frontiers

mEMbrain: an interactive deep learning MATLAB tool for connectomic segmentation on commodity desktops

Elisa C. Pavarino [1,+,*], **Emma Yang** [2,+,*], **Nagaraju Dhanyasi** [2], **Mona Wang** [3,4], **Flavie Bidel** [5], **Xiaotang Lu** [2], **Fuming Yang** [2], **Mukesh BR** [2], **Brandon Drescher** [6], **Binyamin Hochner** [5], **Paul S. Katz** [6], **Mei Zhen** [3], **Jeff W. Lichtman** [2,‡,*] **and Yaron Meirovitch** [2,‡,*]

[1] *Department of Neurobiology, Harvard Medical School, Boston, MA, USA*
[2] *Department of Cellular and Molecular Biology, Harvard University, Cambridge, MA, USA*
[3] *Lunenfeld-Tanenbaum Research Institute, Mount Sinai Hospital, Toronto, Ontario, Canada*
[4] *Department of Molecular Genetics, University of Toronto, Toronto, Ontario, Canada*
[5] *Department of Neurobiology, Silberman Institute of Life Sciences, The Hebrew University, Jerusalem, Israel*
[6] *Department Biology, University of Massachusetts Amherst, Amherst, MA, USA;*
[+] *These authors contributed equally*
[‡] *Senior authors*

Correspondence*:
Elisa Pavarino epavarino@fas.harvard.edu, Emma Yang emmayang@college.harvard.edu, Jeff W. Lichtman jeff@mcb.harvard.edu, Yaron Meirovitch yaron.mr@gmail.com

ABSTRACT

Connectomics is fundamental in propelling our understanding of the nervous system's organization, unearthing cells and wiring diagrams reconstructed from volume electron microscopy (EM) datasets. Such reconstructions, on the one hand, have benefited from ever more precise automatic segmentation methods, which leverage sophisticated deep learning architectures and advanced machine learning algorithms. On the other hand, the field of neuroscience at large, and of image processing in particular, has manifested a need for user-friendly and open source tools which enable the community to carry out advanced analyses. In line with this second vein, here we propose mEMbrain, an interactive MATLAB-based software which wraps algorithms and functions that enable labeling and segmentation of electron microscopy datasets in a user-friendly user interface compatible with Linux and Windows. Through its integration as an API to the volume annotation and segmentation tool VAST, mEMbrain encompasses functions for ground truth generation, image preprocessing, training of deep neural networks, and on-the-fly predictions for proofreading and evaluation. The final goals of our tool are to expedite manual labeling efforts and to harness MATLAB users with an array of semi-automatic approaches for instance segmentation. We tested our tool on a variety of datasets that span different species and developmental stages. With our software, our hope is to provide a solution

for lab-based neural reconstructions which does not require coding by the user, thus paving the way to affordable connectomics.

Keywords: affordable connectomics, semi-automatic reconstruction, segmentation, VAST, local, neural circuits

1 INTRODUCTION

Connectomics, the spearhead of modern neuroanatomy, has vastly expanded our understanding of the nervous system's organization. It was through careful observation of the neural tissue that Santiago Ramon y Cajal, father of modern neuroscience and predecessor of connectomics, reasoned that the nervous system is composed of discrete elements - the nerve cells. He further hypothesized key functional cell and circuit properties, such as neuronal polarity and information flow in neuronal networks, from anatomical observations, documented in extraordinary drawings. Connectomics - in particular based on electron microscopy images - has progressed immensely, and while the first complete connectome - the "mind of a worm" - was a manual decade-long endeavor for a reconstruction of merely 300 neurons (White et al., 1986), technological and methodological strides have enabled the field to elucidate complete circuitry from several other neural systems (Lichtman and Denk, 2011; Denk et al., 2012; Helmstaedter et al., 2013; Lichtman et al., 2014; Morgan and Lichtman, 2013; Hayworth et al., 2015; Kasthuri et al., 2015).

It would be highly impoverishing to view connectomics' purpose as merely the pursuit of neural circuit cataloguing. In recent years, in fact, connectomic reconstructions have been a new tool instrumental to answering outstanding questions in various subfields of neuroscience, which required synaptic resolution. Developmental studies have vastly benefited from microconnectomic reconstructions, opening the possibility of investigating precise synaptic rearrangements that take place in the first stages of life (Tapia et al., 2012; Wilson et al., 2019; Witvliet et al., 2021; Meirovitch et al., 2021). Further, circuit reconstructions have allowed in-depth studies of phylogenetically diverse systems, such as the ciliomotor system of larval *Platynereis*, (Verasztó et al., 2017, 2020), learning and memory in *octopus vulgaris* (Bidel et al., 2022), the olfactory and learning systems of *Drosophila* (Scheffer et al., 2020; Li et al., 2020) and the visuomotor system of *Ciona* (Salas et al., 2018). Connectomes have also provided insights into systems neuroscience, where avenues to pair structural and functional data from the same region of the brain are being explored. Noticeable examples of such endeavors are the study of mechanosensation in the zebrafish (Odstrcil et al., 2022), the study of the posterior parietal mouse cortex, important for decision making tasks (Kuan et al., 2022), and the functional and structural reconstructions of a mouse's primary visual cortex (Bock et al., 2011; Lee et al., 2016; Turner et al., 2022). Further, connectomes have proven to be a useful - and perhaps necessary - resource for computational modeling and simulation of circuits, by providing biological constraints such as connectivity, cell types and their anatomy. For example, the fly hemibrain (Scheffer et al., 2020) was queried to find cell candidates performing specific neural computations (Lu et al., 2022), murine connectomes have been shown to allow for discrimination between different candidate computational models of local circuits (Klinger et al., 2021), and the *C. elegans* connectome is being leveraged to simulate the first digital form of life through the open science project "OpenWorm" (Szigeti et al., 2014). Finally, we are at an exciting moment in connectomics' history, as recent reconstructions allow us to open a window on the human brain (Shapson-Coe et al., 2021). This important milestone, in conjunction with contemporary efforts to develop a whole mouse connectome (Abbott et al., 2020), will enable the community to reconstruct circuits in the context of neuropathology, and shed light on wiring diagram alterations that give rise to the so-called "connectopathies" (Lichtman et al., 2008; Abbott et al., 2020).

All these neural reconstructions have become a reality due to the progress in tissue preparation for electron microscopy and the tremendous progress in computer vision and artificial intelligence techniques. In particular, machine and deep learning techniques have become of common use for segmenting neural processes, thus aiding and expediting hefty manual annotation, which represents one of the main bottlenecks of the connectomic pipeline, and paving the way to high-throughput neural architecture studies. In this frame, convolutional neural networks (CNNs) have emerged as a successful solution for pixel classification. A typical automatic neurite reconstruction first begins by inferring probability maps of each pixel/voxel in the image for classifying boundaries of distinct cells (Ciresan et al., 2012; Turaga et al., 2010). In particular, U-net architectures have become common practice for biomedical image segmentation (Ronneberger et al., 2015), and are widely employed to achieve this first task. In a second step, a different algorithm agglomerates the pixels/voxels confined within the same cell outlines.

In the recent years, the field has benefited from deep learning algorithms designed specifically for the task of connectomic instance segmentation. One notable example of this is the Flood Filling Network architecture, a 3D CNN paired with a recurrent loop which segments in the volume one cell at a time by iteratively predicting and extending the cell's shape (Januszewski et al., 2018). A similar end-to-end approach iteratively segmenting one cross section of a neuron at a time has been pursued independently (Meirovitch et al., 2016). Recently this approach has been extended by training networks to flood fill numerous objects in parallel (Meirovitch et al., 2019). Many of these elaborate and heavily engineered pipelines (see also Section 3.4) present open source code repositories, however they remain of difficult practical use for researchers who do not have a software or computational background. For these reasons, many of the largest connectomics efforts have been carried out in collaboration with teams of computer scientists or even companies, option that requires a great deal of resources, both in terms of funding, and in terms of computing and storage capabilities.

While on one hand it is imperative to ever better the accuracy and scalability of these advanced algorithms, the field of image processing in particular, and science at large, have felt the urge for more democratic and easily accessible tools that can be intuitively employed by independent scientists. To name a few, tools such as ImageJ for general and multi-purpose image processing (Schneider et al., 2012; Schindelin et al., 2012), Ilastik (Berg et al., 2019) and Cellpose (Stringer et al., 2021) for cell segmentation, suite2p for calcium imaging (Pachitariu et al., 2017), Kilosort for electrophysiological data (Pachitariu et al., 2016), DeepLabCut (Mathis et al., 2018) and Moseq (Wiltschko et al., 2020) for behavioral analyses have enabled and empowered a larger number of scientists with the ability to carry out significant studies that previously would have been challenging or unfeasible, requiring non-trivial technical skills, time and resources. More specifically to the field of connectomics, there are a plethora of open software, mostly geared towards image labeling for manual reconstruction. Examples include but are not limited to VAST lite (Berger et al., 2018), Ilastik (Berg et al., 2019), NeuTU (Zhao et al., 2018), Knossos (Helmstaedter et al., 2011) with its online extension webKnossos (Boergens et al., 2017), and Reconstruct (Fiala, 2005). Because most of these software tools do not include a deep learning-based segmentation pipeline, a few software packages have been proposed to supply a CNN-based reconstruction, such as SegEM (Berning et al., 2015) which relies on skeletonized inputs for example from Knossos, and Uni-EM, a python-based software that wraps many of connectomics' image processing techniques (Urakubo et al., 2019).

We reckoned that making connectomics an affordable tool used by single labs meant providing a desktops solution compatible with the most common operating systems and computational frameworks currently used in the field. Thus, we focused our efforts here on creating a package based on MATLAB, which is one of the most commonly used coding environments in the basic science communities, providing its users with

a rich array of image processing and statistical analysis functions. Importantly, our main task here was not to present new functions for computer vision for connectomics, but rather we propose existing functions and machine learning models in a simple and user-friendly software package. Hence, the accuracy of our tool derives from the solutions presented previously in connectomics. As a second goal for our tool, we wished to create a virtuous and rapid EM reconstruction cycle which did not require solving the more expensive automated reconstruction problem. Thus, our deep learning tool greatly accelerates manual reconstruction in a manual reconstruction framework called VAST, an annotation and segmentation tool widespread in the community with numerous tools and benefits for data handling and data visualization. Hence, we expect our ML tools to be valuable to researchers that already use VAST. Thus, we created mEMbrain, a segmenting tool for affordable connectomics with the following attributes:

- mEMbrain has an interactive, intuitive, and simple interface, which leverages image processing and deep learning algorithms without requiring any coding knowledge by the user.
- mEMbrain is a MATLAB-based extension of VAST, a segmentation and annotation tool widely used in the Connectomics community (Berger et al., 2018). Using VAST as a server proves to be a clear-cut solution as it can splice the data and cache the space on demand, allowing mEMbrain to run on any cubical portion of datasets, independently of how the images are stored at the back-end.
- mEMbrain processes datasets locally on commodity hardware, thereby abolishing the need of expensive clusters and time-consuming data transfers.

