

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

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

**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 from ultra-scale images. To solve the problem of high cost in obtaining manual annotations 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 from a terabyte-sized ultra-scale image, we introduce an automatic framework which adaptively traces neurons block by block and fuses fragmented neurites 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 from an ultra-scale mouse brain slice. The proposed adaptive block propagation and fusion strategies greatly improve the completeness of neurites in dense neuronal population reconstruction.

**Index Terms**—Neuronal population reconstruction, ultra-scale images, optical microscopy, progressive learning.

## I. 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 ultra-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 [1], [2]. One of the key techniques in this endeavor is optical microscopy (OM), which allows detailed visualization of neurons and makes the reconstruction

J. Zhao, X. Chen, Z. Xiong, D. Liu, J. Zeng, C. Xie, Y. Zhang, Z. Zha, G. Bi and F. Wu are with the National Engineering Laboratory for Brain-inspired Intelligence Technology and Application, University of Science and Technology of China, Hefei 230027, China (email: {jzhaoch, artzjj}@mail.ustc.edu.cn; {xjchen99, zxwiong, dongeliu, zhyuey, zhazj, gqbi, fengwu}@ustc.edu.cn).

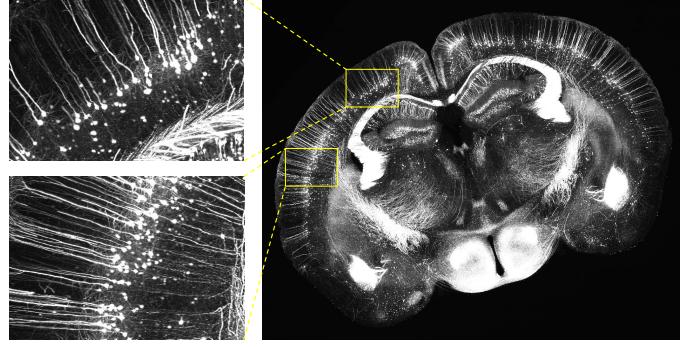


Fig. 1. A 3D OM image for a mouse brain slice captured by the VISoR imaging system [4]. The large density of neurons, low contrast, image noises, and huge volume pose significant challenges for automatic reconstruction of neuronal populations from the ultra-scale image.

of every neuron possible [3], 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 huge 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 algorithm for reconstructing large-scale neuronal populations in these challenging situations is greatly desired in practice.

Attempts have been made for neuron reconstruction from large-scale OM images in recent years, such as Neuron Crawler [5], UltraTracer [6] and MEIT [7]. 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,

ray-shooting, 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 [8]–[11]. However, these approaches require huge amount of manually annotated data to train the deep neuron segmentation networks.

Compared with manual annotations, 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 DNN for segmenting neurons, the neuron reconstruction using conventional tracers could be improved. Then we progressively refine the segmentation DNN 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 [12] on a few image blocks. In this work, we apply PLNPR on dense neuron population reconstruction from an ultra-scale OM image.

Another limitation of existing large-scale neuron reconstruction methods [5]–[7] is that they mainly focus on single neuron reconstruction. For dense neuronal population reconstruction from ultra-scale 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 novel tracing and fusion strategies for reconstructing 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 neuronal population. There are 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 that usually occur at boundary regions of local blocks.

In summary, we propose PLNPR, a progressive learning framework that integrates traditional tracing methods and deep segmentation networks for neuron population reconstruction without using manual annotations. Then we introduce UltraNPR by integrating PLNPR with adaptive block-wise propagation and fusion strategies 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 “VISO-R-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 VISO-R-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. The reconstructed neuron population from an ultra-scale image of a mouse brain slice shows the robustness of our UltraNPR.

## II. RELATED WORK

Early techniques for neuron reconstruction from optical microscopy images typically employ traditional image processing algorithms, such as snakes [13], [14], principal curves [15], graph theory [16]–[18], model-fitting [19], [20], watershed [21], [22], energy minimization [23], [24], mean-shift clustering [25], fast-marching [26]–[28], ray-shooting [29], [30], and so on. Unfortunately, these conventional algorithms rely on hand-crafted features and carefully tuned parameters, thus usually tending 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) [31], minimum spanning tree [32], Bayesian probabilistic model [33], Bootstrap aggregating [34], gradient boosting decision trees (GBDT) [35], Markov chain Monte Carlo (MCMC) [36], [37], and so on. However, these methods employ hand-crafted features that usually suffer from limited representation capability for accurate recognition, considering more challenging images and complex neuron morphologies.

