



## Neuronal population reconstruction in ultra-scale optical microscopy images via progressive learning

Jie Zhao, Xuejin Chen\*, Zhiwei Xiong, Dong Liu, Junjie Zeng, Yueyi Zhang, Zheng-Jun Zha, Guoqiang Bi, Feng Wu

National Engineering Laboratory for Brain-inspired Intelligence Technology and Application, University of Science and Technology of China, No. 96 Jinzhai Road, Hefei, Anhui, 230026, China

---

### ARTICLE INFO

#### *Article history:*

**Keywords:** Neuronal population reconstruction, Ultra-scale images, Optical microscopy, Progressive learning

---

### ABSTRACT

Reconstruction of neuronal populations from ultra-scale optical microscopy (OM) images is essential to investigate neuronal circuits and brain mechanisms. The noises, low contrast, huge memory requirement and high computational cost pose significant challenges in the neuronal population reconstruction. Recently, many studies have been conducted to extract neuron signals using deep neural networks (DNNs). However, training such DNNs usually relies on a huge amount of voxel-wise annotations in OM images, which are expensive in terms of both finance and labor. In this paper, we propose a novel framework for dense neuronal population reconstruction in ultra-scale images. To solve the problem of high cost in obtaining manual annotation for training DNNs, we propose a progressive learning scheme for neuronal population reconstruction (PLNPR) which does not require any manual annotations. Our PLNPR scheme consists of a traditional neuron tracing module and a deep segmentation network that mutually complement and progressively promote each other. To reconstruct dense neuronal populations in a terabyte-sized ultra-scale image, we introduce an automatic framework which adaptively traces neurons block by block and fuses multiple neurons in overlapped regions continuously and smoothly. We build a dataset “VISoR-40” which consists of 40 large-scale OM image blocks from cortical regions of a mouse. Extensive experimental results on our VISoR-40 dataset and the public BigNeuron dataset demonstrate the effectiveness and superiority of our method on neuronal population reconstruction and single neuron reconstruction. Furthermore, we successfully apply our method to reconstruct dense neuronal populations in an ultra-scale mouse brain slice. The proposed adaptive block-wise reconstruction and fusion strategy greatly improve the completeness of neurites in dense neuronal population reconstruction.

© 2020 Elsevier B. V. All rights reserved.

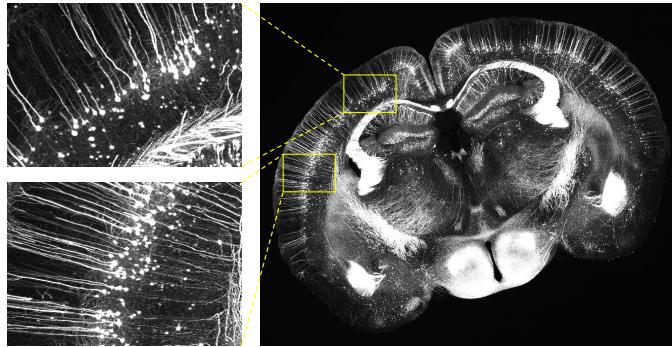
---

### 1. Introduction

A blue print of the brain architecture, including the morphology and interconnectivity of neuronal populations, allows

for measuring and visualizing neuronal structure, understanding neuronal identity, and determining potential connectivity. Therefore, complete reconstruction of neuronal populations from large-scale brain images is essential to investigate the mechanism of the nervous system, analyze brain changes, and facilitate our understanding of brain diseases such as dementia and Alzheimer's disease Petrella et al. (2003); Giorgio and De Stefano (2013). One of the key techniques in this endeavor

\*Corresponding author: Tel.: +86-551-63600852;  
e-mail: [xjchen99@ustc.edu.cn](mailto:xjchen99@ustc.edu.cn) (Xuejin Chen)



**Fig. 1.** A 3D OM image for a mouse brain slice captured by the VISoR imaging system Wang et al. (2019). The large density of neurons, low contrast, image noises and huge volume pose significant challenges for automatic reconstruction of neuronal populations from the ultra-scale image.

is optical microscopy (OM), which allows detailed visualization of neurons and makes the reconstruction of every neuron possible Senft (2011), as shown in Fig. 1. However, despite numerous efforts devoted, this task is still one of the main challenges in computational neuroscience.

The challenges in large-scale neuronal population reconstruction are mainly caused by complex morphology of neurons, low quality and large volume of OM brain images. Due to the complicated process of imaging acquisition and the uneven distribution of fluorescence markers in neurons, the voxel intensities vary dramatically in highly noisy and inhomogeneous environments. Moreover, to detail the structure of neurons in the brain, OM images in high resolution are required. Such an image typically contains trillions of voxels. It is even impractical to directly load the entire image into a computer memory before reconstructing neurons. In the other hand, manual reconstruction of neuronal populations from ultra-scale brain images is an extremely laborious and time-consuming task as well. Moreover, sophisticated knowledge of neuron morphology is also required for manual reconstruction. Therefore, an effective and automatic reconstruction algorithm for large-scale neuronal population in these challenging situations is greatly desired in practice.

Attempts have been made for neuron reconstruction for large-scale OM images in recent years, such as Neuron Crawler Zhou et al. (2015), UltraTracer Peng et al. (2017) and MEIT Wang et al. (2018). A common solution is to divide the large-scale image into blocks and then trace neurons block by block. Despite their great improvements in this task, several limitations still remain. One main bottleneck is that the base tracers used for neuron tracing in image blocks typically employ a series of traditional image processing algorithms such as binarization, fast marching, fusion, etc. Unfortunately, these algorithms using hand-crafted features and rules have difficulty in reconstructing neurons from low-contrast and noisy image blocks. To improve the reconstruction quality, deep learning techniques have been adopted in neuron reconstruction recently Xu et al. (2016); Li et al. (2017); Zhou et al. (2018); Kozinski et al. (2020). However, these approaches require huge amount of manually annotated data for neuron segmentation network training.

Compared with manual annotation, the reconstruction results by conventional algorithms effectively provide approximate locations of neurons. Although these voxels may not cover all neurons precisely, they provide important cues for obtaining complicated patterns of neurons. Therefore, we propose a progressive learning scheme for neuronal population reconstruction (PLNPR) to take advantage of both conventional methods and deep learning techniques. More specifically, we employ conventional methods to produce pseudo-labels for training a deep neural network for neuron segmentation. The network is expected to learn more comprehensive features of neurons from noisy labels. With a more powerful network for segmenting neurons, the neuron reconstruction using conventional tracers could be improved. Then we progressively refine the network with better neuron reconstruction results as pseudo labels, and reconstruct more complete neurons with better neuron segmentation. We first investigate this concept in our preliminary work Zhao et al. (2019) on a few image blocks. In this work, we apply PLNPR on dense neuron population reconstruction in a ultra-scale OM image slice.

Another limitation of existing large-scale neuron reconstruction methods Zhou et al. (2015); Peng et al. (2017); Wang et al. (2018) is that they mainly focus on single neuron reconstruction. For dense neuronal population reconstruction in ultra-scale OM images, dense neurites may cross with each other, making the neuron tracing much more challenging in low-contrast and noisy OM images. Following the commonly used block-by-block framework, we introduce UltraNPR, which utilizes our PLNPR approach as the base tracer in blocks, and design a novel tracing and fusion strategy for dense neuronal populations. We start local reconstruction using our PLNPR in the blocks where soma can be detected, and propagate a group of neurite tips as pseudo somas for neighboring blocks to trace the neuron population. There are typically over-tracing and topological discrepancy in the reconstructed neurites in adjacent blocks. We design a fusion method with spatially varying confidences in order eliminate the tracing errors usually occur at boundary regions of local blocks.

In summary, we propose PLNPR, a progressive learning framework that integrates traditional neuron tracing methods and deep segmentation networks for neuron population reconstruction without using manual annotations. We integrate our PLNPR with a block-wise propagation and fusion strategy which can reconstruct dense neuronal populations from trillions of voxels in ultra-scale OM images. In order to evaluate our method, we build a dataset “VISoR-40” which consists of 40 OM image blocks from mouse cortical regions. Manual annotations of eight blocks in the dataset are available for the community. Extensive experiments on our VISoR-40 dataset and the BigNeuron dataset demonstrate the effectiveness and superiority of our progressive learning algorithm for both neuronal population reconstruction and single neuron reconstruction.

## 2. Related Work

Early techniques for neuron reconstruction from optical microscopy images typically employ traditional image processing algorithms, such as snakes Wang et al. (2011); Cai et al.

(2006), principal curves Bas and Erdogmus (2011), graph theory Peng et al. (2010); Yang et al. (2013); De et al. (2016), model-fitting Zhao et al. (2011); Santamaría-Pang et al. (2015), watershed Navlakha et al. (2013); Sümbül et al. (2016), energy minimization Quan et al. (2013); Liu et al. (2016), mean-shift clustering Frasconi et al. (2014), ray-shooting Wu et al. (2014); Liu et al. (2019), fast-marching Peng et al. (2011); Xiao and Peng (2013); Liu et al. (2018) and so on. Unfortunately, these conventional algorithms rely on hand-crafted features and carefully tuned parameters, and usually tend to fail when the image quality is poor.

