstaedter. Dense connectomic reconstruction in layer 4 of the somatosensory cortex. *Science*, 366(6469), 2019.

John Ngai. Brain 2.0: Transforming neuroscience. *Cell*, 185(1):4–8, 2022.

Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 164–173, 2021.

Toufiq Parag, Daniel Berger, Lee Kamentsky, Benedikt Staffler, Donglai Wei, Moritz Helmstaedter, Jeff W Lichtman, and Hanspeter Pfister. Detecting synapse location and connectivity by signed proximity estimation and pruning with deep nets. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

Elizabeth P Reilly, Jeffrey S Garretson, William R Gray Roncal, Dean M Kleissas, Brock A Wester, Mark A Chevillet, and Matthew J Roos. Neural reconstruction integrity: A metric for assessing the connectivity accuracy of reconstructed neural networks. *Frontiers in neuroinformatics*, 12:74, 2018.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Martin Schmidt, Alessandro Motta, Meike Sievers, and Moritz Helmstaedter. Roboem: automated 3d flight tracing for synaptic-resolution connectomics. *bioRxiv*, 2022.

Alexander Shapson-Coe, Michał Januszewski, Daniel R Berger, Art Pope, Yuelong Wu, Tim Blakely, Richard L Schalek, Peter Li, Shuohong Wang, Jeremy Maitin-Shepard, et al. A connectomic study of a petascale fragment of human cerebral cortex, 2021.

Arlo Sheridan, Tri M Nguyen, Diptodip Deb, Wei-Chung Allen Lee, Stephan Saalfeld, Srinivas C Turaga, Uri Manor, and Jan Funke. Local shape descriptors for neuron segmentation. *Nature Methods*, pages 1–9, 2022.

Donglai Wei, Kisuk Lee, Hanyu Li, Ran Lu, J Alexander Bae, Zequan Liu, Lifu Zhang, Márcia dos Santos, Zudi Lin, Thomas Uram, et al. Axonem dataset: 3d axon instance segmentation of brain cortical regions. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 175–185. Springer, 2021.

Chi Xiao, Xi Chen, Weifu Li, Linlin Li, Lu Wang, Qiwei Xie, and Hua Han. Automatic mitochondria segmentation for em data using a 3d supervised convolutional network. *Frontiers in neuroanatomy*, 12:92, 2018.