We validated the robustness of mEMbrain by testing it on several species across different scales and parts of the nervous system. Further, we tested mEMbrain's speedup in terms of manual annotation time, and observed several fold improvement in manual time. All together, this paper presents new connectomic tools in platforms that had poor support for connectomic research. Furthermore, our tool extends the functionality of VAST to allow semi-automated reconstruction, already offered by other platforms.

2 MEMBRAIN'S CONCEPT

mEmbrain is a software tool that offers a pipeline for semi-automatic and machine learning-aided manual reconstruction of neural circuits through deep convolutional neural network (CNN) segmentation. Its user interface guides the user through all the necessary steps for semi-automatic reconstruction of electron microscopy (EM) datasets, comprising ground truth generation with data augmentation, data preprocessing, CNN training and monitoring, predictions based on electron microscopy datasets loaded in VAST, and on-the-fly validation of such predictions in VAST itself. mEMbrain is written in MATLAB, in order to interface seamlessly with VAST, a widely used annotation and segmentation tool (Berger et al., 2018). Most of today's pipelines involving machine and deep learning rely on Python, which although incredibly proficient and widely used in the computational community, is still less adopted in biological fields. We wanted to bridge this gap to make connectomics more accessible to a larger biological science community. mEMbrain can run on any operating system where both VAST and MATLAB (with parallel computing and deep learning toolboxes installed) are operative.

mEMbrain is a democratizer of computational image processing, which is necessary for EM circuit reconstruction. Its main purpose is to collect functions and processes normally carried out by software or computational scientists, and to embody them in a single software tool, which is intuitive and user-friendly, and accessible to any scientist. Thus, no coding skills are required for mEMbrain's operation.

mEMbrain's practicality starts from its installation. In many cases, software installation represents a hurdle, which in turn makes the frustrated user disinterested. To ease installation, our tool is a 5Mb folder downloadable from our GitHub page (github/mEMbrain). Once running, mEMbrain hosts all of its tools in one unique interface designed to be intuitive and user-friendly. mEMbrain's design is modular, with every tab presenting a different step of the workflow. Thus, the user can either be guided through the pipeline by following the tab order, or they can access directly the processing step of interest.

The main concept of mEMbrain is to create a synergistic dialogue with VAST in order to automate parts of the connectomic pipeline (see Figure 1). Typically, VAST is adopted by researchers for electron microscopy annotation and labeling. Such labeling and labeled microscopy can then be exported and directly used in mEMbrain, where images are processed and used to create datasets for training a deep learning model for semantic segmentation. Other than evaluating the results of the training phase through learning curves, the researcher can directly test how well the trained model performs, by making predictions on (portions of) the EM dataset open in VAST. The predictions are visible on-the-fly in VAST, superimposed on the open dataset. If the results achieved are not satisfactory, the user can improve the model by providing more ground truth examples; it is especially beneficial if the new labels incorporate regions and features of the dataset where the model predicted poorly. Hence, the newly generated ground truth is incorporated in the training dataset, and the deep learning model is retrained. This iterative process is continued until the results are deemed appropriate for the task at hand. In some cases, the iterative generation of new ground truth can be accelerated by making the deep learning segmentation editable in VAST, so that the researcher can swiftly correct such segmentation, saving time. Finally, once the prediction result is satisfactory, the final semantic segmentation can be leveraged for accelerating neural circuit reconstruction, by either using the predictions as a VAST layer, which dramatically speeds up manual painting (the main use-case of mEMbrain), using border predictions or by performing 2D instance segmentation. 3D instance segmentation algorithms are currently not incorporated in mEMbrain, but can be used in synergy with mEMbrain as surveyed in Sections 3.4 and 4.

3 EXAMPLE WORKFLOW

We here report the various steps of the image processing pipeline we have implemented and wrapped within mEMbrain. For the typical flow, refer to Figure 2 for our general purpose GUI, and Figure 3 for the specific pipeline adopted for the *C. elegans* data described in Section 4.3.

3.1 Dataset creation and image preprocessing

The first step towards training neural networks for segmentation is the creation of a training dataset composed of both images and associated ground truth, or labels. It is common wisdom that abundant ground truth will yield a better prediction of the training algorithm. We realize that the preparation and curation of a comprehensive training dataset can represent a hurdle for many researchers. One strategy might be to label many EM images, however, this requires many hours dedicated to tedious manual annotation. Alternatively, computational methods can be leveraged for image processing; however, this requires having a good mastery of coding skills. Thus, we incorporated a dataset creation step, which allows researchers to process the labeled images paired with their EM counterpart with just a few mouse clicks. Once the user has imported the microscopy images coupled with their labels, mEMbrain converts the latter in images with 2 or 3 classes, depending on the task at hand. The EM images are then corrected by stretching their grayscale. Subsequently, patches of a user-chosen dimension are extracted from the pair of EM and label images. Notably, mEMbrain first verifies the portion of the image that presents a saturated annotation, which can

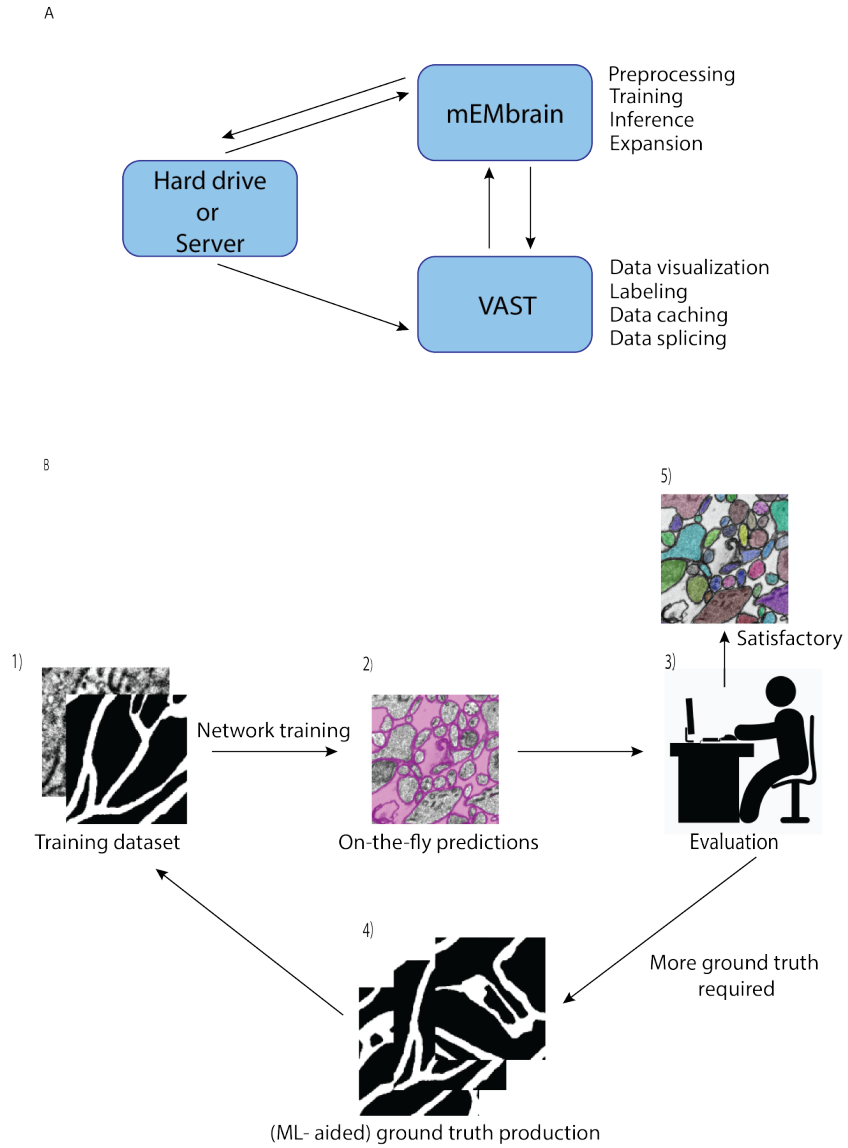

Figure 1. *mMEbrain's workflow and integration with VAST.* **(A):** communication between mEMbrain, VAST and data storage. mEMbrain and VAST communicate bidirectionally, as VAST stores, caches and splices the data which can then be imported into mEMbrain through VAST's application programming interface. mEMbrain's outputs are then transferred back to VAST for visualization and postprocessing. mEMbrain can also access the data directly at where it is stored, and will save there its outputs (if so the user indicates). **(B):** mEMbrain's iterative workflow. The user starts by creating a training dataset of EM and corresponding labels 1), which are then used to train a convolutional deep learning network. The results of such network can be visualized on-the-fly directly on datasets open in VAST 2). Further, if the researcher is satisfied with the current state of network inference, they may proceed to a semi-automatic approach for semantic segmentation 3). However, if they are not satisfied with the current output of the network, the can use these predictions to accelerate further ground truth production 3-4), which is then incorporated in further training of the network to achieve better results 4).

assume any arbitrary shape desired by the researcher. Then, mEMbrain efficiently extracts patches from such regions. Thus, the images do not have to be fully annotated for them to be incorporated in the training dataset, and this feature makes the region of interest selection more flexible, faster and seamless.

Figure 2. *mEMbrain's MATLAB-based GUI for data preprocessing, training, inference, and integration with VAST.* All the functions are collected in one user interface, and can be accessed by clicking the different tabs. **(A)**: These first 3 tabs allow the user to create a training dataset from EM images and corresponding ground truth. **(B)**: Further, the user can train a deep neural network. As default, we make use of MATLAB's built-in U-net, whose training can be customized through the various user-chosen parameters. The training's progress can be monitored by MATLAB's learning curves. **(C)**: To evaluate a network, predictions can be made on small sample images, as seen in the small squares of the GUI. **(D)**: Finally, researchers can infer directly on-the-fly in VAST on the dataset herein open. Further, they can convert such inference to editable layers in VAST, that may be leveraged for machine learning (ML)-based ground truth preparation.

One noteworthy feature of this step is the incorporation of data augmentation, in the hope that fewer annotated images are required to obtain a satisfying result. In particular, we verified that rotations yielded a better result during testing phase, hence we implemented a random rotation of any possible degree for every pair of patches. At each rotation of the ground truth data, mEMbrain uses the chessboard distance (or Chebyshev distance) between labeled pixels of the ground truth to the closest unlabeled pixels. Then, mEMbrain individuates pixels around which a square patch of user-defined size will contain fully annotated pixels, and such pixels are then used for patch generation. Further data augmentation methods such as image flipping, Gaussian blurring, motion blurring and histogram equalizer are also implemented. This ensemble of techniques ensures that nearby regions from the same image can be more heavily sampled for patch generation without making the training overfit such a region, allowing the extraction of "more patches for your brush stroke".