To improve the reconstruction performance from low-quality image blocks, many machine learning techniques have been introduced to extract neuron voxels from noisy and low-contrast images for further reconstruction. This kind of methods employs various classifiers with hand-crafted features, such as support vector machine (SVM) Chen et al. (2015), minimum spanning tree Basu et al. (2016), Bayesian probabilistic model Radojević and Meijering (2017), Bootstrap aggregating Wang et al. (2017), gradient boosting decision trees (GBDT) Gu et al. (2017), Markov chain Monte Carlo (MCMC) Skibbe et al. (2015); Skibbe et al. (2019), and so on. However, these methods employs hand-crafted features that usually suffer from limited representation capability for accurate recognition, considering more challenging and complex image blocks.

Recently, deep-learning-based methods Li et al. (2017); Zhou et al. (2018); Xu et al. (2016); Kozinski et al. (2020) bring the power of DNNs to improve the reconstruction performance. Instead of manually designing sophisticated features, these DNNs learn feature representations in a data-driven way and extract more distinctive features. With more complicated classifiers employed to segment neuron voxels from image blocks, these methods achieve more robust reconstruction results. Despite great improvements, these DNN-based methods rely on extensive manual annotations of neuron voxels for network training. Unfortunately, due to the complicated morphology of neurons and the low quality of OM images, such annotations are very costly to obtain in terms of both time and labor. In comparison, we propose a novel iterative framework to progressively improve the 3D DNN-based neuron reconstruction performance without using manual annotations.

Despite substantial advancements brought by these methods, they often need to load all the voxels into memory. The sheer volume of a large-scale OM image is usually far beyond the processing capability, especially on the memory cost and tracing time. In recent years, some attempts have been made to reconstruct neurons from large-scale OM images, such as Neuron Crawler Zhou et al. (2015), UltraTracer Peng et al. (2017) and MEIT Wang et al. (2018). To tackle the challenge of huge image volume, a common solution is to reconstruct neuronal morphology block by block. In each local block cropped from a large-scale image, existing tracing methods, such as APP2 Xiao and Peng (2013), MOST Wu et al. (2014) and FMST Yang et al. (2019), can be directly used as the base tracer to trace neurites. Starting from the cell body, the neurites are then traced in neighboring blocks and then fused together based on signal strengths and structure continuity. However, all of these

methods focus on single neuron reconstruction. The input images usually contain only one single neuron so that the signal is very sparse without confusion of different neurons. In comparison, we target a more challenging task to reconstruct dense neuronal population in ultra-scale OM images, where closely spaced neurites that belong to different neurons are difficult be distinguished for existing methods that are designed for single neuron reconstruction.

### 3. Proposed Method

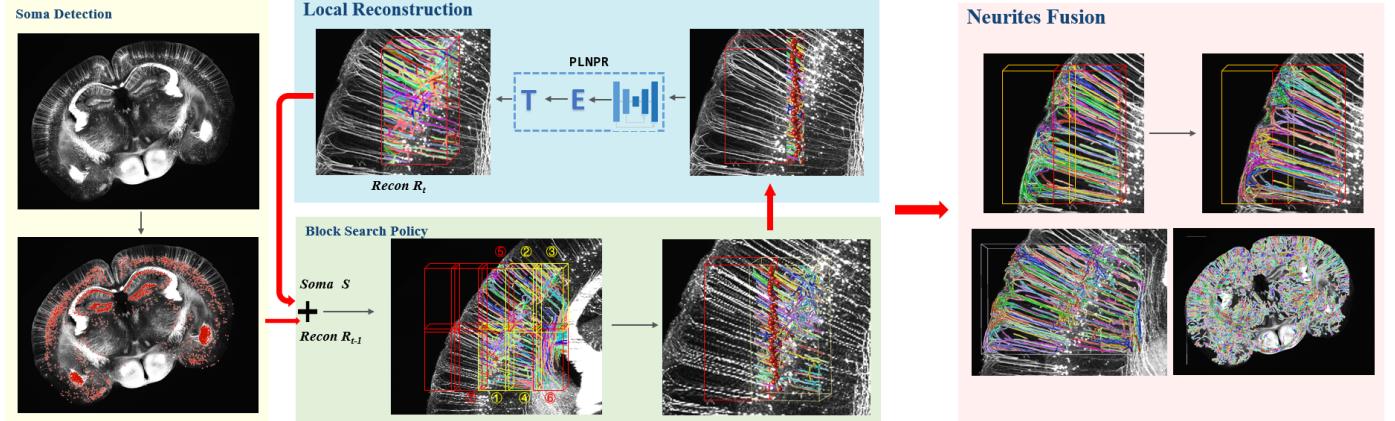
Given a large-scale noisy OM image, we follow a block-by-block reconstruction scheme, as Fig. 2 shows. The ultra-scale input image may contain billions or even trillions of voxels. Therefore, we first divide the input image into blocks that are averagely in size of  $0.5\text{mm} \times 1\text{mm} \times 0.25\text{mm}$  with some overlaps. For each block, we first enhance noisy image signals by deep neural network, which is trained by progressive learning from reconstructed neurons using traditional tracing algorithms. Then we reconstruct neurons in each block using the enhanced image. Since it is more reliably to reconstruct dense neuronal populations starting from somas rather than subtle neurites, our UltraNPR employs an effective block search strategy to trace dense neuronal populations in an adaptive order. Finally, we fuse overlapped neurites in adjacent blocks to reconstruct complete neuronal populations for the whole large-scale image.

#### 3.1. PLNPR for Robust Neuron Reconstruction

To reconstruct neurons in a noisy OM image block, our PLNPR algorithm consists of three key components: a segmentation network, an image enhancement module and a neuron tracing module, as shown in Fig. 3. The segmentation network is designed to extract neuron voxels from noisy and complex backgrounds. Compared with existing segmentation networks are trained with dense voxel-wise annotations for strong supervision, we use pseudo labels that are generated using traditional neuron tracing methods. In each iteration of segmentation and reconstruction, we apply the NGPST Quan et al. (2015) as the neuron tracing module to reconstruct neurons from image blocks. The tracing module can be replaced by any other tracing method that does not require manual annotations for training. It takes an image block  $\mathbf{B}$  as input and reconstructs a neuronal population with separated neurons. From the reconstruction results, we produce a binary mask  $\mathbf{M}$  indicating foreground by  $\mathbf{M}(x) = 1$  and background by  $\mathbf{M}(x) = 0$  for a voxel  $x$ .

Given  $N$  unlabeled image blocks, we train our segmentation network using the neuron masks  $\{\mathbf{M}_i, i = 1, \dots, N\}$  generated from the neuron tracing module. The output of the neuron segmentation network is a 3D probability map  $\mathbf{P}$ , which is computed by a voxel-wise softmax activation function.  $\mathbf{P}(x) \in [0, 1]$  indicating the probability of a voxel  $x$  to be a neuron part.

Then, by fusing the predicted probability map with the raw signals, the image block is further enhanced in order to preserve both local signals and global structures simultaneously. When the enhanced block is fed into the neuron tracing module, more complete neuronal populations can be reconstructed and provide better pseudo labels for the next iteration of network



**Fig. 2. Diagram of our UltraNPR algorithm for neuronal population reconstruction in a large-scale brain slice.**

training. Based on the iterative learning process, the powerful DNNs and tracing methods mutually complement and promote each other to gradually improve the neuron reconstruction performance.

### 3.1.1. DNN for 3D Neuron Segmentation

Extracting neuron voxels from image blocks is not a trivial problem since the size, morphology and intensity of neurons vary significantly. In recent years, many 3D DNNs, such as 3D U-Net Çiçek et al. (2016), 3D DSN Dou et al. (2017) and DenseVoxNet Yu et al. (2017) have demonstrated an outstanding capability in various biological and biomedical image segmentation tasks. Therefore, we take advantage of 3D segmentation networks to extract more representative features to meet the challenges of neuron segmentation. In this work, we extend the 3D DSN Dou et al. (2017) as our neuron segmentation network to balance the performance and computation burden.

Although the original 3D DSN has achieved excellent performance for 3D organ segmentation Dou et al. (2017), it is still prone to overfitting in our case due to the limited training data. We employ the Dropout Srivastava et al. (2014) technique during training to learn more robust features that better generalize to new data. In each convolutional layer, the dropout with a rate of 0.5 is applied in our network.

Another challenge of training 3D DNN is the memory limitation because the 3D feature images are huge with respect to the input size. Therefore, for each input image block, we crop a group of small cubes in size of  $160 \times 160 \times 160$  with 30% overlaps, and set batch size to 1 during training. To have the same physical resolution with the lateral dimension in OM image blocks, voxels in the axial dimension are interpolated after the imaging process. However, this interpolation makes the image quality along different dimensions inhomogeneous. Therefore, a random transposition process is employed for each cube as data augmentation for network training.

In addition, the volume of neuron (foreground) voxels is usually much smaller than that of background in an OM image. To cope with this imbalance problem, a data balancing technique is introduced for network training. Specifically, when computing the training loss, we only consider the neuron voxels and a certain portion of background voxels, which is randomly selected

as non-neuron samples. The number of non-neuron voxels used for training is set as 10 times that of neuron voxels.