Recently, deep learning-based methods [8]–[11] 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 powerful feature representation and classifiers for extracting neurons 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 DNN-based neuron reconstruction performance without using manual annotations.

Despite substantial advancements brought by existing methods, they often need to load all the voxels into memory before neuron tracing. The sheer volume of an ultra-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 [5], UltraTracer [6], and MEIT [7]. To tackle the challenge of huge image volume, a common solution is to reconstruct a neuron block by block. In each local block cropped from a large-scale image, existing tracing methods, such as APP2 [27], MOST [29], neuTube [38], and FMST [39], 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

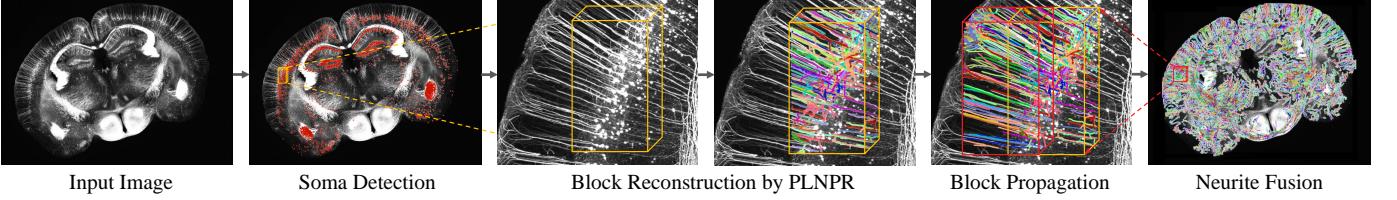


Fig. 2. Diagram of our UltraNPR algorithm for neuronal population reconstruction from an ultra-scale image of a brain slice.

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 populations from ultra-scale OM images, where closely spaced neurites that belong to different neurons are difficult to distinguish for existing methods that are designed for single neuron reconstruction.

### III. PROPOSED METHOD

Given an ultra-scale noisy OM image, our UltraNPR follows a block-by-block neuron reconstruction scheme, as Fig. 2 shows. The ultra-scale input image may contain billions or even trillions of voxels. We first divide the input image into blocks that are averagely in size of  $0.5\text{mm} \times 1.0\text{mm} \times 0.4\text{mm}$  with overlaps. For each block, we first enhance the noisy images by a deep neural network, which is progressively trained from reconstructed neurons using our PLNPR. Then we reconstruct neuronal populations in each block using the enhanced image. Since it is more reliable to start from somas rather than subtle neurites for neuron reconstruction, we first detect somas and start tracing from the blocks where somas can be detected. Our UltraNPR then employs an effective block propagation strategy to trace dense neuronal populations in an adaptive order by finding neighboring blocks with more neurite tips. Finally, we fuse overlapped neurites in adjacent blocks to reconstruct complete neuronal populations from the whole ultra-scale image.

#### A. PLNPR for Robust Neuron Reconstruction

To reconstruct neurons from 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 that 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 [40] 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  $\{\mathbf{M}_i\}_{i=1}^N$  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 image signals, the image block is enhanced in order to preserve both local signals and global structures simultaneously. When the enhanced block is fed into the neuron tracing module, a more complete neuronal population can be reconstructed and provide better pseudo labels for the next iteration of network training. Based on the iterative learning process, the DNN and neuron tracer mutually complement and promote each other to gradually improve the neuron reconstruction performance.

1) *DNN for 3D Neuron Segmentation*: Extracting neuron voxels from image blocks is non-trivial since the size, morphology, and intensity of neurons vary significantly. In recent years, many 3D DNNs, such as 3D U-Net [41], 3D DSN [42], and DenseVoxNet [43], have demonstrated their outstanding capability in various biological and biomedical image segmentation tasks. We extend the 3D DSN 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 [42], it is prone to overfitting in our case due to the limited training data. We employ the dropout [44] 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.