In addition to this "smart" patch generation feature, mEMbrain also includes conversion features for a) instance segmentation ground truth to contours ground truth (i.e. membrane ground truth) and b) membrane ground truth to skeleton ground truth. The former uses erosion and dilation with a user-specified filter radius to transform filled-in neuron segmentation annotations into membrane ground truth of a specified thickness. mEMbrain can also generate this membrane data with or without extracellular space filled in. For the latter feature, mEMbrain uses MATLAB's built-in 2D binary skeletonization functions to generate 2D neuronal skeletons from membrane ground truth. The utility of such ground truth conversion lies in the possibility to then train subsequent deep learning networks in a supervised manner to learn and predict the medial axis of the neuronal backbone. Learning such neuronal backbone enhances the ability of existing reconstruction algorithms to agglomerate objects, as seen in Section 3.4. While we

implemented agglomeration techniques using neuronal backbones predicted by EM (see Figure 4), these will be integrated into mEMbrain's subsequent software release, and are here described for their novelty and to allow the connectomics community to further test and explore such methods.

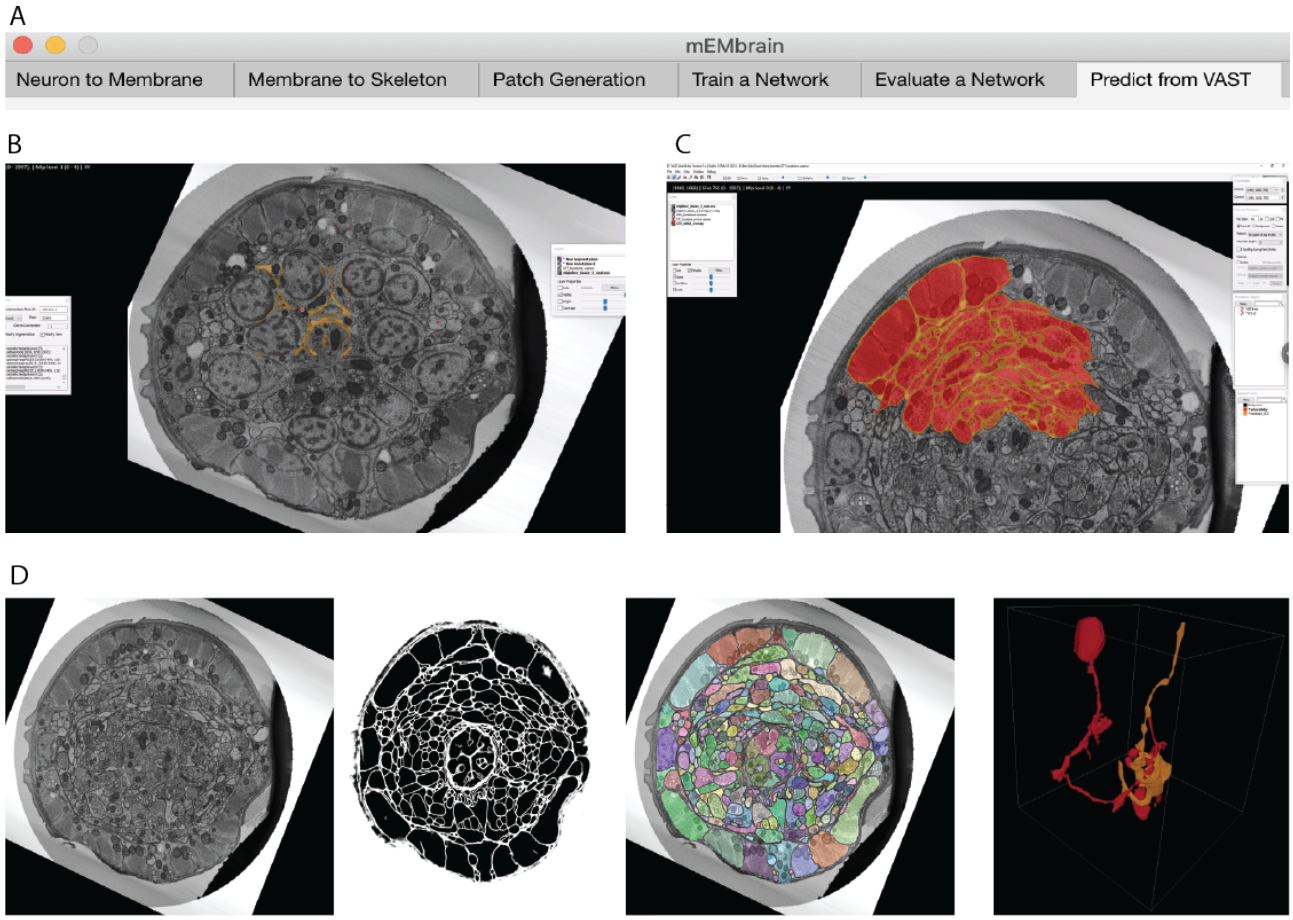

Figure 3. *Example workflow with mEMbrain and VAST.* **(A)**: mEMbrain's header with step-by-step pipeline for deep learning segmentation of EM images. **(B)**: The user can initially apply a pretrained network on the dataset at hand and use these predictions both for a) a first evaluation of which areas of the dataset should be included in ground truth, and b) as a base for ground truth generation. The figure shows VAST's window with the *C. elegans* dataset open. In orange, the predictions of a pretrained network are shown. **(C)**: The user can convert the predictions to an editable layer in VAST, and use these as rough drafts of ground truth. By manually correcting these, one can generate labels in a swifter manner, saving roughly half the time. **(D)**: Left: EM section of the *C. elegans* dauer state dataset. Second panel: mEMbrian's predictions of the same section. Third panel: mEMbrain's 2D expansion. Right: example of 3D reconstruction obtained through automatic agglomeration algorithms. The two reconstructed neurites are shown in VAST's 3D viewer.

3.2 Network training

Once datasets are created and preprocessed, researchers are in the position to train a network for image segmentation. There are two options for approaching the training phase:

1. train a pre-implemented U-net (Ronneberger et al., 2015);
2. load a pre-defined network and continue training upon it.

The implementation of U-net was chosen given the success this deep learning architecture has in the field of biomedical imaging segmentation. Although the implementation of the network is built-in to MATLAB, the user still preserves ample degrees of freedom for customizing parameters of both the network architecture - such as the number of layers - and the training - for example the hyper parameters and the learning algorithm. As of now, most of mEMbrain's features work when predicting 2 or 3 classes for the step of semantic segmentation. Importantly, the network can be saved as a matrix with trained weights, which can then be used for future transfer learning experiments (see section 4.6).

Alternatively, pre-trained networks can be loaded in mEMbrain to be re-trained. As discussed in Section 4.6, learning upon a pre-trained network, a strategy in the domain of transfer learning here referred to as *continuous learning*, typically yields better results with less ground truth. Of note, it is possible to import networks that have been trained with other platforms, such as PyTorch or Tensorflow, thanks to designated MATLAB functions (for a tutorial, the reader is referred to (mat, 2022)), or to import/export trained networks and architectures using the ONNX (Open Neural Network Exchange) open-source AI ecosystem format which is supported by various platforms including mEMbrain and MATLAB. Since much of deep-learning-enabled connectomics is done in Python-based machine learning platforms, we also wanted these users to be able to integrate mEMbrain into their workflows. Thus, we also implemented a feature where users can export neural networks trained on ground truth data in mEMbrain to the Open Neural Network Exchange (ONNX) format. This format preserves the architecture and trained weights of the model, allowing the user to import the model back into Python-based platforms such as Tensorflow and Pytorch for further investigation and analysis.

For a quick assessment of deep learning model training, we implemented an evaluation tab, where one can use such a model to make predictions on a few test images and qualitatively gauge the goodness of the network.

3.3 On-the-fly predictions with VAST

Once one has trained a deep learning model for semantic segmentation and is satisfied with its results, prediction on the dataset may be carried out. mEMbrain has 3 different modalities for prediction, namely:

- predictions on whole EM volumes;
- predictions on specific regions of the EM volume;
- predictions around anchor points positioned in VAST.

When users predict on whole EM datasets - or portions of it - by either inserting the coordinates delimiting the regions of interest or by using VAST's current view range, mEMbrain requests EM images open in VAST at that moment through the application program interface (API). Our implementation speeds up EM exporting by optimizing the image request and tailoring it to VAST's caching system (Berger et al., 2018). Because VAST caches 16 contiguous sections at one given time, mEMbrain requests chunks of 16 [1024 x 1024] sections at a time, reading first in the dataset's z dimension, proceeding then in the x and y dimensions. Data is read at the mip level chosen by the user, which should match the resolution at which the network was trained. Once having read the EM images, mEMbrain corrects them with the same grayscale correction that was applied when preparing training datasets, and then it predicts the semantic segmentation with the chosen deep learning model. Because the training phase occurs on patches that have dimensions in multiples of [128 x 128] pixels, predictions on [1024 x 1024] pixels at a time is a valid operation. Once image pixels are classified, the predictions are saved as .pngs in a folder designated by the user. At the same time, mEMbrain creates a descriptor file (with extension .vsvi), which is a text file

following the JSON syntax that specifies the naming scheme and the storage location of the predicted images, as well as other metadata necessary for the dataset. Once created, the .vsvi file can be loaded (dragged and dropped) in VAST, which then loads the predictions, which can be viewed superimposed on the EM dataset.

It might be useful, in some scenarios, to predict and segment only particular regions which do not all align along the same z axis. Leveraging VAST's skeleton feature, researchers may allocate anchor points in regions of interest throughout the dataset. mEMbrain can then predict locally around such anchor points. One example of such scenario is when trying to determine if a deep learning model provides satisfactory predictions on a large dataset. For such evaluation, mEMbrain can predict a set of cubes centered around pre-selected coordinates (represented by VAST skeletons). Based on the outcome, the user can decide if the model's output is satisfactory. Another example is the prediction only around certain regions of interest sparse through the dataset, such as synapses.

3.4 mEMbrain as part of 3D instance segmentation pipelines

Although currently mEMbrain does not have a 3D pipeline incorporated within its GUI, nevertheless our tool has been used for enabling 3D reconstructions. In fact, many of the pipelines for 3D reconstruction do rely on having high quality membrane prediction, which is one of our software's main output. Thus, we tested mEMbrain as an essential prerequisite for one of the published 3D pipelines, namely the Cross Classification Clustering (3C) algorithm (Meirovitch et al., 2019), and its results are then showcased in the *C. elegans* and the whole mouse brain datasets in Section 4.