### 3.1.2. Image Enhancement

After training the segmentation network using pseudo labels, we use the trained model to predict a probability map each image cube. By averaging the probabilities of the overlapped voxels between adjacent cubes, we can obtain the probability map  $\mathbf{P}$  for the entire block  $\mathbf{B}$ . Each element in  $\mathbf{P}$  indicates the probability that the corresponding voxel in  $\mathbf{B}$  belongs to the neuron. To utilize the probability map, one natural way is to reconstruct neurons directly from it. However, since the pseudo labels are not as accurate as manual annotations, especially at the early iterations, some local details might lose in the probability map. Therefore, we employ an enhanced representation by fusing the probability map and the raw image block, in order to keep detailed structures and suppress noise signals effectively. Specifically, a new probability map  $\widetilde{\mathbf{P}}$  is first constructed by linearly mapping the value range of  $\mathbf{P} \in [0, 1]$  to the value range  $[b_{min}, b_{max}]$  of  $\mathbf{B}$ .

$$\widetilde{\mathbf{P}}(x) = (b_{max} - b_{min})\mathbf{P}(x). \quad (1)$$

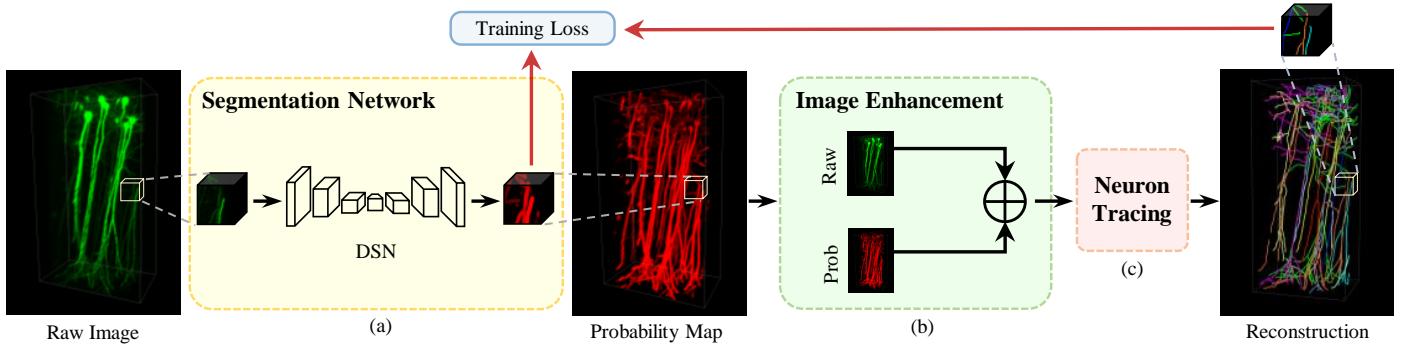
Then, based on the raw signals in block  $\mathbf{B}$  and the probability map  $\widetilde{\mathbf{P}}$ , an enhanced block  $\mathbf{E}$  is computed as

$$\mathbf{E}(x) = \alpha \widetilde{\mathbf{P}}(x) + (1 - \alpha)\mathbf{B}(x), \quad (2)$$

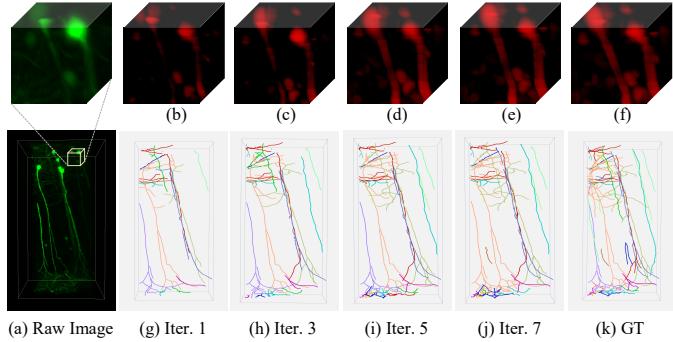
where  $\alpha \in [0, 1]$  is a weight to control the contributions of voxel  $x$  in the original intensity and the probability map. By feeding the enhanced blocks to the neuron tracing module, neuronal populations can be reconstructed more completely.

With more reliable reconstruction results for supervision, the segmentation network could be further trained to learn more discriminative and representative features for producing probability maps, which in turn benefits the tracing module to reconstruct neurons in the next iteration.

As shown in Fig. 4(a), due to the noises and low contrast in the raw image block, the intensity of neuron voxels is inhomogeneous, which makes some neurons subtle. At first, by feeding the raw image to the conventional method Quan et al. (2015),



**Fig. 3.** Diagram of our progressive learning algorithm for neuron reconstruction in an image block. (a) The segmentation network extracts neuron signals from the raw image. (b) The output probability map is employed to enhance the raw image in order to preserve both global structures and local signal details, which facilitates the neuron tracing module (c) for more complete neuronal population reconstruction. To train the segmentation network, we use the reconstructed neurons as pseudo labels (red arrows), and iteratively refine the network learning and neuron reconstruction with a set of images. The black arrows show the reconstruction pass from an image during testing.



**Fig. 4.** Our progressive learning technique gradually improves the segmentation network to extract neuron signals from (a) raw image which has noises and low contrast. (b)(c)(d)(e) The probability maps generated by the segmentation network at different iterations. (f) Combing the probability map and the raw intensity, the enhanced block preserves both global trajectory and local details. (g)(h)(i)(j) More and more complete and accurate reconstruction of the neuronal populations can be obtained with more iterations. (k) The manually labeled neurons are shown for comparison. Separated neurons are shown in different colors.

a neuronal population can be reconstructed. However, compared to the ground truth (GT) shown in Fig. 4(k), the reconstructed neuronal population is incomplete and many neurites are missing, as Fig. 4(g) shows. Then, by utilizing the pseudo labels derived from imperfect reconstruction, the segmentation network can be trained to learn features for global trajectories. Fig. 4(b) shows the predicted probability map, which demonstrates the enhanced trajectories. With more iterations of neuron reconstruction and network training, more distinctive and long-range trajectory features can be progressively captured by the network, as shown in Fig. 4(c)(d)(e). By combining the original image intensities with the predicted probability map, both local signal details and global trajectories are well preserved in the enhanced block, as Fig. 4(f) shows. Iteration by iteration, the completeness and accuracy of neuron reconstruction are progressively improved, as shown in Fig. 4(h)(i)(j).

### 3.2. Large-scale Neuronal Population Reconstruction

Our PLNPR enhances the image signals and traces neurons from each image block. However, for an ultra-scale OM im-

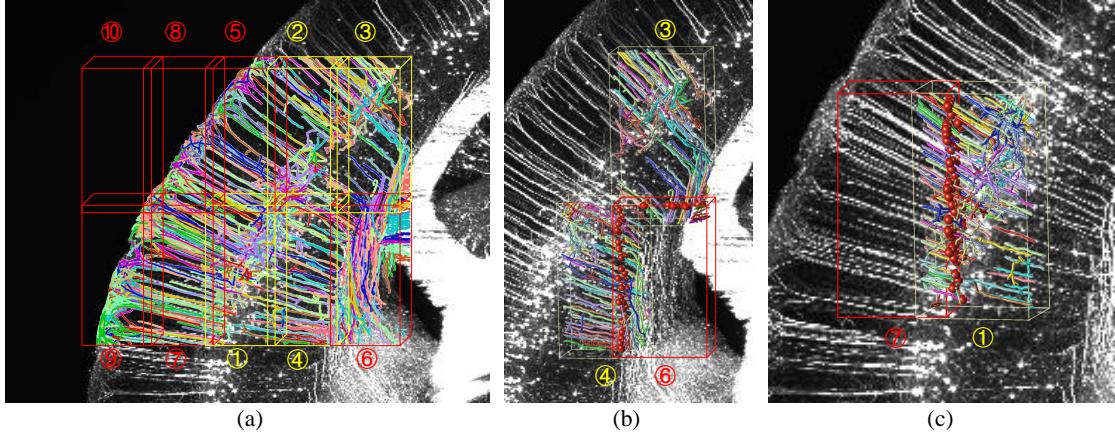
age consisting of a great number of blocks, neurons usually exist across multiple blocks. We propose an effective stitching process by looking for the most continuous block to trace for dense neuron populations from multiple neighboring blocks. As somas are where signals from the dendrites are joined and pass on, the blocks that contain somas are the most probable locations to start the tracing. We start from reconstructing neurons in the blocks where somas can be detected. For blocks where no soma can be detected, we trace the neurons by generating pseudo-somas from the neurite tips from their neighboring blocks. Finally, the neurites reconstructed from local blocks are fused to get complete neurons. Fig. 2 shows the pipeline of our UltraNPR approach.