Appendix A. Qualitative Results

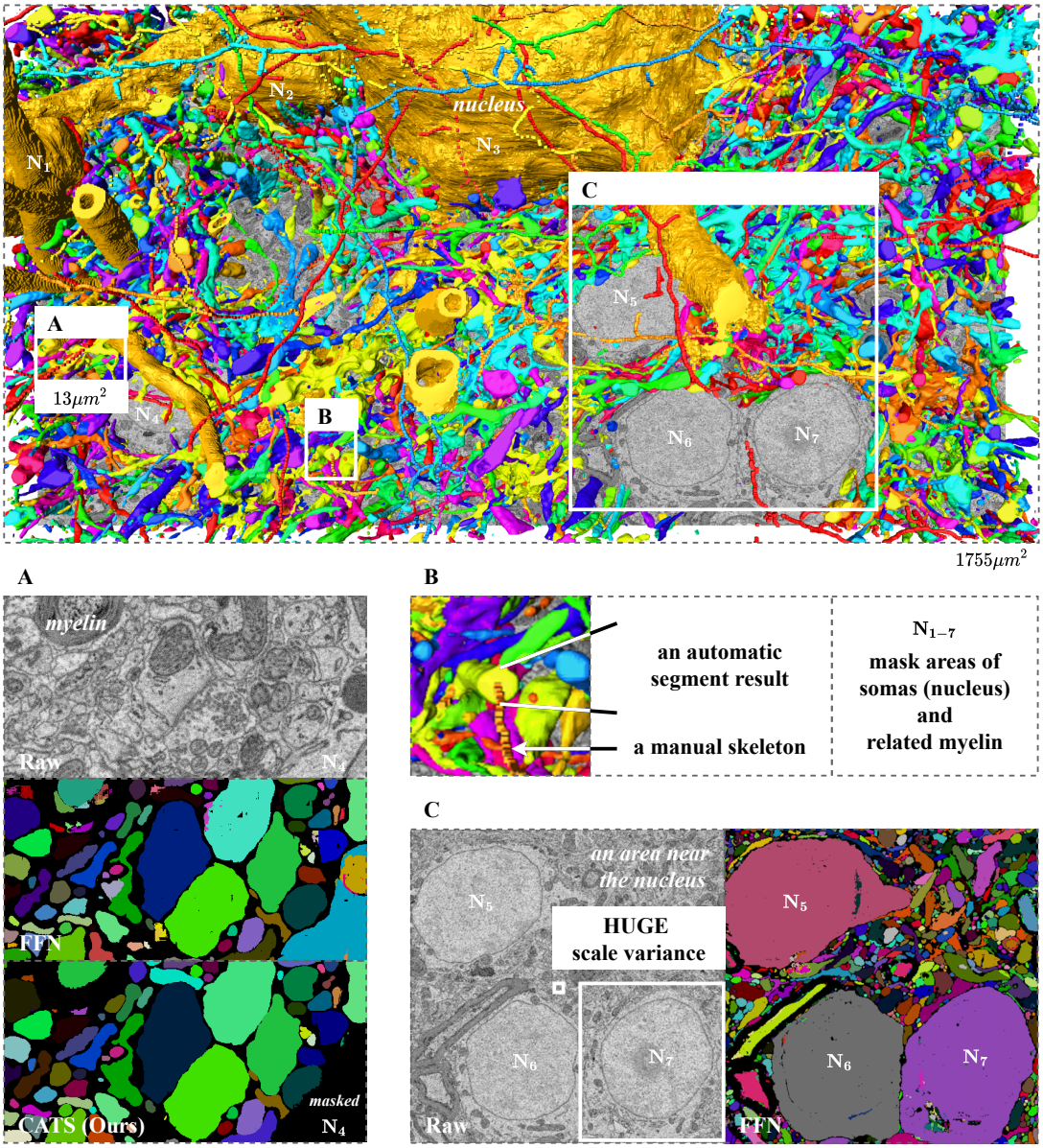

Figure 4: Visualizations of a part of dataset J0126. (A) Qualitative comparison between FFN and CATS (Ours) from an about $13\mu m^2$ patch of $1755\mu m^2$. (B) We also render $4\mu m$ automatic prediction of segments as well as $14\mu m$ somas and myelin of $\mathbf{N_{1-3}}$, and manual skeletons upon the section, demonstrating the results of our results. (C) Nearby the nucleus, other tracking method generates split errors due to the huge scale variance, which could be measured by our nERL metric.

Appendix B. Architecture Details

B.1. Quasi-Dense Tracker

We take the popular region proposal network (RPN) with feature pyramid network (FPN) as the baseline of our tracker. The RPN generates region of interests (RoIs) for detection, while the FPN exploits lateral connections to build the top-down feature pyramid and tackles the scale-variance problem.

On this basis, we use RoIs to learn the instance similarity with quasi-dense matching. As shown in (Pang et al., 2021), the quasi-dense means only considering the potential neuron candidates at informative regions. Given the key section for training, we randomly select a reference section from neighborhood. Assume the object instances on the key section are denoted as $\mathbf{o}$. An instance on the reference section is defined as positive $\mathbf{o}^+$ if two RoIs with IoU higher than 0.7 are associated with the same instance, and negative $\mathbf{o}^-$ otherwise.

As for the architecture after RPN, an extra lightweight embedding head is add in parallel with the original bounding box head. We can use the non-parametric softmax with cross-entropy to optimize the feature embedding $\mathcal{E}(\mathbf{o})$:

$$\mathcal{L}_{\text{quasi}}(\mathbf{o}) = \log\left(1 + \sum_{\mathbf{o}^+}\sum_{\mathbf{o}^-} e^{\mathcal{E}(\mathbf{o})\circ\mathcal{E}(\mathbf{o}^-)-\mathcal{E}(\mathbf{o})\circ\mathcal{E}(\mathbf{o}^+)}\right). \tag{4}$$

The key-reference matched neuron candidates (Figure 2 top-left) is generated by trained $\mathcal{T}$. And these will be used for a part of $(x, y^b) \in \mathbb{D}$ with bounding boxes to further train the few-shot learner $\mathcal{F}$.

B.2. Segmentation Decoder

We take the mirrored 3D residual networks as the segmentation decoder, which is similar to U-Net. The decoder receives the output of the target ω convolution along with features from encoder $\mathcal{E}$ to predict accurate segmentation masks. We use ResNet-50 to balance the simplicity and accuracy. The encoder could be divided as four residual blocks $\mathcal{E}^1 \cdots \mathcal{E}^4$, so that the corresponding output feeds to $\mathcal{D}^1 \cdots \mathcal{D}^4$ (Figure 3 left).