With a few exceptions, all of the widely used reconstruction pipelines make an intermediate use of a membrane probabilities and an additional provisional over-segmentation (Turaga et al., 2010; Lee et al., 2015; Meirovitch et al., 2016; Lee et al., 2017; Wolf et al., 2018; Beier et al., 2017; Funke et al., 2018; Meirovitch et al., 2019; Macrina et al., 2021). We demonstrate the utility of this approach by leveraging recent agglomeration techniques, such as Cross Classification Clustering (Meirovitch et al., 2019), combined with novel predictions of neuronal backbones, described in Section 3.1. In a first step, mEMbrain's output inferences are used to partition the space. This first over-segmentation, which aims at minimizing the number of objects internally contained in a neuronal boundary, is subsequently merged. While some pipelines attempt to achieve this goal with small 3D supervoxels (Macrina et al., 2021), we used here 2D segmentations that have a good representation of the neuronal cross section (Figure 4.2 A). Agglomeration proceeds with an optimization step that matches 2-D objects across sections. This process may be aided and furthered by a novel agglomeration technique based on supervised learning of the medial axis of neurons, or 2-dimensional skeletons, as we presented in Section 3.1, which are learned from the conversion of membrane ground truth to skeleton ground truth. Our tests on the various datasets (see Section 4) demonstrate such convolutional neural networks can learn the medial axis of thin neurites and the intricate morphology of enwrapping glial cells, leading to complex topology of the skeletons such as circular skeletons for enwrapping objects and confident detection of bifurcations, including spine protrusions from dendritic shafts (see Figure 4). Overall, our empirical tests propose that learning and predicting skeletons directly from the EM images provides additional information to traditional pipelines. In particular, the agglomeration step can make use of such information by assembling objects whose skeleton strongly overlap across planes. Results of this algorithm are visible in Figure 5.

3.5 Expansion

mEMbrain's output prediction until this step is a categorical image (*i.e.* each pixel is assigned to one of the classes the network was trained on) accompanied by its relative probability map (*i.e.* how sure the network is that a said pixel pertains to an assigned class). However, for the vast majority of connectomics tasks, each cell should be individually identifiable. Predictions of EM images in different classes are a powerful resource that can either strongly expedite manual reconstruction, or can be the first step necessary for many semi-automatic reconstruction methods. These labels can be directly imported in VAST and used in the following manners:

- **Machine learning-aided manual annotation with membrane-constrained painting** (i.e. "membrane detection+pen" mode). In this modality, the manual stroke of paint is restricted to be contiguous with mEMbrain's membrane prediction. This allows the user to proceed in a swift manner, negligent of details such as complex borders that require a hefty amount of time if done precisely by hand.
- **Annotation with VAST's flood filling functionality with underlying mEMbrain's 2D segmentation**. By clicking once on the neurite of interest with the filling tool, the object is colored and expanded until it reaches the borders predicted by mEMbrain.
- Any other expansion algorithm that creates an instance segmentation starting from a semantic one.

4 DATASET SHOWCASE

mEMbrain has been used to reconstruct neurons and neural circuits in a number of datasets, spanning different regions of the nervous system (including central and peripheral) at multiple scales (from cellular organelles to multi-nucleated cells) and across diverse species (including various invertebrates and mammals). Here follow some of the most interesting uses of mEMbrain insofar, showcasing a variety of unpublished datasets where our software had the opportunity to be tested, and where it played a pivotal role. The predictions carried out by mEMbrain were done on a Nvidia RTX 2080Ti GPU, which computed at a speed of 0.2 seconds/MB.

4.1 The whole mouse brain dataset

We employed mEMbrain in our ongoing efforts to develop staining and cutting protocols that will eventually enable the reconstruction of a whole mouse brain (Lu *et al.*, in preparation). In the current phase of the project, a newborn whole mouse brain was stained and cut, and several sections were stitched. The region of interest here shown is from the mouse's motor cortex M2, covering layers II/III through VI. The sample was imaged with a Zeiss multibeam scanning electron microscope, at a resolution of 4x4x40 nm/px, resulting in a total volume of 180×303×4m.

The role of mEMbrain in this project was to assess the feasibility of reconstructing neural circuits when using such staining and cutting protocols. We started from a network pre-trained on adult mouse cortex. 6 iterations of network training and manual corrections were needed in order to achieve good results, which amounted to 50 hours of ground truth preparation. We then predicted all the cell membranes in the volume and segmented each 2D section. The predictions were carried out on a desktop with a single GPU Nvidia RTX 2080 Ti, which required 5 days. Further, we used an automatic agglomeration algorithm (Meirovitch et al., 2019) to reconstruct 3D cells; the high quality results with an exceptionally low rate of merge errors (see Figure 5), reassure that these new protocols may consent larger scale mouse brain reconstructions.

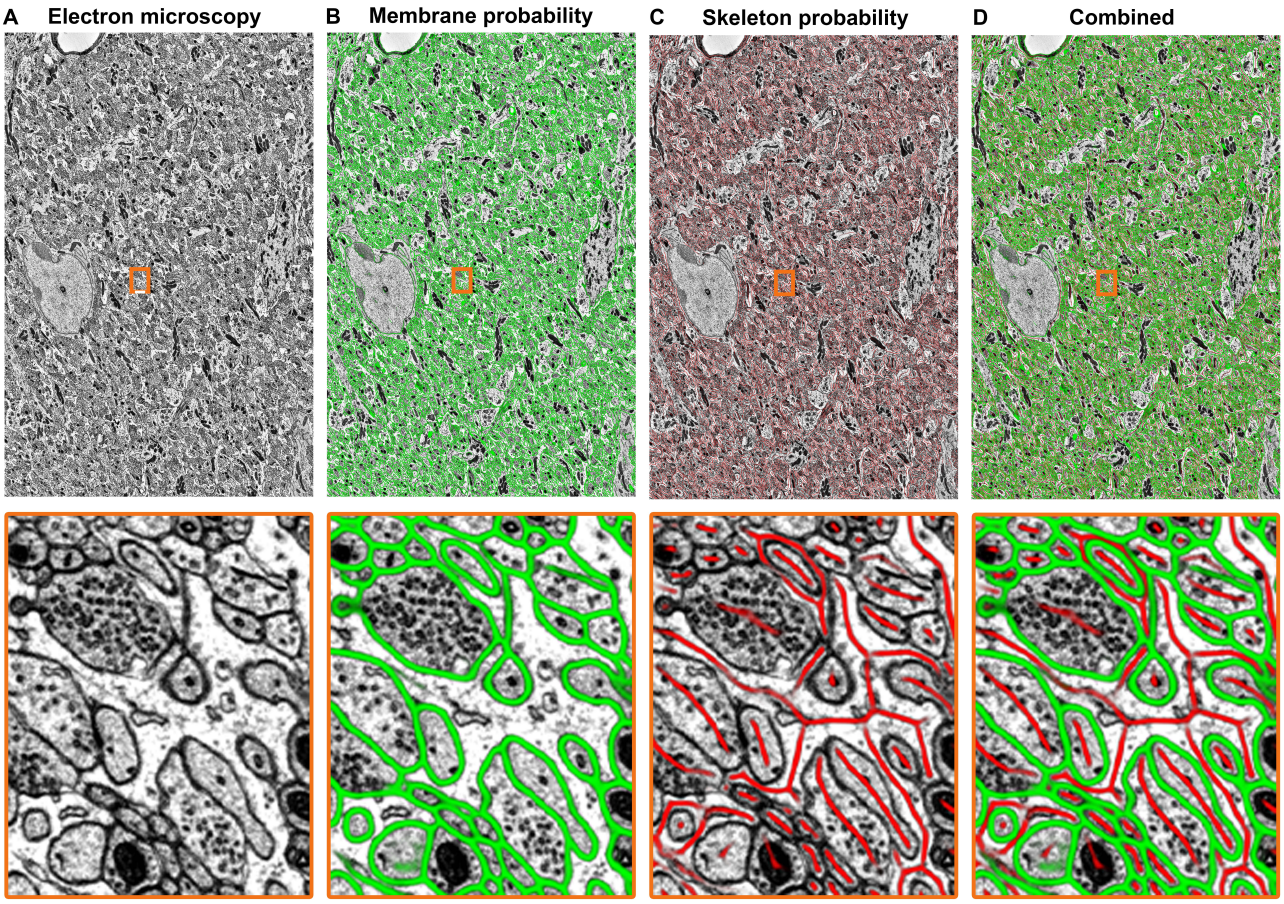

Figure 4. *mEMbrain's skeleton predictions.* **(A)**: Electron microscopy tile from from the Cerebellum dataset (see Section 4.2). Dimensions: 5731 x 3813 pixels. Resolution: 8nm per pixel along the x and y axes. **(B)**: mEMbrain's membrane predictions achieved with supervised learning using expert-made membrane ground truth. **(C)**: Skeleton predictions achieved with supervised learning. The ground truth of such learning is provided by mEMbrain's automatic conversion of expert-made membrane ground truth into skeleton ground truth (see Section 3.1). Similar to the membrane neural network, the input to the skeletonizing network is the EM image with 4nm pixel resolution. **(D)**: Superposition of the membrane and skeleton probabilities as provided by the two separately trained convolutional neural networks. The magnified images show the ability of the skeleton predictions to correctly detect continuous structures - often glia - that enwrap around neurons, and thin neurites whose membrane probabilities are not always neatly defined.

4.2 The Mouse Cerebellum dataset

We tested mEMbrain on different regions of the mouse nervous system. Here, we report about our software's use on the developmental mouse Cerebellum dataset (Dhanyasi *et al.*, in preparation). The rationale behind this research is to study the development of the cerebellar circuits using electron microscopy. The region of interest is from the vermis, a midline region of the cerebellar cortex (Strata et al., 2012). The sample was imaged with a Zeiss multibeam scanning electron microscope at a resolution of 4x4x30 nm/px, yielding a traceable volume of 650x320x240 µm.

In the context of this dataset, mEMbrain was used to expedite manual annotation, by both using mEMbrain's predicted cell boundaries as constraints in VAST (see Section 3.5, Method 1), and by carrying out 2D instance segmentation provided by our tool. The researcher reported the greatest speed-up for this