#### 3.2.1. Initial Soma Detection

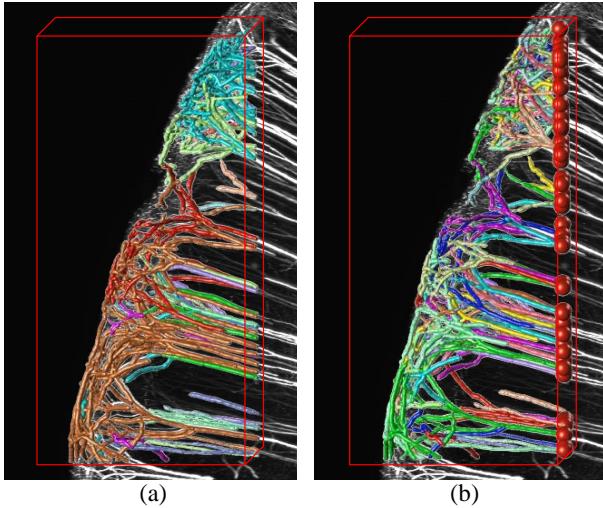
In order to detect somas in an ultra-scale image efficiently, we apply the soma detection algorithm Quan et al. (2013) on each block separately. For each block  $B_i$ , we get a set of somas. Due to the overlap between blocks, somas in the overlapped area would be detected repeatedly. We merge the overlapped somas in adjacent blocks by averaging their position and radius and get a set of somas  $S$  in the large-scale image, as shown in Fig. 4 of the supplementary file.

#### 3.2.2. Block Search and Local Reconstruction

After soma detection, if a block  $B_i$  contains somas, we apply our PLNPR to reconstruct neurites and get a set of neurites  $N_i$  in this block, as shown by the blocks in yellow frames in Fig. 5(a). For the remaining unreconstructed blocks, we check its neighboring blocks that have been reconstructed and add the neurite tips from the reconstructed blocks as its pseudo somas. Though NGPST can perform neuron tracing without any somas, it typically fails to separate neurons with dense neurites, as shown in Fig. 6(a). We follow the structure of the reconstructed neurons in the blocks that have been traced by set the neurite tips as pseudo somas for NGPST in the unreconstructed block. The unreconstructed block  $B^*$  which has the largest number of neighboring reconstructed blocks and the largest number of pseudo somas is selected for reconstruction next. More specifically, for each neighboring reconstructed block of  $B^*$ , we col-



**Fig. 5.** An example of the iterative block reconstruction for ten blocks. (a) The number indicate the order in which the ten blocks are traced. The blocks 1, 2, 3, 4 are firstly reconstructed using our PLNPR since there are somas detected in these four blocks. (b) The block-6 is reconstructed by setting the neurite tips (red dots) from its two neighboring blocks (3, 4) as pseudo somas for tracing. (c) The block-7 is reconstructed by setting the neurite tips as pseudo somas from its neighboring block (1).



**Fig. 6.** Comparison of neuronal population reconstruction on an enhanced image block. (a) Using NGPST without using neurite tips as pseudo somas. (b) Using NGPST with neurite tips as pseudo somas. Separated neurons are shown in different colors. **(What about two neurites belong to the same neuron but separate at the boundary?)**

lect the neuronal points that are close to the boundary of  $B^*$ , and use them as the pseudo-somas for growing the neuronal structure in  $B^*$ . After that, we get its neurites set  $\mathcal{N}^*$ . Fig. 5 (b)(c) show two examples of unreconstructed blocks (red) that to be traced from its neighboring blocks (yellow) that have been reconstructed. This block search and reconstruction process continues iteratively until all blocks have been reconstructed.

### 3.2.3. Neurite Fusion in Adjacent Blocks

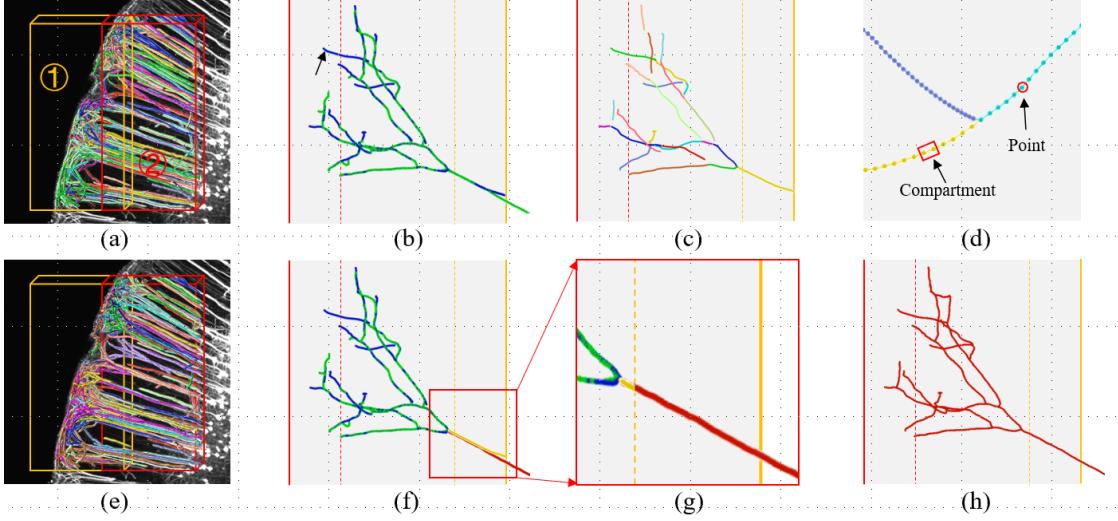
Since neurons would be split into fragmented neurites when dividing the raw image into blocks, we now fuse the neurites from adjacent blocks that are reconstructed separately into connected and complete neurons. Given the reconstructed neurite sets  $\mathcal{N}_a$  and  $\mathcal{N}_b$  in two overlapped blocks  $\mathbf{B}_a$  and  $\mathbf{B}_b$  respectively, for each neurite  $N_a$ , we look for the neurite  $N_b$  which has the largest overlapping volume with  $N_a$ . If the overlapped vol-

ume between them is larger than a threshold  $\delta_{ovlp}$ , we fuse these two neurites together to get a smooth and complete neuron.

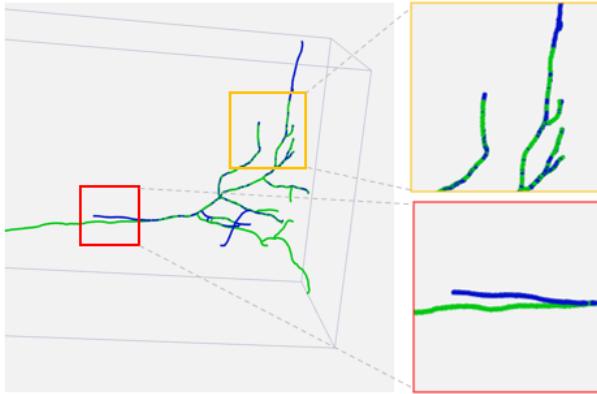
There might be over-tracing and topological discrepancy between two neurites that are reconstructed in two blocks separately, as shown in Fig. 8. Generally, the reconstruction quality near the block boundary is less accurate and reliable than that in the middle of blocks because of less context. Therefore, when fusing two neurites, we tend to keep the neurite segments in the middle of blocks as following.

For two matched neurites, we select the longer one as the reference neurite  $N_a$ , and merge the other neurite  $N_b$  to the reference neurite, as shown in Fig 7 (b). Each neurite is decomposed to a set of neurite branches, as Fig. 7 (c) shows. Each branch is sampled uniformly into a set of fragments, as Fig. 7 (d) shows. For each branch of neurite  $N_b$ , we search a branch which has the largest overlap with it in the reference neurite  $N_a$ , as Fig 7 (f) shows. To fuse the matched two branches, for each point  $p_a$  in branch  $F_a$ , if  $p$  lies in the boundary region of block  $\mathbf{B}_a$  ( $\delta_a$  voxels to the block boundary of  $\mathbf{B}_a$ ), it will be removed and all its child branches are also deleted, as shown in Fig 7 (g). **(where is this operation shown?)** The same deletion operation is performed on the points on branches  $F_b$  that are located in the boundary regions of block  $\mathbf{B}_b$ .

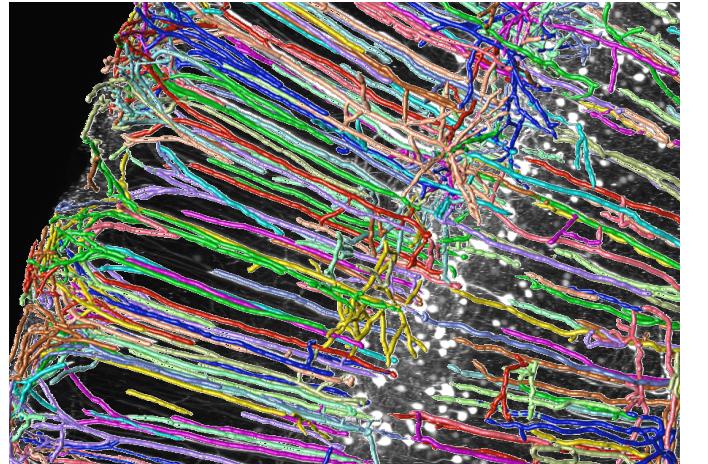
Then, for each remaining point  $p_b$  in branch  $F_b$ , if there is a point  $p_a$  on branch  $F_a$  so that  $dis(p_b, p_a) < \beta$ , it will be removed, as Fig 7 (f) shows. **(i can not find which point is deleted.)** After that, the remaining part of branch  $F_b$  is merged to the reference branch  $F_a$  by connecting it to the nearest point in  $F_a$ . After all matched branches are processed as above, we can obtain the fused neurite  $N_{ab}$ . If there still remains a branch  $F_b$  which is not connected to the neurite  $N_{ab}$ ,  $N_{ab}$  will be updated by assembling the remaining branch. Fig 7 (h) shows the final fusion results of the two neurites. By fusing all neurites in two adjacent blocks, we can obtain more complete reconstruction of dense neuronal populations, as shown in Fig 7 (e).