Another challenge in 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 leads to inhomogeneous image quality along different dimensions. Therefore, a random transposition process is employed for each cube as data augmentation for network training.

In addition, the volume of neuron voxels is usually much smaller than that of background in an OM image. A data balancing technique is employed for network training. When computing the training loss, we only consider the neuron voxels and a certain portion of background, which is randomly selected as non-neuron samples. The number of non-neuron

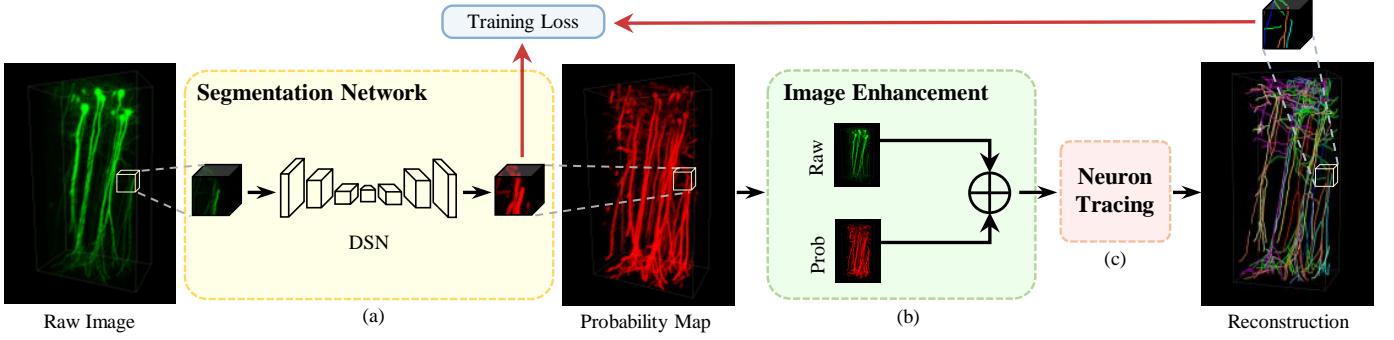


Fig. 3. Diagram of our progressive learning algorithm for neuron reconstruction from an image block. (a) The segmentation network extracts neuron signals from the raw image. (b) The output probability map is employed to enhance the raw image in order to preserve both global structures and local signal 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 model and neuron reconstruction with a set of images. The black arrows show the reconstruction pass from an image block at the test stage.

voxels used for training is set as 10 times that of neuron voxels.

2) *Image Enhancement*: After training the segmentation network using pseudo labels, the trained model is used to predict a probability map for 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 of the corresponding voxel in  $\mathbf{B}$  being a neuron voxel. 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  $\tilde{\mathbf{P}}$  is first constructed by linearly mapping the value range of  $\mathbf{P} \in [0, 1]$  to the value range  $[b_{min}, b_{max}]$  of  $\mathbf{B}$  as

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

Then, an enhanced block  $\mathbf{E}$  is computed as

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

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 block to the neuron tracing module, neuronal populations can be reconstructed more completely.

With more reliable reconstruction results as pseudo labels, the segmentation network could be further trained to learn more discriminative and representative features for producing probability maps, which in turn benefit the tracing module to reconstruct more complete neurons in the next iteration. As shown in Fig. 4 (a), a raw image block typically contains many noises and inhomogeneous intensities. At first, by feeding the raw block to the tracer [40], the reconstructed neuronal population is incomplete and many neurites are missing, as Fig. 4(g) shows, comparing with the ground truth (GT) shown in Fig. 4(k). Then, by utilizing the pseudo labels derived from imperfect reconstruction, the segmentation network can be trained to learn features for global morphology. Fig. 4(b)