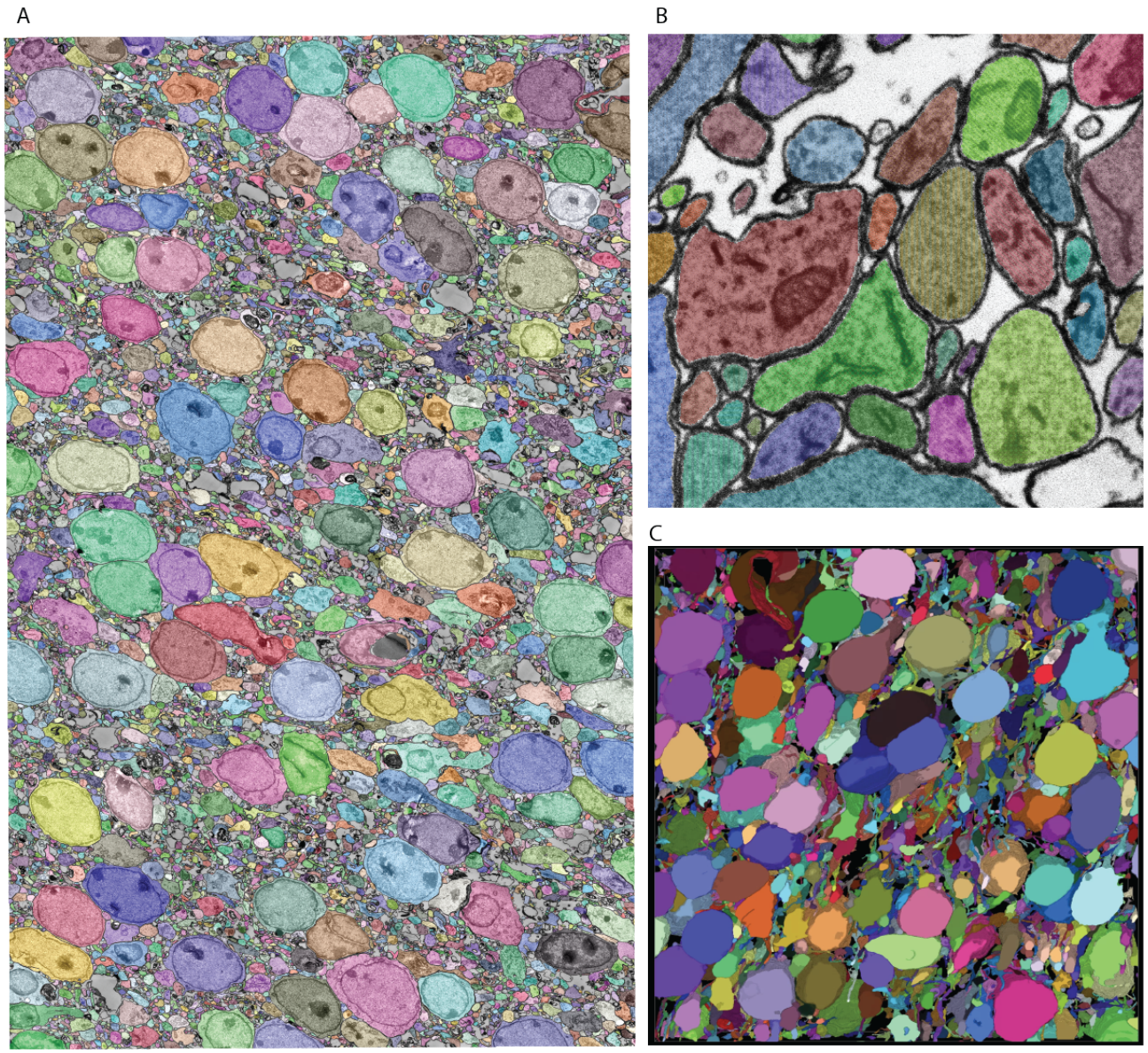

Figure 5. *Showcase results of mEMbrain on the whole mouse brain dataset.* **(A)**: 2D section of the whole mouse dataset segmented by using mEMbrain's cell contour prediction in combination with automatic agglomeration methods (Meirovitch et al., 2019). **(B)**: Example of a small region of interest of the dataset, meant to highlight the good quality of the results. **(C)**: Portion of a stack of sections, visualized in VAST's 3D viewer. Lu *et al.*, in preparation.

dataset to be provided by the 2D instance segmentation. To corroborate this assessment, an additional speed test was performed by three annotators, and an estimate of the expedition offered by our semi-automatic methods is recounted in Section 5.

4.3 The *C. elegans* dataset

We assessed our software on a number of invertebrates. Here we show mEMbrain's employment on one *C. elegans* dataset. This sample (Britz et al., 2021) was a wildtype nematode in the dauer diapause, an alternative, stress-resistant larval stage geared towards survival (Cassada and Russell, 1975). The sample,

with a cylindrical shape in a diameter of 15.800 µm was imaged with a focused ion beam - scanning electron microscope (FIB-SEM) at a resolution of 5x5x8 nm (Britz et al., 2021).

For this dataset, mEMbrain was used as a semi-automatic segmentation tool. A network was trained on an original set of ground truth, and then was used to predict on portions of the dataset for a swift qualitative evaluation of its output. After identifying regions that required more representation in the training set, more ground truth was generated by using the network's predictions and rapidly correcting them manually. The researchers computed that semi-automatic ground truth generation cut the manual annotation labor time by a little less than 50%: the ground truth required for the first training iteration took 14 hours of manual annotation. Similarly, also subsequent iterations cumulatively required 18 hours of painting. However, the volume traced in this amount of time is doubled with respect to the first iteration. The workflow of this dataset is shown in Figure 2.

4.4 The Octopus Vertical Lobe dataset

We had the unique opportunity to test mEMbrain on non-conventional model organisms in the neuroscience community, thus testing the usefulness and generalizability of our tool across species. In particular, we were excited to assess mEMbrain on a sample from the *Octopus vulgaris* dataset (Bidel et al., 2022). The region of interest is in a lateral lobule of the *Octopus vulgaris*' vertical lobe (VL), a brain structure mediating acquisition of long-term memory in this behaviorally advanced mollusk (Shomrat et al., 2008; Turchetti-Maia et al., 2017). The sample was imaged at high resolution with a Zeiss FEI Magellan scanning electron microscope equipped with a custom image acquisition software (Hayworth et al., 2014). The ROI was scanned over 891 sections each 30 nm thick at a resolution of 4 nm/px, constituting a traceable 3D stack of 260x390x27 µm.

mEMbrain was here mostly used for aiding manual annotation. As described in Section 3.5, the output semantic segmentation obtained with mEMbrain can be directly utilized in VAST as constraints for the annotation of objects. In this manner, a single drop of paint floods the entirety of the neurite, and allows the researcher to proceed in a swift manner, without needing to pay attention to anatomical details. For this dataset, the researchers using our software reported that there is a 2-fold increase in speed with mEMbrain's aid when the purpose is to simply roughly skeletonize a cell, not being mindful of morphological details. However, the most significant advantage of using mEMbrain is the expediency of precise anatomical reconstructions, given that accurate reconstructions consume a sizeable amount of manual time. Instead, with mEMbrain, the time to skeletonize a neurite matches the time it takes to reconstruct it accurately; explaining why in this modality there is a 10-fold increase in speed when using mEMbrain. For example, this allowed for a fast and precise reconstruction of axonal boutons and cell bodies, which enabled subsequent morphometric analysis (see Figure 6).

4.5 The *Berghia stephanieae* dataset

We tested our tool on a second mollusc, the nudibranch *Berghia stephanieae*, a species of sea slug newly introduced for neuroscience research. The aim of this project is to determine the synaptic connectivity of neurons in the rhinophore ganglion, which receives input from the olfactory sensory organs. The rhinophore connective contains axons that travel between the rhinophore ganglion and the cerebral ganglion. The sample of the rhinophore connective here was sectioned at 33 nm and imaged with a Zeiss scanning electron microscope at a resolution of 4 nm/px (Drescher et al., 2021).

The *Berghia* dataset was the first one on which we witnessed the power of transfer learning (see Section 4.6). 7-10 hours of ground truth annotating produced a handful of labels, that were used to perform

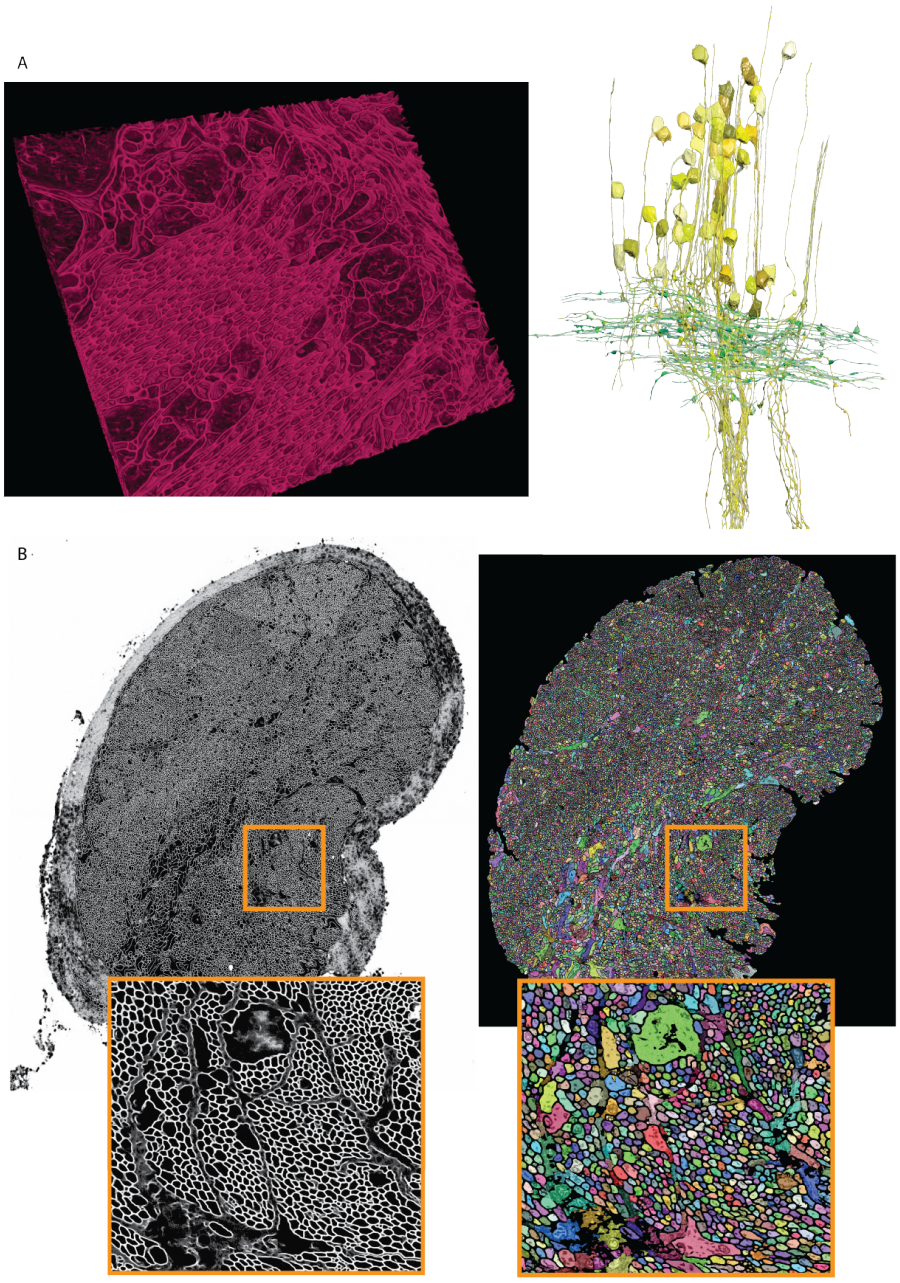

Figure 6. *Showcase results of mEMbrain on the molluscs' datasets.* **(A):** Examples of the *Octopus vulgaris* dataset, by Bidel *et al.* (Bidel et al., 2022). On the left, sample of the cell boundaries predicted by mEMbrain and shown in VAST's 3D viewer. On the right: 3D rendering of interneurons (yellow) and afferents (green) in the learning and memory brain center in the octopus brain. Reconstruction mode: pen annotation constrained "on-the-fly" in VAST by mEMbrain's border probabilities. **(B):** Examples of the dataset from the rhinophore connective of the nudibranch, *Berghia stephanieae* (Drescher et al., 2021). The left image shows membrane predictions in a whole connective slice, while the right panel shows a section from the same region with instance segmentation applied, which was obtained starting from mEMbrain's cell boundaries and applying the automatic algorithm 3C (Meirovitch et al., 2019). To appreciate the sheer number of processes in this brain area, small regions are zoomed out in orange.

continuous learning from a network pre-trained on the *Octopus vulgaris* dataset. mEMbrain's output was used to obtain 3D segmentation when agglomerated with automatic algorithms (Meirovitch et al., 2019). This reconstruction enabled the possibility to automatically count the number of processes present in the

rhinophore connective tissue region, and revealed that this part of the nudibranch nervous system harbors an exceedingly high number of processes (roughly 30 000 - the counting was double checked by manual inspection). This was an important finding, as the *Berghia stephanieae*'s rhinophore ganglion itself contains only 9000 cell bodies (Drescher et al., 2021). The complex organization and the abundance of processes (shown in Figure 6) suggest that such peripheral organs are highly interconnected with the central nervous system of the animal, sharing similarities with octopuses and other cephalopods (Hochner, 2012; Zullo and Hochner, 2011).