**Fig. 7.** An example of fusing neurites from two adjacent blocks. (a) The neurites in two adjacent blocks are reconstructed using our PLNPR. For each neurite in the block (1), we look for the neurite in the block (2) that has the largest overlapping volume with it. (b) For two matched neurites, we select the longer one as the reference neurite (green) and merge the other neurite (red) to the reference neurite. (c)



**Fig. 8.** Examples of over-tracing (yellow) and topological discrepancy (red) when assembling two overlapped neurites. These reconstructions are shown in skeleton mode for better visualization.



**Fig. 9.** One example of reconstructing neuronal populations from four adjacent blocks using our method. We can observe that, the fragmented neurites from adjacent blocks are assembled continuously and smoothly.

## 4. Experiments and Results

We evaluate our neuron population reconstruction approach in a large-scale OM image slice of a mouse brain in dimension of  $25397 \times 18516 \times 869$  (761GB), as Fig. 1 shows. The image is captured by the VISO-R imaging system Wang et al. (2019) at a physical resolution of  $0.5 \times 0.5 \times 0.5 \mu\text{m}^3$  per voxel. The image intensity is in 16-bit dynamic range, which preserves sufficient signal details. In order to evaluate our PLNPR method for neuronal reconstruction in local blocks, we first conduct extensive experiments on the VISO-R-40 dataset which we build and the BigNeuron dataset Peng et al. (2015). Then we test our Ultra-NPR algorithm for neuronal population reconstruction in a slice of the mouse brain image.

### 4.1. Evaluation of PLNPR on VISO-R-40 Dataset

#### 4.1.1. VISO-R-40 Dataset

Though many neuron tracing techniques have been proposed, there is no dataset of OM images for dense neu-

ronal population reconstruction. We construct a neuron image dataset ‘‘VISO-R-40’’ (available at <https://braindata.bitahub.com>) for evaluation. The VISO-R-40 dataset consists of 40 OM image blocks cropped from the mouse brain image. The dimension of the blocks ranges from  $419 \times 1197 \times 224$  to  $869 \times 1853 \times 575$ . We randomly select 32 blocks for progressively training the segmentation network in our PLNPR. The remaining 8 blocks with manual annotations are used as the testing data. Each testing block is first labeled manually and independently by two experts. Then, by cross-checking each other’s result, their agreed annotation is approved by an expert (the third one?) to generate the final ground truth.

#### 4.1.2. Experimental Settings and Evaluation Metrics

Pytorch is adopted to implement the DSN model. At each iteration of the progressive learning, the network is trained from scratch with weights initialized from Gaussian distribution with

zero-mean and variance of 0.01. The optimization is realized with the stochastic gradient descent algorithm with the Adam update rule (batch size of 1, weight decay of 0.0005, momentum of 0.9). The base learning rate is set to 0.001 and descended with “poly” learning rate policy (power of 0.9 and the maximum iteration number of 24000). The cube size is set as  $160 \times 160 \times 160$  considering the GPU memory limitation.

To quantitatively evaluate our method, four commonly used metrics defined in Quan et al. (2015), including Precision, Recall, F-Score, and Jaccard, are computed to measure the fidelity between the reconstruction results and the ground truth. Their definitions are defined as follows:

$$\text{Precision}(R, G) = \frac{|R \cap G|}{|R|} = \frac{|TP|}{|R|}, \quad (3)$$

$$\text{Recall}(R, G) = \frac{|R \cap G|}{|G|} = \frac{|TP|}{|G|}, \quad (4)$$

$$F\text{-Score}(R, G) = \frac{2|R \cap G|}{|R| + |G|} = \frac{2|TP|}{|R| + |G|}, \quad (5)$$

$$\text{Jaccard}(R, G) = \frac{|R \cap G|}{|R \cup G|} = \frac{|TP|}{|R \cup G|}, \quad (6)$$

where  $R$  denotes the set of points on the reconstructed neurons,  $G$  denotes the set of neuron points in the ground truth and  $TP$  denotes the set of true positive points,  $|\cdot|$  denotes the number of points in a set. The four metrics are first computed on each individual neuronal tree according to the manually labeled skeleton, and then averaged in a neuronal population weighted by the total length of the neuronal processes of each neuron, the same as Quan et al. (2015).

#### 4.1.3. Progressive Learning

The key idea of PLNPR is to progressively improve the performance of neuron reconstruction by making the neuron segmentation network and the conventional tracing method complementary and synergistic without using any manual annotations. In order to demonstrate the performance improvement, four widely-used tracing methods, including APP1 Peng et al. (2011), APP2 Xiao and Peng (2013), MOSTWu et al. (2014) and NGPST Quan et al. (2015), are tested as the neuron tracing module in our framework. We use their implementations in the software Vaa3D Peng et al. (2014). Eight iterations are tested on our VISoR-40 dataset, and the improvement of neuronal population reconstruction is shown in Fig. 10. We only show the F-Score which is widely used to reflect the overall performance of neuron reconstruction. Moreover, the neuron reconstruction performance on a testing block at different iterations are shown in the Fig. 11 and Table 1. More qualitative and quantitative results are reported in the supplementary materials. The results show that our progressive learning strategy effectively facilitates conventional tracing methods to reconstruct more complete neuronal populations. In addition, the performance improvement gets stable about five iterations of the progressive learning for all the tested tracing methods.

#### 4.1.4. Neuron Segmentation Network

To further verify the effectiveness and robustness of our progressive learning strategy, we test three commonly used deep

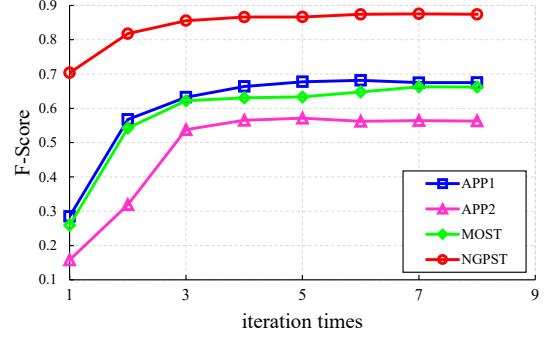


Fig. 10. Progressively improved F-score of neuron reconstruction results under eight iterations in our progressive learning framework using four neuron tracing methods.

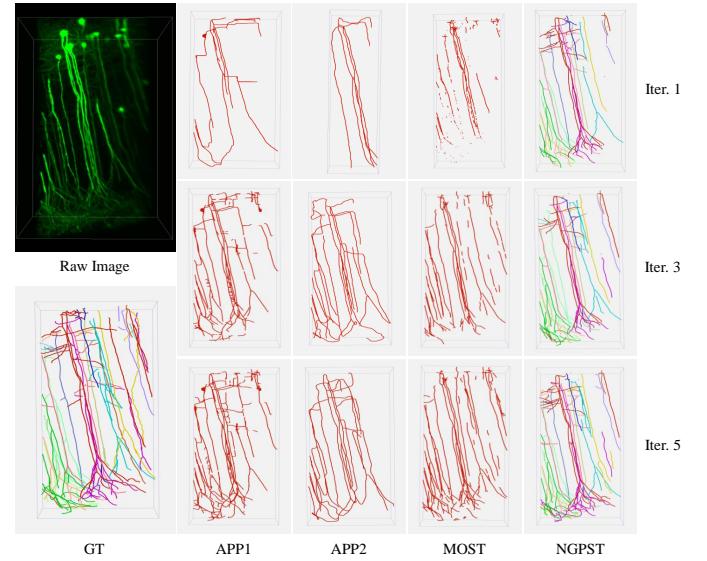


Fig. 11. Neuronal population reconstruction results of a test block at different iterations (top to bottom) using four neuron tracing methods.

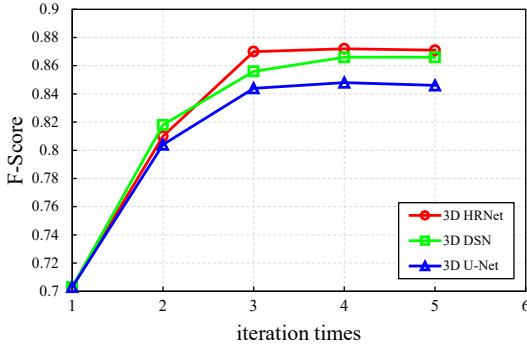
segmentation networks, including 3D DSN Dou et al. (2017), 3D U-Net Çiçek et al. (2016) and a 3D version of HRNet Sun et al. (2019), for generating the neuron probability map. Five iterations are tested on our VISoR-40 dataset, and the F-Score improvement of reconstruction results is shown in the Fig. 12. It can be seen that our PLNPR algorithm effectively improves the neuron reconstruction performance by combining any one of the three neuron segmentation networks. Consequently, the segmentation network and traditional tracing method can complement and promote each other, leading to more complete reconstruction.