For each decoder block $\mathcal{D}^i$, we designed three parts for feature projection and fusion. First, the encoding features need to project to a lower-dimensional representation, which could be concatenated with the ω representation. Second, the concatenated features are processed by convolutional layers followed by a small residual block. Third, the resulting features are then merged with features from deeper decoder $\mathcal{D}^{i+1}$ by a channel attention block. Note that the interpolation ratio of ω representation for each $\mathcal{D}^i$ is equal to 2^{3-i} (Figure 3 right).

The final results for each proposal will be agglomerated by a post processing step (Meirovitch et al., 2019) in order to generate the automatic connectome.

Appendix C. Metrics

The evolution of large-scale connectomic metrics matters. It may be risky that only extrapolating from the benchmark evaluation of limited data, instead the large-scale datasets are robust to estimate real-word accuracy of the neural circuit reconstruction (Lee

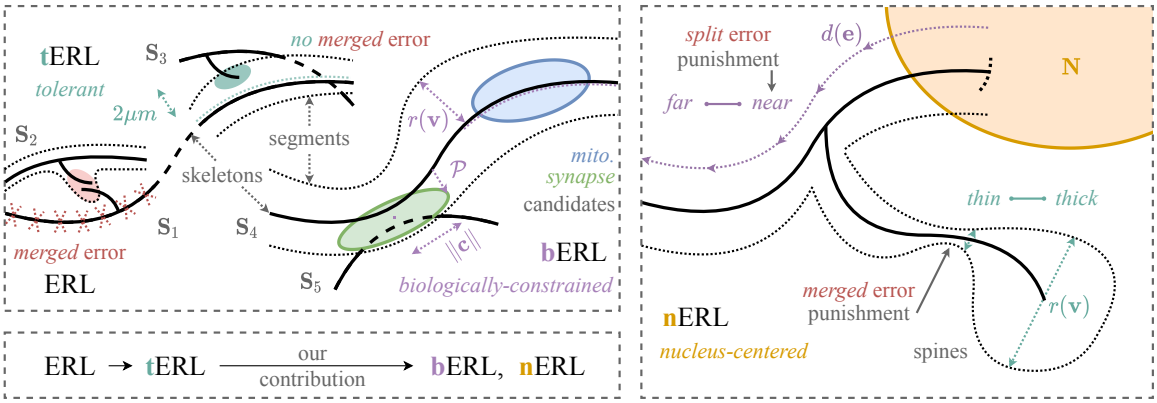

Figure 5: Sketches of the ERL, tERL, bERL and nERL metric. These ERL-based metrics are evaluated by proofread skeletons (solid line) and predicted segments (dotted envelope line). (Left) Our proposed bERL punishes errors nearby mitochondria (blue) and synaptic clefts (green). (Right) The soma or nucleus (yellow) and its neurites indicate the different error punishment of nERL from near to far and from thin to thick.

et al., 2017). The original ERL-based metrics should have be suitable for large-scale due to they only need manual skeletons (instead of segments), but they are missing key biological constrains (for instance, the nuclei information only occurs in large-scale volumes). **The insight into subcellular structures matters.** Nuclei are located in the center of neurons. Synapses (and their clefts) extract the inter-cellular connections of neurons. Mitochondria (and their unique texture) reflect the intra-cellular stability of neurons. Other structures such as myelin and glia may also helpful (Dorkenwald et al., 2017). We think the integrated evaluation of neurons need to embrace a priori of organelles and a structure centered on nucleus.

Considering the proofread or generated skeletons $\mathbf{S} \in \mathbb{S}$ (ground truth) and predicted segments $\mathbf{L} \in \mathbb{L}$ (labeled supervoxel), we define the mapping function $m(\mathbf{e})$ for any edge $\mathbf{e} = \{\mathbf{v}, \mathbf{v}'\} \in \mathbf{S}$, and the length function $f(\mathbf{S}, \mathbf{L})$ for the *correct* paths at the intersection of $\mathbf{S}$ and $\mathbf{L}$. Give:

$$m(\mathbf{e}) = \begin{cases} 0, & \mathbf{v} \in \mathbf{L_1}, \mathbf{v}' \in \mathbf{L_2} \\ \|\mathbf{e}\|, & \mathbf{v}, \mathbf{v}' \in \mathbf{L} \end{cases}, \tag{5}$$