4.6 Transfer Learning

One tool that we found incredibly valuable in our reconstructions was using knowledge learnt from one dataset and applying it towards others, leveraging the concept of *transfer learning*, and more specifically of domain adaptation Roels et al. (2019). We experimented with a variety of modalities for transfer learning. We started by freezing all the model's weights except for the last layer, a strategy that maintains the internal representations previously learned by the model, while fine tuning the last layer for the specific new dataset at hand. We then tested the idea of freezing only the model's encoding weights, in other words the first half of a U-Net architecture, while allowing the decoder's weights to fine tune for the new dataset. Further, we explored allowing the encoder to learn at a very slow rate (maintaining most of the pre-trained knowledge), typically 10 times smaller than the decoder's learning rate, in a technique called "leaky freeze". Moreover, we tested applying a continuous learning approach, whereby after training on a first dataset, the same network is trained on a second one without modification of its learning rates. One concern that might arise with this approach is the occurrence of catastrophic forgetting, which is the tendency of a network to completely and abruptly forget previous learned information, upon learning new information (McCloskey and Cohen, 1989). For this reason, we also tested an episodic memory strategy, where the training schedule interleaves learning from the two datasets at hand.

The main conclusion of our multiple experiments is that the strategy of transfer learning significantly reduces the time needed to achieve satisfactory results; pre-trained networks have already learned multiple fundamental features of EM images, tentatively distinguishing membranes of cells. Thus, the training of networks on subsequent datasets is geared towards fine-tuning their *a priori* knowledge and adapting it to the specific dataset at hand. This means that the number of epochs - that is the number of passes of the whole training dataset that the deep learning network has completed - required for good performance is significantly less than when training a network from scratch. Furthermore, the amount of ground truth needed to achieve satisfactory results is also drastically reduced, as many of the features - such as edge detection, boundary detection, and general interpretation of different gray scales of electron microscopy images - have already been assimilated from learning on the previous data. The second conclusion from our tests highlights that the strategy of continuous learning is the one that yielded the best results. Further, this method is particularly user-friendly given that no alterations to the network need to be made.

It is important to note that transfer learning works best when the network trains on datasets that share many common features. One striking example where transfer learning proved to be a powerful technique was in the *Berghia stephanieae* dataset. For this project, the human-generated ground truth was reasonably scarce, and hence when a network was trained with mEMbrain for semantic segmentation, the outcomes were quite poor, as can be seen in Figure 7. However, we noticed a qualitatively strong resemblance between the EM image properties of the *Berghia stephanieae* and of the *Octopus vulgaris*. We reasoned that this could be a case in which transfer learning techniques would be especially impactful in aiding the paucity of ground truth to learn from. Thus, we took the best-performing network trained on the *Octopus*

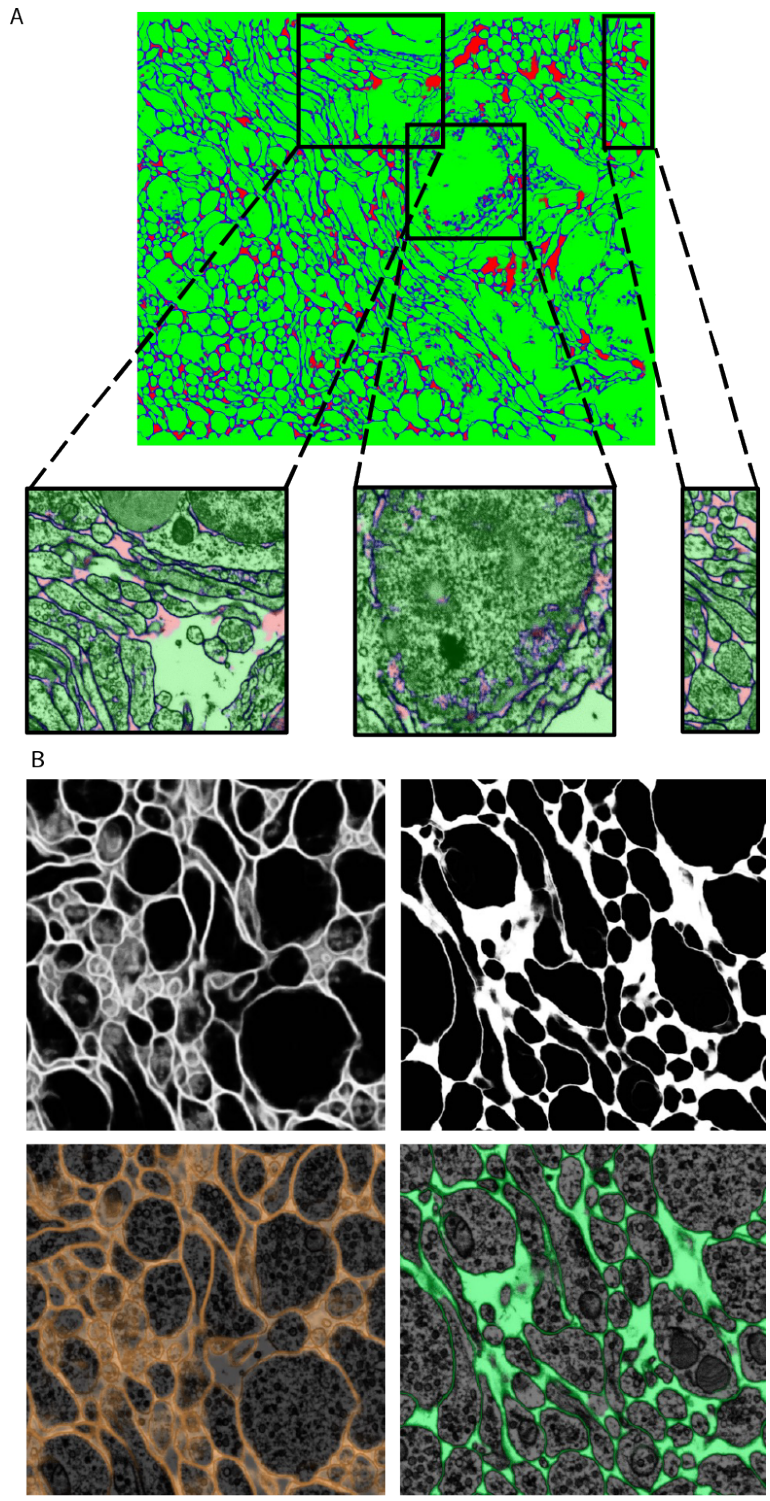

Figure 7. *Transfer learning approaches from the Octopus to the Berghia stephanieae datasets*. **(A)**: example of poor generalization of the network on the *Berghia* dataset, due to limited training ground truth. In green the intracellular space, in blue the cell boundaries, and in red the remainders. **(B):** networks pretrained on the *Octopus* dataset predictions on the *Berghia* dataset without continuous learning (left) and with continuous learning (right). In grayscale are membranes only, while below they are overlayed to EM images.

vulgaris and we trained it in a continuous learning fashion for 5 subsequent epochs on 3 ground truth images from the *Berghia stephanieae* dataset. Within only 10 minutes of training, the validation accuracy of the network reached 97% and the results were of high quality, as can be seen from Figure 7.

Hence, working with pre-trained networks and fine-tuning them on the specific dataset at hand dramatically reduces the time invested both in ground truth generation and in training of the network. We highly recommend to save previously trained networks and to further their learning on new datasets in order to expedite the segmentation process.

5 EVALUATING SPEED UP WITH MACHINE LEARNING-AIDED PAINTING

We tested the speed up provided by mEMbrain's output by conducting a proof-of-concept timed experiment. We asked three experienced researchers to manually annotate one neurite for 10 minutes. We then compared the resulting labeled volume with the volumes annotated by the same researchers when using mEMbrain's output in combination with VAST's tools. In particular, we tested:

- using mEMbrain's 2D segmentation in combination with VAST's pen annotation mode (Section 3.5, Method 1);
- using mEMbrain's 2D segmentation in combination with VAST's filling tool (Section 3.5, Method 2);
- using machine learning-aided manual annotation with membrane-constrained painting carried out with VAST's pen annotation mode;

We benchmarked such methods against manual annotations only. The tests were carried on the *Mouse Cerebellum* dataset, presented in Section 4.2. The results are quantified in Figure 8C. The main finding is that painting with an underlying machine learning aid is at least 20 times faster than labeling purely with manual approaches. More specifically, the combination of mEMbrain's 2D segmentation together with VAST's pen annotation model yields the fastest results, particularly when striving for accuracy. In contrast, opting for mEMbrain's 2D segmentation in tandem with VAST's flooding tool, while vastly accelerating manual labor, might be suboptimal in scenarios in which VAST's flooding tool could yield to merge errors, which in turn require more time for correction and label postprocessing. However, this modality has been reported by our user to be most ergonomic. This speed evaluation will need to be corroborated by future tests on different datasets.

6 COMPARISON WITH OTHER TOOLS

While there are many free software tools in the field for labeling and manual annotation, visualization and proofreading, there are fewer software providing a comprehensive and user-friendly pipeline for CNN training geared towards EM segmentation. One first aspect to notice is that all software, mEMbrain included, rely on other packages for visualization and proofreading. The power of mEMbrain relies precisely in its synergy with VAST, which is excellent for data handling, visualization, annotation, and offers a variety of tools that can be co-leveraged together with our software. For these reasons, mEMbrain features the very useful ability to predict on-the-fly in regions chosen by the researcher and immediately visualizable in VAST. This greatly enables the scientist to assess the quality of mMEbrain's outcome, and mitigates the the time for import and export of datasets and segmentations.