#### 4.1.5. Enhancement Parameter

In order to explore the influence of parameter  $\alpha$  in Eq. (2) for image enhancement, we adopt different values for  $\alpha$ , and the results are shown in Fig. 13.  $\alpha = 0$  means that the raw image block is directly used as input for the tracing module.  $\alpha = 1$  means that only the probability maps are used as input for neuron tracing. It indicates that the performance is improved by combining the probability map with the raw intensity, mainly because that the probability map reflects the long-range

**Table 1. F-Score of neuron reconstruction on a test image block from the VISO-R-40 dataset at different iterations.**

Method	iter-1	iter-3	iter-5
APP1 Peng et al. (2011)	0.315	0.599	0.599
APP2 Xiao and Peng (2013)	0.251	0.511	0.538
MOST Wu et al. (2014)	0.322	0.588	0.721
NGPST Quan et al. (2015)	0.758	0.825	0.888

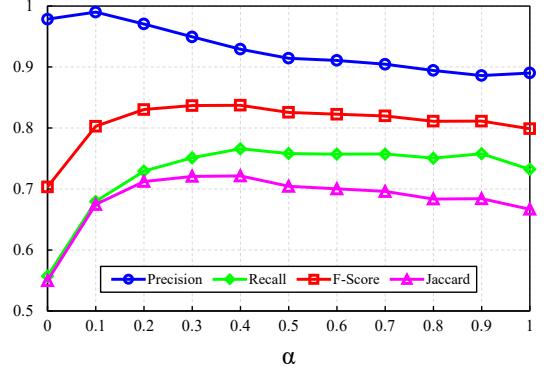


**Fig. 12. F-Score of neuron reconstruction on the VISO-R-40 testing dataset (the whole dataset?) at five iterations. Combining any one of the three neuron segmentation networks, our approach progressively improves the reconstruction performance.**

trajectory structures while the original intensity preserves signal details to some extent. In this paper, we empirically select  $\alpha = 0.1$  to reduce the influence of false positive predictions in probability maps due to the limited performance of the DNN model trained by pseudo labels and increase the robustness of the whole framework.

#### 4.1.6. Comparison on VISO-R-40 Dataset

To prove the effectiveness of our method on neuronal population reconstruction, we compare it with seven widely used neuron tracing methods, including APP1 Peng et al. (2011), APP2 Xiao and Peng (2013), MOST Wu et al. (2014), Smart-Tracing Chen et al. (2015), NGPST Quan et al. (2015), TReMAP Zhou et al. (2016) and FMST Yang et al. (2019). The parameters of these tracing methods are manually adjusted for each image to get optimal performance in the experiments. We utilize NGPST with 3D DSN enhancement as “Ours”. Note that the segmentation network in our approach is trained progressively on the VISO-R-40 dataset in the training stage. We use the trained model directly at the testing stage for evaluation. Table 2 compares the quantitative results of different methods with regard to Precision, Recall, F-Score and Jaccard. It shows that our method makes a significant improvement on the overall performance compared with other methods. Though APP2 achieves the highest precision, the reconstructed neurons are significantly sparser than others. (What happens if you provide multiple somas for APP2? ) Fig. 14 shows the neuronal populations reconstructed from three testing image blocks. Compared with other methods, our PLNPR is superior in both sparse and dense neurons. Conventional methods Peng et al. (2011); Xiao and Peng (2013); Wu et al. (2014); Zhou et al. (2016) and learning-based methods Chen et al. (2015); Yang et al. (2019)



**Fig. 13. Neuron reconstruction performance with different  $\alpha$  in Eq. (2) for image enhancement on the VISO-R-40 test dataset. From left to right, the value of  $\alpha$  increases from 0 to 1 by a step of 0.1.**

tend to extract the main trunk of neurons, while missing a large portion of subtle neurites. Therefore, these methods have very high precision but significantly lower recall. Although NGPST Quan et al. (2015) achieves better performance of neuronal identity, it still remains difficult to extract subtle neuron voxels by using hand-crafted features. In comparison, our method benefits from the progressively trained segmentation network, and reconstructs more complete neurons from challenging blocks, even there exhibit noises, low contrast and blending of fluorescence in the blocks.

#### 4.2. Evaluation of PLNPR on BigNeuron Dataset

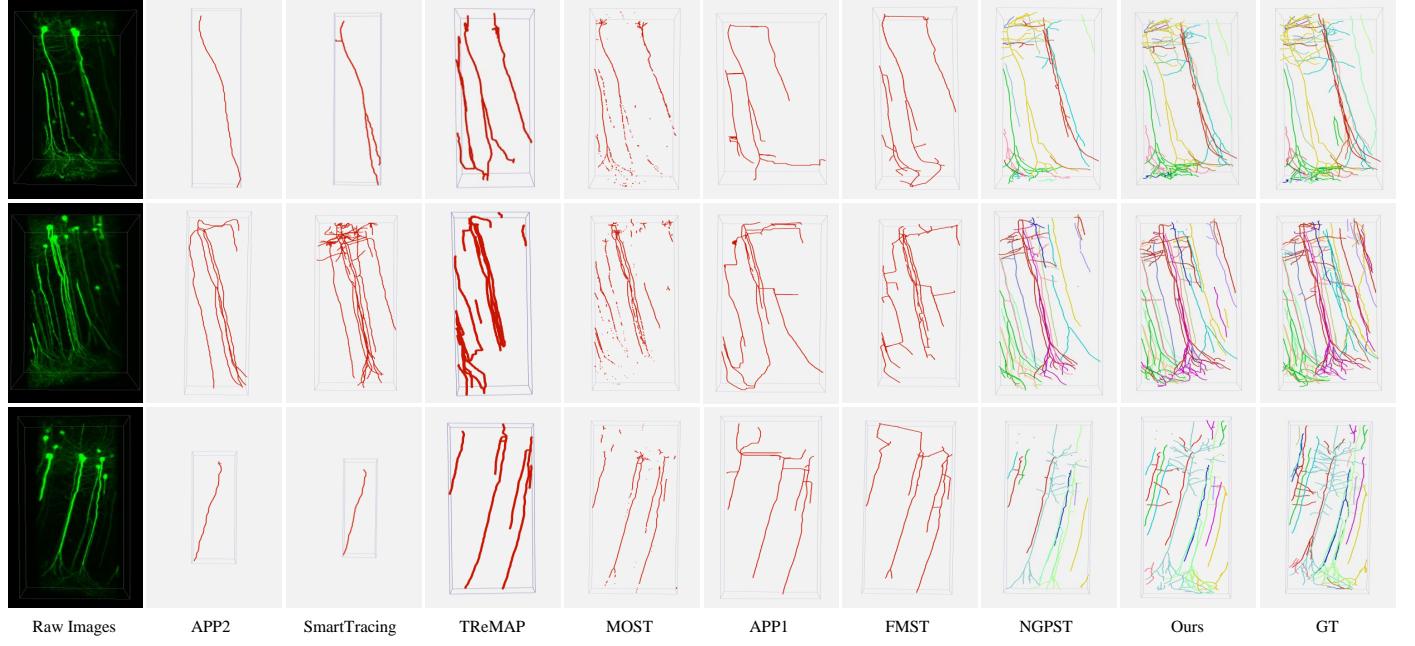
##### 4.2.1. BigNeuron Dataset

To validate our PLNPR method on single neuron reconstruction, we employ the BigNeuron Peng et al. (2015) dataset which is a well-known community-derived neuron dataset. This dataset totally consists of about 20,000 3D OM images, acquired from a variety of species and optical imaging systems. Unlike our VISO-R-40 dataset which is built for the evaluation of neuronal population reconstruction, each block in the BigNeuron dataset contains a single neuron or fragmented neurites. Following Li et al. (2017), we select the same 68 images that are from a variety of species. Manual reconstruction by experts is associated with each image. 51 images are used for network training and the remaining 17 images are used for evaluation.

##### 4.2.2. Experimental Settings and Evaluation Metrics

To evaluate our PLNPR on the single neuron reconstruction, we use the DSN model pretrained on the VISO-R-40 dataset as initialization and fine-tune it on the BigNeuron dataset using pseudo labels generated by NGPST Quan et al. (2015) instead of the provided manual annotations. The learning rate was initialized as  $1 \times 10^{-4}$  and decayed using the “poly” learning rate policy (power of 0.9). The maximum iteration number is set to 24000. We cropped patches of size  $160 \times 160 \times 8$  as input to the network, considering consumption of the GPU memory, and also unequal image sizes along three directions. (Do you use  $160 \times 160 \times 160$  on the VISO-R-40 dataset?)

Since the implementation of most learning-based tracing methods, such as Li et al. (2017) are not publicly available, to



**Fig. 14.** Comparison of neuronal population reconstruction results of three image blocks. Our PLNPR method reconstructs more complete and accurate neurons compared to other methods.

**Table 2. Performance comparison with different methods for neuronal population reconstruction on the VISO-R-40 dataset.**

Method	Precision	Recall	F-Score	Jaccard
APP2 Xiao and Peng (2013)	<b>0.980</b>	0.091	0.157	0.091
SmartTracing Chen et al. (2015)	0.961	0.133	0.205	0.128
TReMAP Zhou et al. (2016)	0.917	0.147	0.253	0.145
MOST Wu et al. (2014)	0.969	0.151	0.258	0.151
APP1 Peng et al. (2011)	0.935	0.169	0.284	0.167
FMST Yang et al. (2019)	0.884	0.179	0.296	0.176
NGPST Quan et al. (2015)	0.978	0.557	0.703	0.549
Ours	0.971	<b>0.801</b>	<b>0.875</b>	<b>0.781</b>

compare with Li et al. (2017), the testing data and three evaluation metrics for evaluation are the same as those used in Li et al. (2017). The three measurements defined in Peng et al. (2010) include entire structure average (ESA), different structure average (DSA) and percentage of different structures (PDS). For all of these three scores, larger values indicate higher discrepancy between the tracing results and the manual reconstruction.