$$f(\mathbf{S}, \mathbf{L}) = \begin{cases} 0, \exists \mathbf{v_1}, \mathbf{v_2} \in \mathbf{L}, \mathbf{v_1} \in \mathbf{e_1} \in \mathbf{S_1}, \mathbf{v_2} \in \mathbf{e_2} \in \mathbf{S_2} \\ \sum_{\mathbf{e} \in \mathbf{S}, \mathbf{v} \in \mathbf{L}} m(\mathbf{e}), \quad \text{otherwise} \end{cases}, \tag{6}$$

where each skeleton $\mathbf{S}$ is defined as a set of edges, each segment $\mathbf{L}$ is defined as a set of vertices, and $\|\mathbf{e}\|$ denotes the length of edge. The $m(\mathbf{e}) = 0$ shows the *split* paths, and the $f(\mathbf{S}, \mathbf{L}) = 0$ shows the *merged* paths. The original ERL (Januszewski et al., 2018) metrics are:

$$\mathrm{ERL}(\mathbf{S}) = \frac{1}{\|\mathbf{S}\|} \sum_{\mathbf{L}\in\mathbb{L}} f(\mathbf{S},\mathbf{L})^2, \tag{7}$$

$$\mathrm{ERL}(\mathbb{S}) = \frac{1}{\sum_{\mathbf{S}\in\mathbb{S}} \|\mathbf{S}\|} \sum_{\mathbf{S}\in\mathbb{S}} \sum_{\mathbf{L}\in\mathbb{L}} f(\mathbf{S},\mathbf{L})^2, \tag{8}$$

where $\|\mathbf{S}\| = \sum_{\mathbf{e}\in\mathbf{S}} \|\mathbf{e}\|$ denotes the length of skeleton.

C.1. tERL

However, the original ERL metric has a few drawbacks. For example, it emphasizes merge errors disproportionally (Sheridan et al., 2022), and may assign unexpected zero run length from skeletonization artifacts. Wei *et al.* (Wei et al., 2021), therefore, designed a tolerance threshold τ of *merged* vertices (around 2 μm in length) for the more robust tERL, which could substitute $f_{\mathrm{t}}(\mathbf{S},\mathbf{L})$ for $f(\mathbf{S},\mathbf{L})$ in the original ERL:

$$f_{\mathrm{t}}(\mathbf{S},\mathbf{L}) = \begin{cases} 0, & |\mathbf{T}| > \tau \\ \sum_{\mathbf{e}\in\mathbf{S},\mathbf{v}\in\mathbf{L}} m(\mathbf{e}), & \text{otherwise} \end{cases}, \tag{9}$$

where $\mathbf{T} = \{(\mathbf{v_1},\mathbf{v_2}) \mid \mathbf{v_1},\mathbf{v_2} \in \mathbf{L}, \mathbf{v_1} \in \mathbf{e_1} \in \mathbf{S_1}, \mathbf{v_2} \in \mathbf{e_2} \in \mathbf{S_2}\}$, which is the vertex pair of *merged* paths.

The following two subsections show some of our efforts, which have been dedicated to extend the original ERL metric to the biological implications (Figure 5).

C.2. bERL

We design the biologically-constrained expected run length (bERL), which could substitute $m_{\mathrm{b}}(\mathbf{e})$ and $f_{\mathrm{b}}(\mathbf{S},\mathbf{L})$:

$$m_{\mathrm{b}}(\mathbf{e}) = \begin{cases} -\epsilon, & \mathbf{v} \in \mathbf{V}_{mito}, \mathbf{v} \in \mathbf{L_1}, \mathbf{v}' \in \mathbf{L_2} \\ \|\mathbf{e}\|, & \mathbf{v},\mathbf{v}' \in \mathbf{L} \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

$$f_{\mathrm{b}}(\mathbf{S},\mathbf{L}) = \begin{cases} 0, & |\mathbf{T}| > \tau \quad \text{or} \quad \exists \mathbf{v} \in \mathbf{T}, \mathbf{v} \in \mathbf{V}_{syna} \\ \sum_{\mathbf{e}\in\mathbf{S},\mathbf{v}\in\mathbf{L}} m_{\mathrm{b}}(\mathbf{e}), & \text{otherwise} \end{cases}, \tag{11}$$

where the constraints of $\mathbf{V}_{mito}$ and $\mathbf{V}_{syna}$ are shown as:

$$\begin{aligned} &\forall \mathbf{v} \in \mathbf{V}_{mito}, \exists \mathbf{u} \in \mathbf{U}_{mito}, \|\mathbf{e}\| := \|\mathbf{v} - \mathbf{v}'\| \\ &\text{s.t.} \quad \|\mathcal{P}_{\mathbf{e}}(\mathbf{u}) - \mathbf{v}\| \le \|\mathbf{e}\| + r(\mathbf{v}) \\ &\qquad \|\mathcal{P}_{\mathbf{e}}(\mathbf{u}) - \mathbf{u}\| \le r(\mathbf{v}) \end{aligned} \tag{12}$$

$$\begin{aligned} &\forall \mathbf{v} \in \mathbf{V}_{syna}, \exists \mathbf{u} \in \mathbf{U}_{syna}, \|\mathbf{c}\| := \max_{\mathbf{u}} \left\| \mathbf{u} - \frac{\sum \mathbf{u}}{|\mathbf{U}_{syna}|} \right\| \\ &\text{s.t.} \quad \|\mathcal{P}_{\mathbf{c}}(\mathbf{v}) - \mathbf{v}\| \le \|\mathbf{c}\| + r(\mathbf{v}) \\ &\qquad \|\mathcal{P}_{\mathbf{c}}(\mathbf{v}) - \mathbf{u}\| \le r(\mathbf{v}) \end{aligned} \tag{13}$$

where $\mathbf{U}_{mito}$ and $\mathbf{U}_{syna}$ are existing mitochondrion and synapse candidates. Operator $\mathcal{P}_{\mathbf{i}}$ is the projection into $\mathbf{i}$. We define the radius function $r(\mathbf{v})$ to describe average radius of vertices, and $\|\mathbf{c}\|$ to describe radius of fitted synaptic clefts. The fitted cleft could be abstracted to a center point and a surface with a radius.

The bERL metric could be quickly deployed when one have the proofread mitochondrion or synaptic cleft instances, which are far easier to acquire than all of the neuron instances. Moreover, this metric screens subcellular related fragments and checks them out, which will be helpful for optimizing the areas consisting biological information.

C.3. nERL

We further design the nucleus-centered expected run length (nERL), which could substitute $m_{\mathrm{n}}(\mathbf{e})$ and $f_{\mathrm{n}}(\mathbf{S}, \mathbf{L})$. We define the logarithmic distance function $d(\mathbf{e})$ to measure the crow flight between $\mathbf{e} \in \mathbf{S}$ and the precomputed nucleus center of $\mathbf{S}$. The $r(\mathbf{v})$ and $d(\mathbf{e})$ share the same unit.

$$m_{\mathrm{n}}(\mathbf{e}) = \begin{cases} 0, & \mathbf{v} \in \mathbf{L_1}, \mathbf{v}' \in \mathbf{L_2} \\ r(\mathbf{v}) \cdot \|\mathbf{e}\|, & \mathbf{v}, \mathbf{v}' \in \mathbf{L} \end{cases}, \tag{14}$$

$$f_{\mathrm{n}}(\mathbf{S}, \mathbf{L}) = \begin{cases} 0, & |\mathbf{T}| > \tau \\ \sum_{\mathbf{e} \in \mathbf{S}, \mathbf{v} \in \mathbf{L}} \frac{m_{\mathrm{n}}(\mathbf{e})}{d(\mathbf{e})}, & \text{otherwise} \end{cases}, \tag{15}$$

Beside Figure 5 the sketches of our metrics, we also give the real examples from SNEMI dataset (same as Figure 2). The extra figures below are viewed in the *Neuroglancer*[2].

The two Figure 7 and 8 illustrates one of the real situation of bERL in Figure 5.

In addition, we would like to describe the equations of our proposed metrics step by step in detail.

In Equation 10, based on the original $m(\mathbf{e})$, a more severe penalty is given to the vertex $\mathbf{v}$ in $\mathbf{V}_{mito}$. The penalty term ϵ can be given according to the reliability of the mitochondrial segmentation results in the dataset.

In Equation 11, except for the case where the penalty is greater than the tolerance threshold τ, the vertex $\mathbf{v}$ in $\mathbf{V}_{syna}$ is directly penalized. Such drastic measures can significantly reflect error conditions around the synapse.

Equation 12 and Equation 13 present a correspondence, and both show how we use the projection operator to achieve the above idea. It should be noted that we abstract the synaptic cleft as a kind of disk by fitting, and save it as three parameters of position $\mathbf{c}$, direction $\bar{\mathbf{c}}$, and radius $\|\mathbf{c}\|$, which is convenient for the subsequent projection process.

The significance of bERL is that for the areas that biologists pay more attention to, such as the location of these important subcellular structures, we need to extract relevant sites through automated algorithms and further optimize them.