Another feature of mEMbrain we deem fundamental is its wrapping of all the pipeline in one unique GUI, without the user having to interact with code and having to master different interfaces. Importantly, mEMbrain provides the ability to create datasets necessary for the training phase, which are data-augmented

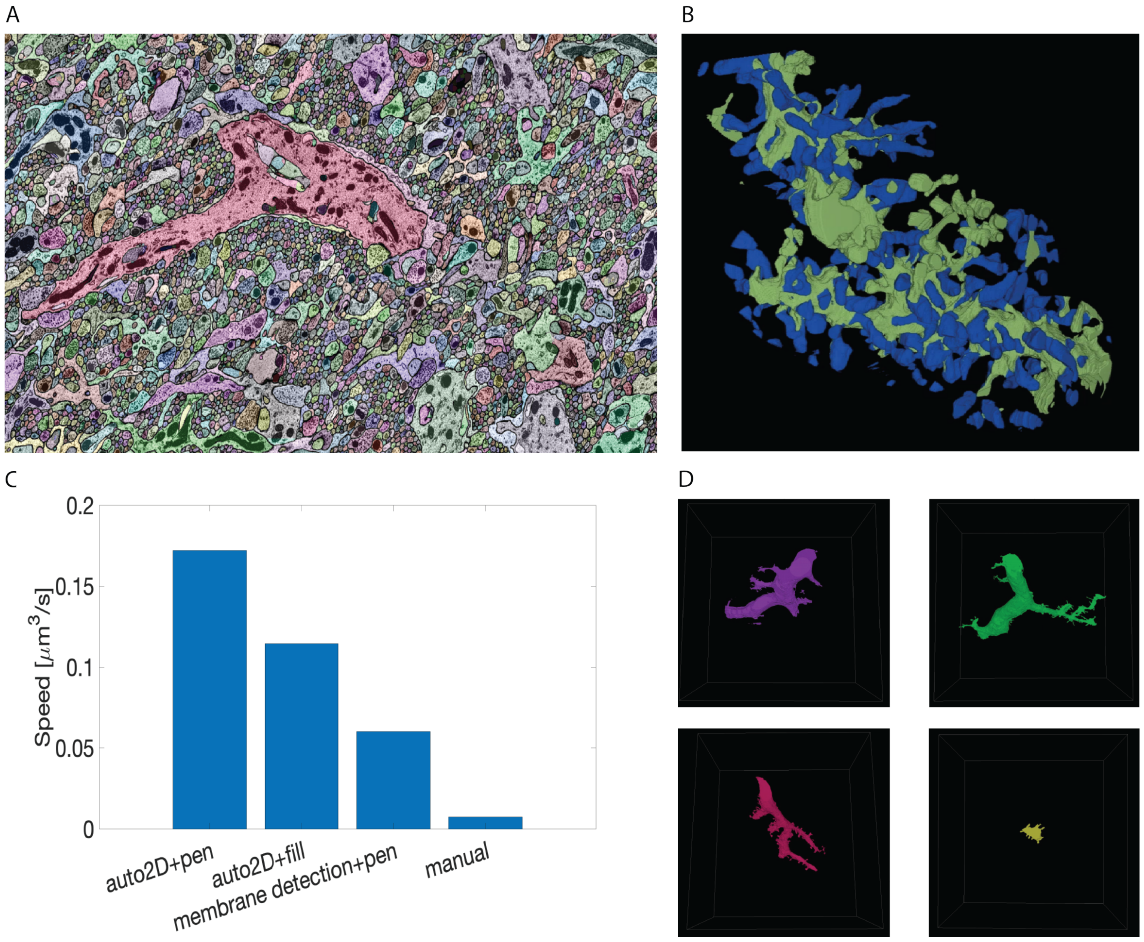

Figure 8. *Examples of the mouse Cerebellum dataset and summary of speed up test with machine learning-aided painting.* **(A)**: Portion of a section from the cerebellum dataset, from Dhanyasi *et al.*, in preparation. The 2D instance segmentation was achieved through mEMbrain's ML-aided painting. **(B)**: Example of a 3D structure, visualized in VAST's 3D viewer. **(C)**: bar graph of the average speed of three machine learning-aided painting modalities together with manual annotation. The test was performed by 3 experienced annotators on the Cerebellum dataset.**(D):** sample visual results of the speed test from one annotator painted in 10 minutes. Top left: volume painted with the 2D segmentation in tandem with VAST's pen mode. Top right: volume painted with the 2D segmentation together with VAST's fill mode. Bottom left: volume painted when membrane detections are used with VAST's paint mode. Bottom right: volume painted with manual annotation only.

in order to enhance the learning abilities of the network. In Table 1 we show a brief summary of the salient points we reckoned important for a user-friendly software tool compared across the packages most similar to mEMbrain.

7 DISCUSSION AND OUTLOOK

Here, we presented a software tool - mEMbrain - which provides a solution for carrying out semi-automatic CNN-based segmentation of electron microscopy datasets. Importantly, our package does not require any installation, and it does not assume any prior experience of the user in coding. mEMbrain works synergistically with VAST, a widely used annotation and segmentation tool in the connectomics community (Berger et al., 2018). Our hope is that VAST users will be enabled in their reconstructions thanks to mEMbrain.

	mEMbrain	**SegEM**	**Uni-EM**
Language	Matlab	Matlab	python
Has GUI	Yes	No	Yes
Contains all the steps for the segmentation pipeline	Yes	No	Yes
Trains 2D networks	Yes	Yes	Yes
Trains 3D networks	No	Yes	Yes
Predicts on-the-fly	Yes	No	No
Designed to train locally	Yes	No	Yes

Table 1. Summary of the comparison between the state-of-the-art (semi) automatic segmentation pipelines in connectomics. The qualities in the various rows represent some of the parameters we deemed important when designing mEMbrain.

Our tool compares favorably to other similar published software tools. One feature that we hope to incorporate in future editions of mEMbrain is the possibility to train on state-of-the-art 3D CNNs, such as 3C (Meirovitch et al., 2019), thereby allowing for better results. Nevertheless, it is important to note that 2D section segmentation can provide satisfactory results, depending on the quality of the sample staining and the dataset alignment.

One of the main motivations for coding mEMbrain was its capability for processing datasets and running deep learning algorithms on local computers. Although at first sight this may appear as a set-back, it represents a tangible means for affordable connectomics by abolishing the costs for expensive clusters. Furthermore, it avoids the need of transferring massive datasets in different locations, which results in a gain in terms of time, and allows for a rapid validation of results due to its close dialogue with VAST. Many of the results showed in this paper were obtained by using a single Nvidia GPU RTX 2080 Ti. Thus, with the current technology the use of mEMbrain is best when the dataset is within the terabyte range. To this end, a useful extension of the toolbox would be to allow the possibility of predicting on a computing cluster when the user necessitates it. Moreover, we noticed how the main bottleneck of the predicting time is created by MATLAB reading chunks of data from VAST. Therefore another possible future direction is to allow for the prediction of multiple classifiers at the same time (e.g. co-prediction of mitochondria and vesicles) in order to avoid reading the dataset multiple times. Nevertheless, it is foreseeable that the available technology will improve, and with it also the prediction time with mEMbrain.

Finally, from the locality of our solution stems the exciting opportunity to place the segmentation step of the connectomics pipeline next to the scope, and to readily predict each tile scanned by the electron microscope, allowing researchers to access their on-the-fly reconstruction in a more timely fashion (Lichtman et al., 2014).

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

E.C.P., Y.M. and J.W.L. conceptualized the idea of mEMbrain. E.C.P. and Y.M. coded the first prototype of the software. E.Y. and Y.M. wrote the code for the current version of the software. F.B., B.H., X.L, N.D., Y.M., M.W., M.Z., B.D., P.S.K. and F.Y. generated ground truth, tested and evaluated mEMbrain on different datasets, provided figures for the manuscript. N.D. conducted the machine learning speed up test

and Y.M. analyzed the results. M.B. pioneered the work on the skeleton network under supervision of Y.M.. E.C.P. wrote the manuscript, with input and help from all the other co-authors.

FUNDING

This work was supported by NIH grants U19NS104653, P50MH094271 and 1UG3MH12338601A1.

ACKNOWLEDGMENTS

The authors would like to thank Anna Steyer, Sebastian Britz, Sebastian Markert, Yannick Schwab and Christian Stigloher for generating the *C. elegans* FIB-SEM dataset. The authors thank Daniel R. Berger for his assistance in interfacing MATLAB codes with VAST.

SUPPLEMENTAL DATA

-

DATA AVAILABILITY STATEMENT

The code generated for this study can be found in the mEMbrain repository [https://github.com/emmay78/mEMbrain-new].

REFERENCES

[Dataset] (2022). Using matlab with tensorflow and pytorch for deep learning

Abbott, L. F., Bock, D. D., Callaway, E. M., Denk, W., Dulac, C., Fairhall, A. L., et al. (2020). The mind of a mouse. *Cell* 182, 1372–1376

Beier, T., Pape, C., Rahaman, N., Prange, T., Berg, S., Bock, D. D., et al. (2017). Multicut brings automated neurite segmentation closer to human performance. *Nature methods* 14, 101–102

Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., et al. (2019). Ilastik: interactive machine learning for (bio) image analysis. *Nature Methods* 16, 1226–1232

Berger, D. R., Seung, H. S., and Lichtman, J. W. (2018). Vast (volume annotation and segmentation tool): efficient manual and semi-automatic labeling of large 3d image stacks. *Frontiers in neural circuits* 12, 88

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

Bidel, F., Meirovitch, Y., Schalek, R. L., Lu, X., Pavarino, E. C., Yang, F., et al. (2022). Connectomics of the octopus vulgaris vertical lobe provides insight into conserved and novel principles of a memory acquisition network. *bioRxiv*

Bock, D. D., Lee, W.-C. A., Kerlin, A. M., Andermann, M. L., Hood, G., Wetzel, A. W., et al. (2011). Network anatomy and in vivo physiology of visual cortical neurons. *Nature* 471, 177–182

Boergens, K. M., Berning, M., Bocklisch, T., Bräunlein, D., Drawitsch, F., Frohnhofen, J., et al. (2017). webknossos: efficient online 3d data annotation for connectomics. *nature methods* 14, 691–694

Britz, S., Markert, S. M., Witvliet, D., Steyer, A. M., Tröger, S., Mulcahy, B., et al. (2021). Structural analysis of the caenorhabditis elegans dauer larval anterior sensilla by focused ion beam-scanning electron microscopy. *Frontiers in neuroanatomy* 15

Cassada, R. C. and Russell, R. L. (1975). The dauerlarva, a post-embryonic developmental variant of the nematode caenorhabditis elegans. *Developmental biology* 46, 326–342

Ciresan, D., Giusti, A., Gambardella, L., and Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in neural information processing systems* 25

Denk, W., Briggman, K. L., and Helmstaedter, M. (2012). Structural neurobiology: missing link to a mechanistic understanding of neural computation. *Nature Reviews Neuroscience* 13, 351–358

Drescher, B. R., Meirovitch, Y., Tait, C., Sant, H. H., Pavarino, E. C., Schalek, R., et al. (2021). Poster: Ultrastructure of the chemosensory rhinophore ganglion and its connective in the nudibranch mollusc berghia stephanieae reveals an unexpectedly complex anatomical organization. In *Society for Neurocience*. –

Fiala, J. C. (2005). Reconstruct: a free editor for serial section microscopy. *Journal of microscopy* 218, 52–61