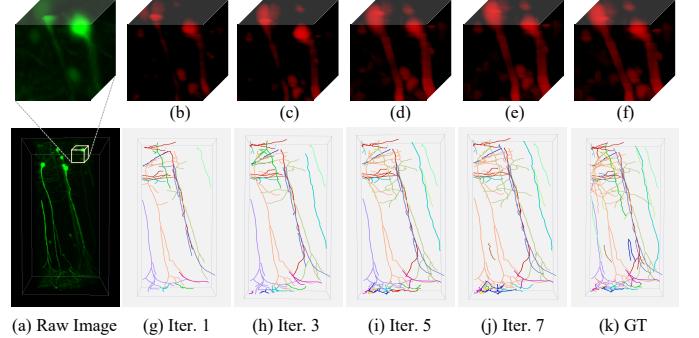


Fig. 4. Our progressive learning technique gradually improves the segmentation network to extract neuron signals from a raw image block (a). (b-f) The probability maps generated by the segmentation network at different iterations. (g-j) More and more complete and accurate reconstruction of the neuronal populations can be obtained with more iterations. (k) Manually labeled neurons. Separated neurons are shown in different colors.

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).

### B. Ultra-scale Neuronal Population Reconstruction

Our PLNPR enhances the image signals and traces neurons from each image block. However, neurons usually exist across multiple blocks in an ultra-scale image. We propose an effective stitching process by looking for the most continuous block to trace dense neuron populations from all unreconstructed blocks. As somas are where signals from the dendrites are joined and pass on, 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 of their neighboring

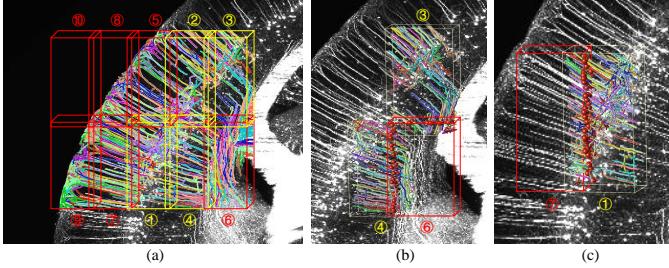


Fig. 5. An example of the iterative block reconstruction for ten adjacent blocks. (a) The numbers indicate the order in which the ten blocks are traced. The blocks 1, 2, 3, 4 are firstly reconstructed using our PLNPR since somas are detected in these four blocks. (b) Block-6 is then reconstructed by finding pseudo somas from the reconstructed neurites of its two neighboring blocks (3, 4) for tracing. (c) Block-7 is reconstructed by finding pseudo somas from the reconstructed neurites in block-1.

blocks. Finally, the overlapped neurites reconstructed from adjacent blocks are fused to get a complete reconstruction of neurons. Fig. 2 shows the pipeline of our UltraNPR approach.

1) *Block Reconstruction and Propagation:* We apply the soma detection algorithm [23] on each block in an ultra-scale image separately. If a block  $B_i$  contains somas, we apply our PLNPR to reconstruct neurites and get a set of neurites  $\mathcal{N}_i$  in this block, as shown by the blocks in yellow frames in Fig. 5 (a). Each neurite consists of a series of neuron points with their radius, type, and connectivity information [45]. For the remaining unreconstructed blocks, though NGPST can perform neuron tracing without any somas, it typically fails to distinguish multiple neurons with dense neurites, as shown in Fig. 6 (a). In order to avoid under-segmentation of neurons in an unreconstructed block  $B_u$ , we follow the neuron structure in its neighboring blocks that have been reconstructed. More specifically, for each reconstructed block  $B_a$  that is adjacent to  $B_u$ , we collect the intersection points of the reconstructed neurites in  $B_a$  with the boundary plane of  $B_u$  as the pseudo-somas for tracing neurons in  $B_u$ , as shown in Fig. 5 (b)(c). After that, we get the neurites set  $\mathcal{N}_u$  in  $B_u$ . Therefore, for each iteration of our block propagation, the unreconstructed block which has the largest number of neighboring reconstructed blocks and the largest number of pseudo somas is selected for reconstruction next. The reconstructed neurites in this block can then be used to find pseudo somas for its neighboring block that has not been reconstructed. Fig. 5 shows the block propagation and reconstruction process of ten neighboring blocks. This process continues iteratively until all the blocks in the ultra-scale image have been reconstructed.