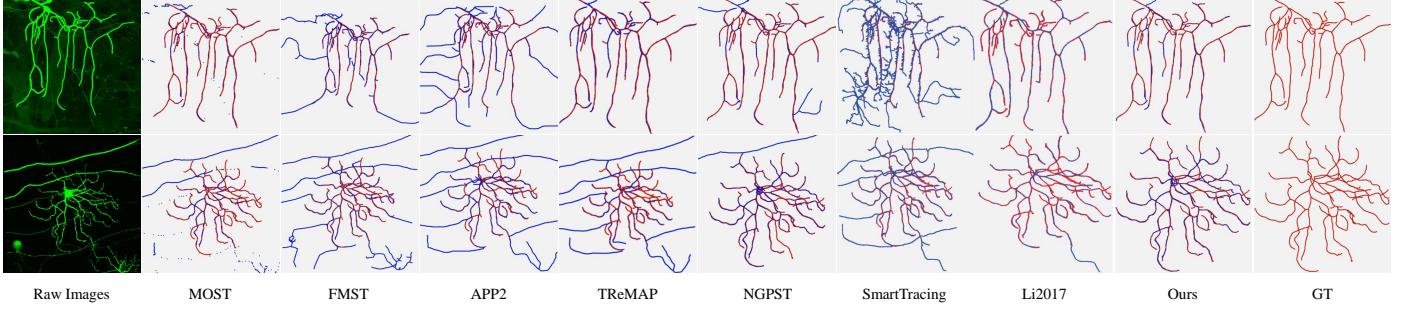
#### 4.2.3. Comparison on BigNeuron Dataset

On the BigNeuron dataset, we compare with seven widely used tracing methods to validate the effectiveness of our proposed method. They are APP2 Xiao and Peng (2013), MOST Wu et al. (2014), SmartTracing Chen et al. (2015), NGPST Quan et al. (2015), Li2017 Li et al. (2017), TReMAP Zhou et al. (2016) and FMST Yang et al. (2019) respectively. Fig. 15 visualizes the neurons reconstructed from two testing blocks. Our method reconstructs more accurate neuronal morphology compared with other methods. In Table 3, we list the weighted average of the DSA, PDS and ESA on all testing data using different methods. The weight of each testing block is proportional

to the neuron length identified in the corresponding manual annotation. From Table 3, we can observe that our method outperforms others in both PDS and ESA metrics and also achieves comparable performance in DSA metric with the best one. In particular, unlike Li et al. (2017) requiring on a strongly supervised network, our method obtains even better performance without any manual annotations. With ever-increasing number of unlabeled neuron datasets are collected, our method can easily utilize them to further improve the performance of neuron reconstruction.

#### 4.3. Evaluation of UltraNPR on a Mouse Brain Slice

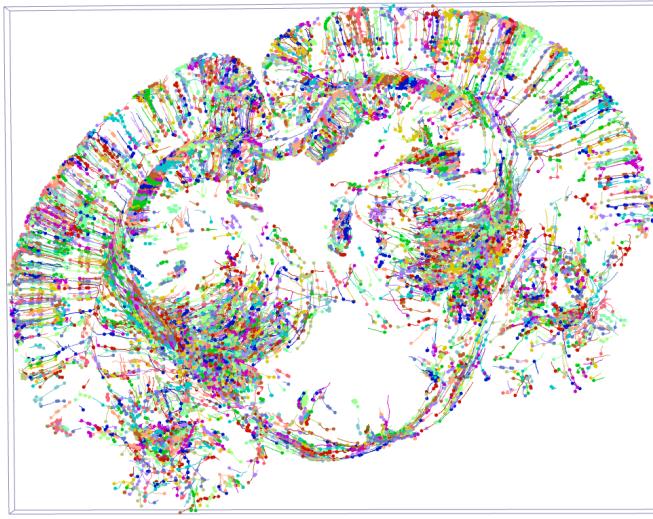
To reconstruct dense neuron populations in an ultra-scale image, we divide the entire image into blocks in size of  $1120 \times 2048 \times 869$  considering the memory and computational efficiency of our PLNPR. The overlap between adjacent blocks is set to be 300 voxels along each dimension. The  $\delta_\alpha$  is set to be 70 voxels and the  $\beta$  is set to be 10 voxels. Our UltraNPR took just over one day (**in hours**) to reconstruct the image on a cluster computer with 64 GB of working memory and 20 NVIDIA



**Fig. 15.** Comparison of single neuron reconstruction results on two test images from the BigNeuron dataset. The reconstruction neurites are shown in blue and the corresponding ground truth (GT) are shown in red. Our method reconstructs more complete and accurate neurons compared to other methods.

**Table 3. Performance comparison for single neuron reconstruction on the BigNeuron test dataset.**

Method	Precision	Recall	F-Score	Jaccard	ESA	DSA	PDS
MOST Wu et al. (2014)	0.619	0.361	0.456	0.295	31.730	38.211	0.633
FMST Yang et al. (2019)	0.575	0.629	0.601	0.429	17.878	23.459	0.558
APP2 Xiao and Peng (2013)	<b>0.799</b>	0.492	0.608	0.437	13.457	17.923	0.562
TReMAP Zhou et al. (2016)	0.771	0.415	0.539	0.369	11.269	17.941	0.539
NGPST Quan et al. (2015)	0.710	0.680	0.695	0.532	10.168	14.880	0.587
SmartTracing Chen et al. (2015)	0.701	0.648	0.674	0.508	8.532	11.609	0.543
Li2017 Li et al. (2017)	-	-	-	-	4.917	<b>7.972</b>	0.461
Ours	0.790	<b>0.707</b>	<b>0.746</b>	<b>0.595</b>	<b>4.784</b>	8.309	<b>0.451</b>



**Fig. 16.** The reconstruction result of neuronal populations in a large-scale 3D mouse brain slice using our UltraNPR method.

1080Ti GPUs. As shown in Fig. 16, a neuronal population which consists of 5348 neurons is successfully reconstructed from the large-scale image.

Several neurons selected from the reconstructed neuronal population are visualized in Fig. 17. These reconstructions provide detailed neuronal structures and enable further neuronal morphology analysis. In summary, our UltraNPR is capable of reconstructing dense neuronal populations from noisy and large-scale OM brain images.

Neuron length, number of branches.



**Fig. 17.** Single neurons selected from the reconstructed neuronal populations in a mouse brain slice using our UltraNPR method.

Since the manual annotation of neuronal populations from noisy OM images is difficult to obtain, let along the annotation of large-scale neuronal populations from a mouse brain slice. In this section, some qualitative results are visualized for verifying the effectiveness of our UltraNPR method.

## 5. Conclusion

In this work, we propose PLNPR, a progressive learning framework for neuronal population reconstruction from noisy and low-quality OM image blocks. Without using any manual annotations, our PLNPR method takes advantage of neuron tracing techniques and deep segmentation networks, and makes them mutually complement and promote each other progressively. We extensively validate the proposed PLNPR for neuron reconstruction and the results demonstrate the effectiveness and superiority of our method. Integrating PLNPR with an efficient block-wise tracing and fusion strategy, our UltraNPR successfully reconstruct dense neuronal populations in a ultra-scale OM images of a mouse brain slice. We construct a new dataset

“ViSoR-40” which consists of 40 OM image blocks for evaluation of neuronal population reconstruction. This dataset is now available and we believe that it will facilitate further brain studies, including neuron counting, neuron reconstruction, neuron morphology analysis, and so on.

## Acknowledgements

This work was supported by the Natural Science Foundation of China [grant number 91732304]; and the Fundamental Research Funds for the Central Universities [grant numbers WK2380000002, WK3490000003].