Funke, J., Tschopp, F., Grisaitis, W., Sheridan, A., Singh, C., Saalfeld, S., et al. (2018). Large scale image segmentation with structured loss based deep learning for connectome reconstruction. *IEEE transactions on pattern analysis and machine intelligence* 41, 1669–1680

Hayworth, K. J., Morgan, J. L., Schalek, R., Berger, D. R., Hildebrand, D. G., and Lichtman, J. W. (2014). Imaging atum ultrathin section libraries with wafermapper: a multi-scale approach to em reconstruction of neural circuits. *Frontiers in neural circuits* 8, 68

Hayworth, K. J., Xu, C. S., Lu, Z., Knott, G. W., Fetter, R. D., Tapia, J. C., et al. (2015). Ultrastructurally smooth thick partitioning and volume stitching for large-scale connectomics. *Nature methods* 12, 319–322

Helmstaedter, M., Briggman, K. L., and Denk, W. (2011). High-accuracy neurite reconstruction for high-throughput neuroanatomy. *Nature neuroscience* 14, 1081–1088

Helmstaedter, M., Briggman, K. L., Turaga, S. C., Jain, V., Seung, H. S., and Denk, W. (2013). Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature* 500, 168–174

Hochner, B. (2012). An embodied view of octopus neurobiology. *Current biology* 22, R887–R892

Januszewski, M., Kornfeld, J., Li, P. H., Pope, A., Blakely, T., Lindsey, L., et al. (2018). High-precision automated reconstruction of neurons with flood-filling networks. *Nature methods* 15, 605–610

Kasthuri, N., Hayworth, K. J., Berger, D. R., Schalek, R. L., Conchello, J. A., Knowles-Barley, S., et al. (2015). Saturated reconstruction of a volume of neocortex. *Cell* 162, 648–661

Klinger, E., Motta, A., Marr, C., Theis, F. J., and Helmstaedter, M. (2021). Cellular connectomes as arbiters of local circuit models in the cerebral cortex. *Nature communications* 12, 1–15

Kuan, A. T., Bondanelli, G., Driscoll, L. N., Han, J., Kim, M., Hildebrand, D. G., et al. (2022). Synaptic wiring motifs in posterior parietal cortex support decision-making. *bioRxiv*

Lee, K., Zlateski, A., Ashwin, V., and Seung, H. S. (2015). Recursive training of 2d-3d convolutional networks for neuronal boundary prediction. *Advances in Neural Information Processing Systems* 28

Lee, K., Zung, J., Li, P., Jain, V., and Seung, H. S. (2017). Superhuman accuracy on the snemi3d connectomics challenge. *arXiv preprint arXiv:1706.00120*

Lee, W.-C. A., Bonin, V., Reed, M., Graham, B. J., Hood, G., Glattfelder, K., et al. (2016). Anatomy and function of an excitatory network in the visual cortex. *Nature* 532, 370–374

Li, F., Lindsey, J. W., Marin, E. C., Otto, N., Dreher, M., Dempsey, G., et al. (2020). The connectome of the adult drosophila mushroom body provides insights into function. *Elife* 9, e62576

Lichtman, J. W. and Denk, W. (2011). The big and the small: challenges of imaging the brain's circuits. *Science* 334, 618–623

592 Lichtman, J. W., Livet, J., and Sanes, J. R. (2008). A technicolour approach to the connectome. *Nature*
593 *Reviews Neuroscience* 9, 417–422
594 Lichtman, J. W., Pfister, H., and Shavit, N. (2014). The big data challenges of connectomics. *Nature*
595 *neuroscience* 17, 1448–1454
596 Lu, J., Behbahani, A. H., Hamburg, L., Westeinde, E. A., Dawson, P. M., Lyu, C., et al. (2022).
597 Transforming representations of movement from body-to world-centric space. *Nature* 601, 98–104
598 Macrina, T., Lee, K., Lu, R., Turner, N. L., Wu, J., Popovych, S., et al. (2021). Petascale neural circuit
599 reconstruction: automated methods. *bioRxiv*
600 Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., et al. (2018). Deeplabcut:
601 markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience* 21,
602 1281–1289
603 McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The
604 sequential learning problem. In *Psychology of learning and motivation* (Elsevier), vol. 24. 109–165
605 Meirovitch, Y., Kang, K., Draft, R. W., Pavarino, E. C., Echeverri, M. F. H., Yang, F., et al. (2021).
606 Neuromuscular connectomes across development reveal synaptic ordering rules. *bioRxiv*
607 Meirovitch, Y., Matveev, A., Saribekyan, H., Budden, D., Rolnick, D., Odor, G., et al. (2016). A multi-pass
608 approach to large-scale connectomics. *arXiv preprint arXiv:1612.02120*
609 Meirovitch, Y., Mi, L., Saribekyan, H., Matveev, A., Rolnick, D., and Shavit, N. (2019). Cross-classification
610 clustering: An efficient multi-object tracking technique for 3-d instance segmentation in connectomics.
611 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8425–8435
612 Morgan, J. L. and Lichtman, J. W. (2013). Why not connectomics? *Nature methods* 10, 494–500
613 Odstrcil, I., Petkova, M. D., Haesemeyer, M., Boulanger-Weill, J., Nikitchenko, M., Gagnon, J. A.,
614 et al. (2022). Functional and ultrastructural analysis of reafferent mechanosensation in larval zebrafish.
615 *Current Biology* 32, 176–189
616 Pachitariu, M., Steinmetz, N. A., Kadir, S. N., Carandini, M., and Harris, K. D. (2016). Fast and accurate
617 spike sorting of high-channel count probes with kilosort. *Advances in neural information processing*
618 *systems* 29
619 Pachitariu, M., Stringer, C., Dipoppa, M., Schröder, S., Rossi, L. F., Dalgleish, H., et al. (2017). Suite2p:
620 beyond 10,000 neurons with standard two-photon microscopy. *BioRxiv*
621 Roels, J., Hennies, J., Saeys, Y., Philips, W., and Kreshuk, A. (2019). Domain adaptive segmentation
622 in volume electron microscopy imaging. In *2019 IEEE 16th International Symposium on Biomedical*
623 *Imaging (ISBI 2019)* (IEEE), 1519–1522
624 Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical
625 image segmentation. In *International Conference on Medical image computing and computer-assisted*
626 *intervention* (Springer), 234–241
627 Salas, P., Vinaithirthan, V., Newman-Smith, E., Kourakis, M. J., and Smith, W. C. (2018). Photoreceptor
628 specialization and the visuomotor repertoire of the primitive chordate ciona. *Journal of Experimental*
629 *Biology* 221, jeb177972
630 Scheffer, L. K., Xu, C. S., Januszewski, M., Lu, Z., Takemura, S.-y., Hayworth, K. J., et al. (2020). A
631 connectome and analysis of the adult drosophila central brain. *Elife* 9, e57443
632 Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: an
633 open-source platform for biological-image analysis. *Nature methods* 9, 676–682
634 Schneider, C. A., Rasband, W. S., and Eliceiri, K. W. (2012). Nih image to imagej: 25 years of image
635 analysis. *Nature methods* 9, 671–675

Shapson-Coe, A., Januszewski, M., Berger, D. R., Pope, A., Wu, Y., Blakely, T., et al. (2021). A connectomic study of a petascale fragment of human cerebral cortex. *BioRxiv*

Shomrat, T., Zarrella, I., Fiorito, G., and Hochner, B. (2008). The octopus vertical lobe modulates short-term learning rate and uses ltp to acquire long-term memory. *Current Biology* 18, 337–342

Strata, P., Provini, L., and Redman, S. (2012). On the concept of spinocerebellum. *Proceedings of the National Academy of Sciences* 109, E622–E622

Stringer, C., Wang, T., Michaelos, M., and Pachitariu, M. (2021). Cellpose: a generalist algorithm for cellular segmentation. *Nature methods* 18, 100–106

Szigeti, B., Gleeson, P., Vella, M., Khayrulin, S., Palyanov, A., Hokanson, J., et al. (2014). Openworm: an open-science approach to modeling caenorhabditis elegans. *Frontiers in computational neuroscience* 8, 137

Tapia, J. C., Wylie, J. D., Kasthuri, N., Hayworth, K. J., Schalek, R., Berger, D. R., et al. (2012). Pervasive synaptic branch removal in the mammalian neuromuscular system at birth. *Neuron* 74, 816–829

Turaga, S. C., Murray, J. F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., et al. (2010). Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural computation* 22, 511–538

Turchetti-Maia, A., Shomrat, T., and Hochner, B. (2017). The vertical lobe of cephalopods. In *The Oxford Handbook of Invertebrate Neurobiology*

Turner, N. L., Macrina, T., Bae, J. A., Yang, R., Wilson, A. M., Schneider-Mizell, C., et al. (2022). Reconstruction of neocortex: Organelles, compartments, cells, circuits, and activity. *Cell* 185, 1082–1100

Urakubo, H., Bullmann, T., Kubota, Y., Oba, S., and Ishii, S. (2019). Uni-em: an environment for deep neural network-based automated segmentation of neuronal electron microscopic images. *Scientific reports* 9, 1–9

Verasztó, C., Jasek, S., Gühmann, M., Shahidi, R., Ueda, N., Beard, J. D., et al. (2020). Whole-animal connectome and cell-type complement of the three-segmented platynereis dumerilii larva. *BioRxiv*

Verasztó, C., Ueda, N., Bezares-Calderón, L. A., Panzera, A., Williams, E. A., Shahidi, R., et al. (2017). Ciliomotor circuitry underlying whole-body coordination of ciliary activity in the platynereis larva. *Elife* 6, e26000

White, J. G., Southgate, E., Thomson, J. N., Brenner, S., et al. (1986). The structure of the nervous system of the nematode caenorhabditis elegans. *Philos Trans R Soc Lond B Biol Sci* 314, 1–340

Wilson, A. M., Schalek, R., Suissa-Peleg, A., Jones, T. R., Knowles-Barley, S., Pfister, H., et al. (2019). Developmental rewiring between cerebellar climbing fibers and purkinje cells begins with positive feedback synapse addition. *Cell reports* 29, 2849–2861

Wiltschko, A. B., Tsukahara, T., Zeine, A., Anyoha, R., Gillis, W. F., Markowitz, J. E., et al. (2020). Revealing the structure of pharmacobehavioral space through motion sequencing. *Nature neuroscience* 23, 1433–1443

Witvliet, D., Mulcahy, B., Mitchell, J. K., Meirovitch, Y., Berger, D. R., Wu, Y., et al. (2021). Connectomes across development reveal principles of brain maturation. *Nature* 596, 257–261

Wolf, S., Pape, C., Bailoni, A., Rahaman, N., Kreshuk, A., Kothe, U., et al. (2018). The mutex watershed: efficient, parameter-free image partitioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 546–562

Zhao, T., Olbris, D. J., Yu, Y., and Plaza, S. M. (2018). Neutu: software for collaborative, large-scale, segmentation-based connectome reconstruction. *Frontiers in Neural Circuits* 12, 101

Zullo, L. and Hochner, B. (2011). A new perspective on the organization of an invertebrate brain. *Communicative & integrative biology* 4, 26–29