2) *Neurite Fusion in Adjacent Blocks:* Since a neuron might be split into fragmented neurites when the ultra-scale image is divided into blocks, we now merge and fuse the neurites reconstructed separately from adjacent blocks into individual and complete neurons. Given the reconstructed neurite sets  $\mathcal{N}_a$  and  $\mathcal{N}_b$  in two overlapping blocks  $B_a$  and  $B_b$  respectively, for any pair of neurites  $N_a$  and  $N_b$  in the two sets, if their overlapping volume is larger than a threshold  $\delta_{ovlp}$ , we fuse them together to get a more complete neurite.

There might be over-tracing and topological discrepancy between two neurites that are reconstructed in two blocks

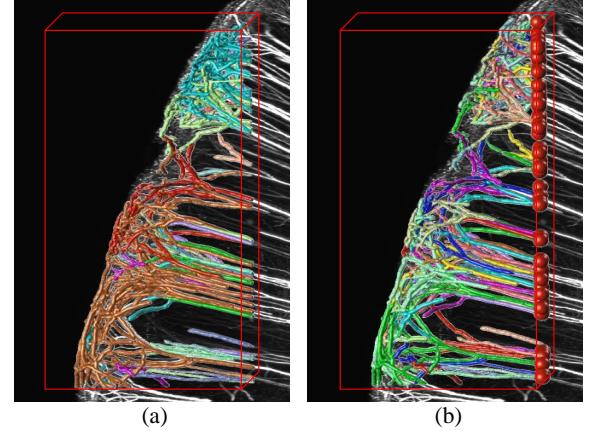


Fig. 6. Comparison of neuron reconstruction results from an enhanced image block by NGPST without setting somas (a) and setting neurite tips (red points) from a neighboring block as pseudo somas (b). Separated neurons are shown in different colors.

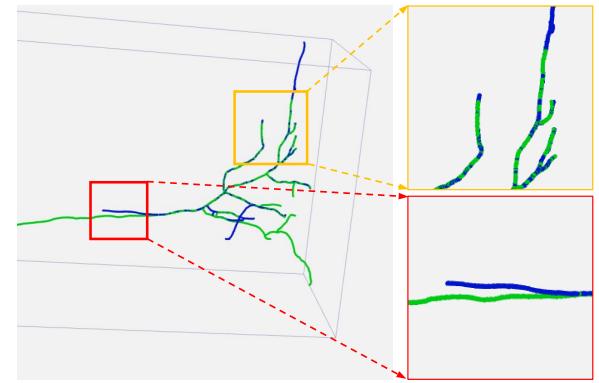


Fig. 7. An example of over-tracing (yellow) and topological discrepancy (red) when assembling two overlapped neurites. These reconstructions are shown in skeleton mode for better visualization.

separately, as shown in Fig. 7. Generally, the traced neurites that are close to the block boundary is less reliable than those 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. We first select the longer one of two overlapped neurites as the reference neurite  $N_a$  and fuse the other neurite  $N_b$  to the reference neurite, as shown in Fig. 8 (b). Each neurite is decomposed to a set of neurite branches, as Fig. 8 (c) shows. Each branch is sampled uniformly into a set of neuron compartments, as Fig. 8 (d) shows. For each branch  $B_b$  of neurite  $N_b$ , we search the branch  $B_a$  of  $N_a$  which has the largest overlapped volume with  $B_b$ , as Fig 8 (f) shows.

To fuse the two matched branches, for each point  $p_a$  in the branch  $B_a$ , if  $p_a$  lies in the boundary region of block  $B_a$  ( $\delta_{bound}$  voxels to the block boundary of  $B_a$ ), it will be removed and all its child branches are also deleted, as shown in Fig 8 (g). The same deletion operation is performed on the points in the branch  $B_b$  that are located in the boundary regions of block  $B_b$ . Then, for each remaining point  $p_b$  in branch  $B_b$ , if there is a point  $p_a$  in branch  $B_a$  so that  $dis(p_b, p_a) < \delta_{pt}$ , it will be removed while the point  $p_a$  is kept, as Fig 8 (g) shows. After that, the remaining part of branch  $B_b$  is merged to the