## References

- Bas, E., Erdogmus, D., 2011. Principal curves as skeletons of tubular objects. *Neuroinformatics* 9, 181–191.
- Basu, S., Ooi, W.T., Racoceanu, D., 2016. Neurite tracing with object process. *IEEE Trans. Med. Imag.* 35, 1443–1451.
- Cai, H., Xu, X., Lu, J., Lichtman, J.W., Yung, S.P., Wong, S.T.C., 2006. Repulsive force based snake model to segment and track neuronal axons in 3D microscopy image stacks. *NeuroImage* 32, 1608–1620.
- Chen, H., Xiao, H., Liu, T., Peng, H., 2015. Smarttracing: self-learning-based neuron reconstruction. *Brain Informatics* 2, 135. doi:10.1007/s40708-015-0018-y.
- Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O., 2016. 3D U-Net: learning dense volumetric segmentation from sparse annotation, in: Proc. MICCAI, pp. 424–432.
- De, J., Cheng, L., Zhang, X., Lin, F., Li, H., Ong, K.H., et al., 2016. A graph-theoretical approach for tracing filamentary structures in neuronal and retinal images. *IEEE Trans. Med. Imag.* 35, 257–272. doi:10.1109/TMI.2015.2465962.
- Dou, Q., Yu, L., Chen, H., Jin, Y., Yang, X., Qin, J., et al., 2017. 3D deeply supervised network for automated segmentation of volumetric medical images. *Med. Imag. Anal.* 41, 40–54.
- Frasconi, P., Silvestri, L., Soda, P., Cortini, R., Pavone, F.S., Iannello, G., 2014. Large-scale automated identification of mouse brain cells in confocal light sheet microscopy images. *Bioinformatics* 30, 587–593. doi:10.1093/bioinformatics/btu469.
- Giorgio, A., De Stefano, N., 2013. Clinical use of brain volumetry. *J. Magn. Reson. Imaging* 37, 1–14.
- Gu, L., Zhang, X., Zhao, H., Li, H., Cheng, L., 2017. Segment 2D and 3D Filaments by Learning Structured and Contextual Features. *IEEE Trans. Med. Imag.* 36, 596–606.
- Kozinski, M., Mosinska, A., Salzmann, M., Fua, P., 2020. Tracing in 2D to reduce the annotation effort for 3D deep delineation. *Medical Image Analysis* 60, 1–11. doi:10.1016/j.media.2019.101590.
- Li, R., Zeng, T., Peng, H., Ji, S., 2017. Deep learning segmentation of optical microscopy images improves 3-D neuron reconstruction. *IEEE Trans. Med. Imag.* 36, 1533–1541. doi:10.1109/TMI.2017.2679713.
- Liu, M., Chen, W., Wang, C., Peng, H., 2019. A Multiscale Ray-Shooting Model for Termination Detection of Tree-Like Structures in Biomedical Images. *IEEE Trans. Med. Imag.* 38, 1923–1934.
- Liu, S., Zhang, D., Liu, S., Feng, D., Peng, H., Cai, W., 2016. Rivulet: 3D neuron morphology tracing with iterative back-tracking. *Neuroinformatics* 14, 387. doi:10.1007/s12021-016-9302-0.
- Liu, S., Zhang, D., Song, Y., Peng, H., Cai, W., 2018. Automated 3-D Neuron Tracing With Precise Branch Erasing and Confidence Controlled Back Tracking. *IEEE Trans. Med. Imag.* 37, 2441–2452.
- Navlakha, S., Ahammad, P., Myers, E.W., 2013. Unsupervised segmentation of noisy electron microscopy images using salient watersheds and region merging. *BMC Bioinform.* 14, 294. doi:10.1186/1471-2105-14-294.
- Peng, H., Hawrylycz, M., Roskams, J., Hill, S., Spruston, N., Meijering, E., et al., 2015. BigNeuron: large-scale 3d neuron reconstruction from optical microscopy images. *Neuron* 87, 252–256.
- Peng, H., Long, F., Myers, G., 2011. Automatic 3D neuron tracing using all-path pruning. *Bioinformatics* 27, 239–247. doi:10.1093/bioinformatics/btr237.
- Peng, H., Ruan, Z., Atasoy, D., Sternson, S., 2010. Automatic reconstruction of 3d neuron structures using a graph-augmented deformable model. *Bioinformatics* 26, 38–46.
- Peng, H., Tang, J., Xiao, H., Bria, A., Zhou, J., Butler, V., et al., 2014. Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis. *Nature Communications* 5, 4342.
- Peng, H., Zhou, Z., Meijering, E., Zhao, T., Ascoli, G.A., Hawrylycz, M., 2017. Automatic tracing of ultra-volumes of neuronal images. *Nature Methods* 14, 332.
- Petrella, J.R., Coleman, R.E., Doraiswamy, P.M., 2003. Neuroimaging and early diagnosis of alzheimer disease: A look to the future. *Radiology* 226, 315–336.
- Quan, T., Zheng, T., Yang, Z., Ding, W., Li, S., Li, J., et al., 2013. NeuroGPS: automated localization of neurons for brain circuits using L1 minimization model. *Scientific Reports* 3, 1414.
- Quan, T., Zhou, H., Li, J., Li, S., Li, A., Li, Y., et al., 2015. NeuroGPS-Tree: automatic reconstruction of large-scale neuronal populations with dense neurites. *Nature Methods* 13, 51–54.
- Radojević, M., Meijering, E., 2017. Automated neuron tracing using probability hypothesis density filtering. *Bioinformatics* 33, 1073–1080. doi:10.1093/bioinformatics/btw751.
- Santamaría-Pang, A., Hernandez-Herrera, P., Papadakis, M., Saggau, P., Kakadiaris, I.A., 2015. Automatic morphological reconstruction of neurons from multiphoton and confocal microscopy images using 3d tubular models. *Neuroinformatics* 13, 297–320.
- Senft, S.L., 2011. A brief history of neuronal reconstruction. *Neuroinformatics* 9, 119–128.
- Skibbe, H., Reisert, M., Maeda, S., Koyama, M., Oba, S., Ito, K., Ishii, S., 2015. Efficient monte carlo image analysis for the location of vascular entity. *IEEE Trans. Med. Imag.* 34, 628–643.
- Skibbe, H., Reisert, M., Nakae, K., Watakabe, A., Hata, J., Mizukami, H., et al., 2019. Pat-probabilistic axon tracking for densely labeled neurons in large 3-d micrographs. *IEEE Trans. Med. Imag.* 38, 69–78.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958.
- Sümbül, U., Roossien, D., Cai, D., Chen, F., Barry, N., Cunningham, J.P., et al., 2016. Automated scalable segmentation of neurons from multispectral images, in: Proc. Adv. Neural Inf. Process. Syst., pp. 1912–1920.
- Sun, K., Xiao, B., Liu, D., Wang, J., 2019. Deep high-resolution representation learning for human pose estimation, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 5693–5703.
- Wang, C.W., Lee, Y.C., Pradana, H., Zhou, Z., Peng, H., 2017. Ensemble neuron tracer for 3d neuron reconstruction. *Neuroinformatics* 15, 185–198.
- Wang, H., Zhang, D., Song, Y., Liu, S., Gao, R., Peng, H., Cai, W., 2018. Memory and Time Efficient 3D Neuron Morphology Tracing in Large-Scale Images, in: Proc. DICTA, pp. 1–8.
- Wang, H., Zhu, Q., Ding, L., Shen, Y., Yang, C.Y., Xu, F., et al., 2019. Scalable volumetric imaging for ultrahigh-speed brain mapping at synaptic resolution. National Science Review URL: <https://doi.org/10.1093/nsr/nwz053>. doi:10.1093/nsr/nwz053.
- Wang, Y., Narayanaswamy, A., Tsai, C.L., Roysam, B., 2011. A broadly applicable 3-D neuron tracing method based on open-curve snake. *Neuroinformatics* 9, 193–217.
- Wu, J., He, Y., Yang, Z., Guo, C., Luo, Q., Zhou, W., et al., 2014. 3D BrainCV: Simultaneous visualization and analysis of cells and capillaries in a whole mouse brain with one-micron voxel resolution. *NeuroImage* 87, 199–208.
- Xiao, H., Peng, H., 2013. APP2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics* 29, 1448–1454. doi:10.1093/bioinformatics/btt170.
- Xu, K., Su, H., Zhu, J., Guan, J., Zhang, B., 2016. Neuron segmentation based on CNN with semi-supervised regularization, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, pp. 1324–1332. doi:10.1109/CVPRW.2016.167.
- Yang, J., Gonzalez-Bellido, P.T., Peng, H., 2013. A distance-field based automatic neuron tracing method. *BMC Bioinform.* 14, 93. doi:10.1186/1471-2105-14-93.
- Yang, J., Hao, M., Liu, X., Wan, Z., Zhong, N., Peng, H., 2019. FMST: an automatic neuron tracing method based on fast marching and minimum spanning tree. *Neuroinformatics* 17, 185–196. doi:10.1007/s12021-018-9392-y.
- Yu, L., Cheng, J.Z., Dou, Q., Yang, X., Chen, H., Qin, J., et al., 2017. Automatic

- 3D cardiovascular mr segmentation with densely-connected volumetric convnets, in: Proc. MICCAI, pp. 287–295.
- Zhao, J., Chen, X., Xiong, Z., Liu, D., Zeng, J., Zhang, Y., et al., 2019. Progressive learning for neuronal population reconstruction from optical microscopy images, in: Proc. MICCAI, pp. 750–759.
- Zhao, T., Xie, J., Amat, F., Clack, N., Ahammad, P., Peng, H., et al., 2011. Automated reconstruction of neuronal morphology based on local geometrical and global structural models. Neuroinformatics 9, 247–261.
- Zhou, Z., Kuo, H.C., Peng, H., Long, F., 2018. Deepneuron: an open deep learning toolbox for neuron tracing. Brain Informatics 5, 3.
- Zhou, Z., Liu, X., Long, B., Peng, H., 2016. TReMAP: Automatic 3D neuron reconstruction based on tracing, reverse mapping and assembling of 2d projections. Neuroinformatics 14, 41. doi:10.1007/s12021-015-9278-1.
- Zhou, Z., Sorensen, S.A., Peng, H., 2015. Neuron Crawler: An automatic tracing algorithm for very large neuron images, in: Proc. IEEE Int. Symp. Biomed. Imag., pp. 870–874. doi:10.1109/ISBI.2015.7164009.