In Equation 14, we achieve a strategy of increasing the *merged* error penalty for thin places (like spines) by introducing the radius $r(\mathbf{v})$ of the skeleton.

In Equation 15, we increase the *split* error penalty for objects that are too close by introducing $d(\mathbf{e})$. It is important to note that the reason this function varies logarithmically

2. https://neuroglancer-demo.appspot.com

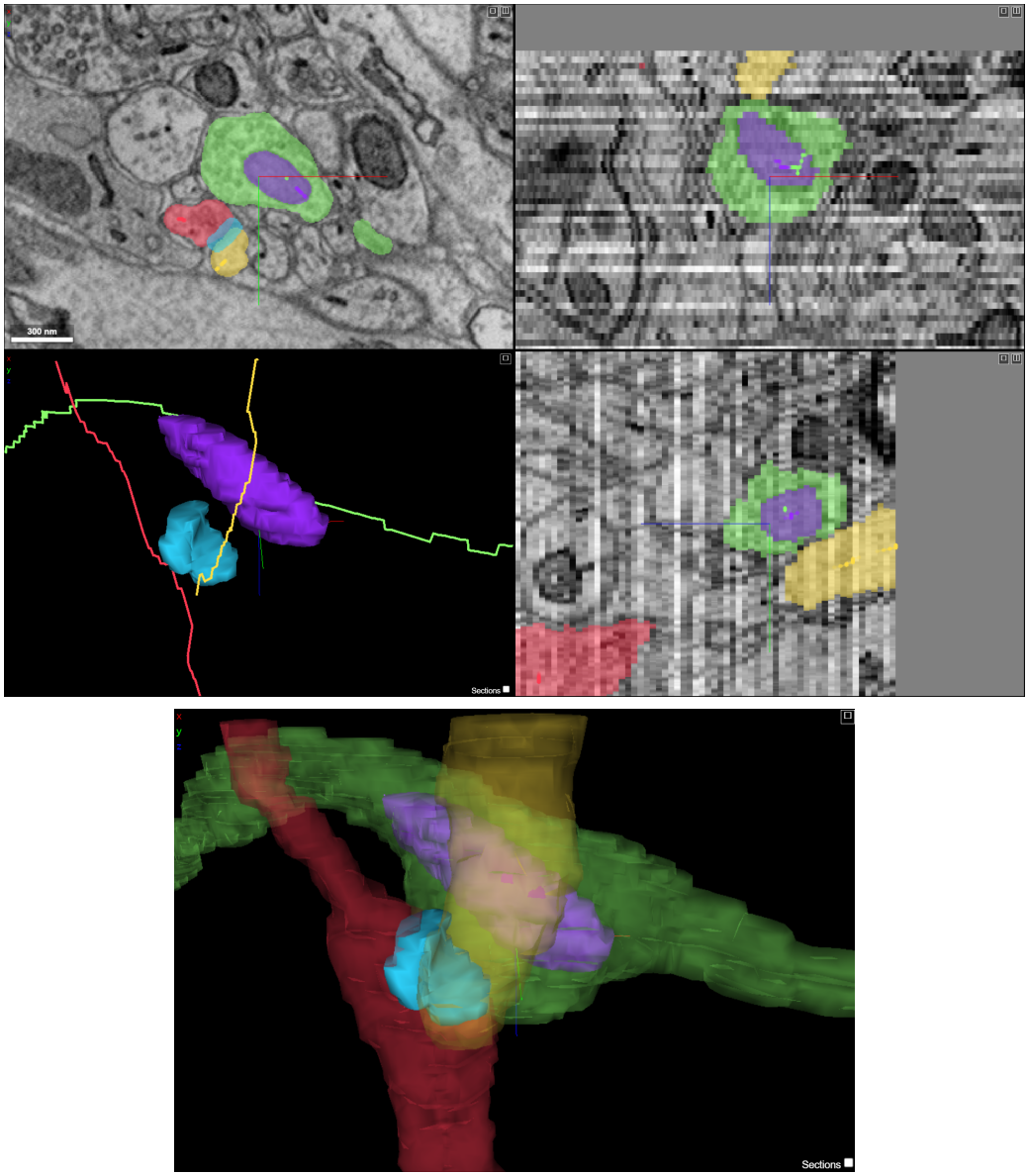

Figure 6: (Top) Three views and 3D rendering of the relevant area in Figure 2 bottom-right. Three lines indicates the skeleton of neurites. Purple represents a mitochondrion, and blue represents a synaptic cleft. (Bottom) 3D rendering of the relevant area in Figure 2 bottom-right. Purple represents a mitochondrion, and blue represents a synaptic cleft. Neurites displayed in translucent form.

is that we do not want to cause objects that are too far away to completely lose their contribution to the ERL.

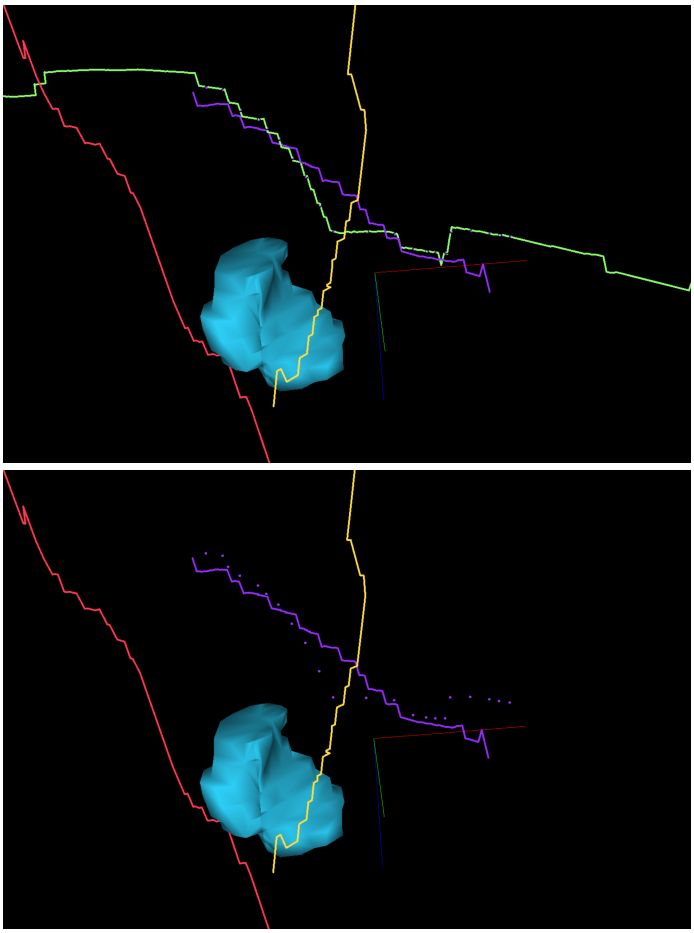

Figure 7: The purple line shows the skeleton of this mitochondrion, and the points of them will be projected to corresponding neurite ($\mathcal{P}_{\mathbf{e}}(\mathbf{u})$ in Equation 12, see the purple points).

The importance of nERL is that it gives us a perspective to evaluate the tracking IoU of different RoI location of skeletons on a larger scale, which cannot be reflected in previous small-scale datasets.

Appendix D. Implementation Details

We try to not use any meaningless trick for network architecture design. Some of the subsequent network settings are based on previous researches (Januszewski et al., 2018; Bhat et al., 2020; Pang et al., 2021).

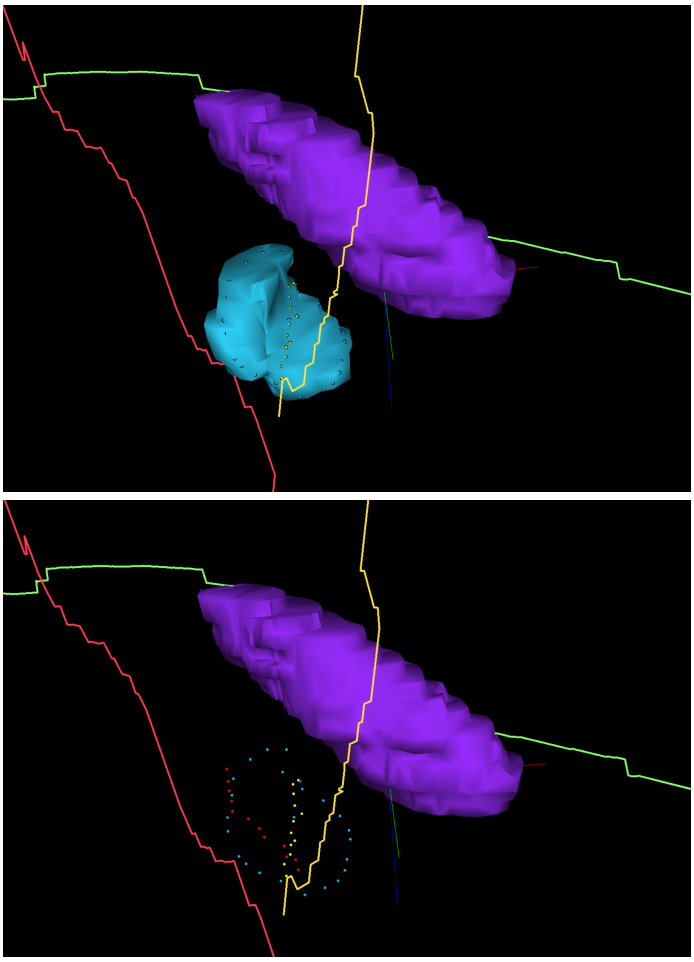

Figure 8: The blue points shows the counter and the center point of this synaptic cleft, which are the results of fitting processing ($\mathbf{c}$ in Equation 13). One neurite (yellow line) skeleton projects to the synaptic cleft ($\mathcal{P}_{\mathbf{c}}(\mathbf{v})$ in Equation 13, see the yellow points), another neurite (red line) skeleton also projects to the synaptic cleft (see the red points), and they construct a connection of neurites in the connectome.

D.1. FFN-a

We used the open source on the GitHub repository[3] as the FFN-a method without consensus and agglomeration step. The results of this code appear in Table 1, 2, 3.

For J0126 dataset, we use FFN ROI size 33x33x17, RoI step size 8x8x4, 9 of residual modules, initial RoI fill value 0.05, initial RoI seed value 0.95, RoI movement threshold 0.9, image irregularity detection (tissue classification, patch-wise cross-correlation), consensus

3. https://github.com/google/ffn

input segmentations of 36x36x40, 18x18x20, 18x18x20, 9x9x20, 9x9x20, and agglomeration criteria of $f_* \geq 0.6$, $\frac{d_A}{N_A} < 0.02$, $\frac{d_B}{N_B} < 0.02$, $J_{AB} \geq 0.8$.

For SNEMI dataset, we use CLAHE and mirror padding for data pre-processing, FFN ROI size 33x33x17, RoI step size 8x8x4, 9 of residual modules, initial RoI fill value 0.5, initial RoI seed value 0.95, RoI movement threshold 0.6, consensus input segmentations of 6x6x30, and agglomeration criteria of $f_* \geq 0.5$, $\frac{d_A}{N_A} < 0.04$, $\frac{d_B}{N_B} < 0.04$, $J_{AB} \geq 0.5$.

Note that the qualitative result of Figure 4 A is a patch in FFN results from Funke *et al.*, which are kindly provided by (Sheridan et al., 2022).

D.2. FFN-3C

For the so-called color encoding mechanism we use for FFN, it named cross-classification clustering (Meirovitch et al., 2019). Their goal is to extend a single-object classification from one image to the next so as to simultaneously classify pixels for an a priori unknown set of object labels.

D.3. FFN w/ *mito.*, *syna.*

For add another channel for *mito.* or *syna.* on SNEMI dataset, we also use CLAHE and mirror padding for raw images along with the voxel-level candidate segmentations. The FFN then has the ROI size 65x65x33, RoI step size 8x8x4, 12 of residual modules, initial RoI fill value 0.5, initial RoI seed value 0.95, RoI movement threshold 0.6, consensus input segmentations of 6x6x30.

D.4. U-Nets

We not only use the long-range affinities, but also implement some of the LSD (Sheridan et al., 2022). To be specific, we find the layer 0 to 2 and 9 are special for our tasks in their code[4] of generating 3D neuron LSD.

The layer 0 to 2 denote the Gaussian point offset to shape center and layer 9 denotes the shape size. The LSD is $lsd^y(v) = (\hat{s}(v), \hat{m}(v) - v, \hat{c}(v))$, and we use the first 2 components to formulate an auxiliary learning task that complements the prediction of affinities.

We use difference settings such as SNEMI-Affinity-ResNet, SNEMI- Affinity-UNet, SNEMI-Affinity-UNet-LR, SNEMI-Affinity-UNet-MER, Zebrafinch-Affinity-UNet, Zebrafinch-Affinity-UNet-MER, Distance-Transform-Quantized, and Multi-class-Semantic-Seg. These are freely provided by Lin *et al.* on the documents of *Pytorch Connectomics*[5].

4. https://github.com/funkelab/lsd
5. https://github.com/zudi-lin/pytorch_connectomics