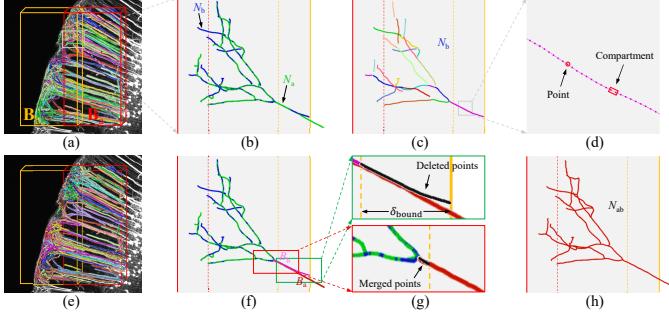


Fig. 8. Neurite fusion in two adjacent blocks. (a) Neurites reconstructed separately in two blocks. For two matched neurites  $N_a$  and  $N_b$  (b), each neurite is decomposed to branches (c) and each branch is sampled uniformly into compartments (d). We delete the neurite points in the boundary regions and merge the points in the middle area of the overlapping region (g) to get a more smooth and complete neurite  $N_{ab}$  (h). (e) Fusion results of two blocks.

reference branch  $B_a$  by connecting it to the nearest point in  $B_a$ . After all matched branches are processed as above, we obtain the fused neurite  $N_{ab}$ . For the remaining branches in  $N_b$  which are not fused to any branches in  $N_a$ , they will be merged to the neurite  $N_{ab}$  by connecting each of them to the nearest point in  $N_{ab}$ . Fig 8 (h) shows the final fusion result of the two neurites. By fusing all neurites in two adjacent blocks, more complete reconstruction of dense neuronal populations can be obtained, as shown in Fig 8 (e).

#### IV. EXPERIMENTS AND RESULTS

We evaluate our neuronal population reconstruction approach on an ultra-scale OM image of a mouse brain slice in dimension of  $25397 \times 18516 \times 869$  (761 GB), as Fig. 1 shows. This image was captured by the VISO-R imaging system [4] at a physical resolution of  $0.5 \mu\text{m} \times 0.5 \mu\text{m} \times 0.5 \mu\text{m}$  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 from local blocks, we first conduct extensive experiments on the VISO-R-40 dataset which we build for evaluation and the BigNeuron dataset [46]. Then we test our UltraNPR algorithm for neuronal population reconstruction from the ultra-scale OM brain image.

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

1) *VISO-R-40 Dataset*: Though many neuron tracing techniques have been proposed, no dataset of OM images has been built for dense neuronal population reconstruction. We construct a neuron image dataset ‘‘VISO-R-40’’ (available at [https://braindata.bitahub.com/Neuronal\\_population\\_reconstruction.html](https://braindata.bitahub.com/Neuronal_population_reconstruction.html)) 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 without manual annotations in our PLNPR. For the remaining 8 blocks, one is used for validation and the other seven blocks are used for testing. Each of them was first independently labeled by two experts. Then, by cross-checking their results, the agreed annotations were approved by another expert to generate the final ground truth.

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 a 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, and momentum of 0.9). The base learning rate is set to 0.001 and descended with the ‘‘poly’’ learning rate policy (power of 0.9 and the maximum iteration number of 24000).

To quantitatively evaluate our method, four commonly used metrics defined in NGPST [40], 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:

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

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

$$F\text{-Score}(R, G) = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (5)$$

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

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 neuron and then averaged in a neuronal population by weighting each neuron with the total length of its neurites, same as [40].

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 [26], APP2 [27], MOST [29], and NGPST [40], are tested as the base tracer in our PLNPR framework. We use their implementations in the software Vaa3D [47]. Eight iterations are tested on our VISO-R-40 dataset. The improvement of neuronal population reconstruction is shown in Fig. 9 (a). We only show the F-Score which is widely used to reflect the overall performance of neuron reconstruction. Moreover, the reconstructed neurons on a test block at different iterations are shown in Fig. 10. 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 neurons. In addition, the performance improvement gets stable after five iterations for all the tested tracing methods.

4) *Neuron Segmentation Network*: To further verify the effectiveness and robustness of our progressive learning strategy, we test three commonly-used deep segmentation networks, including 3D DSN [42], 3D U-Net [41], and a 3D version of HRNet [48], for generating the neuron probability map. Five iterations are tested on our VISO-R-40 dataset, and the F-Score

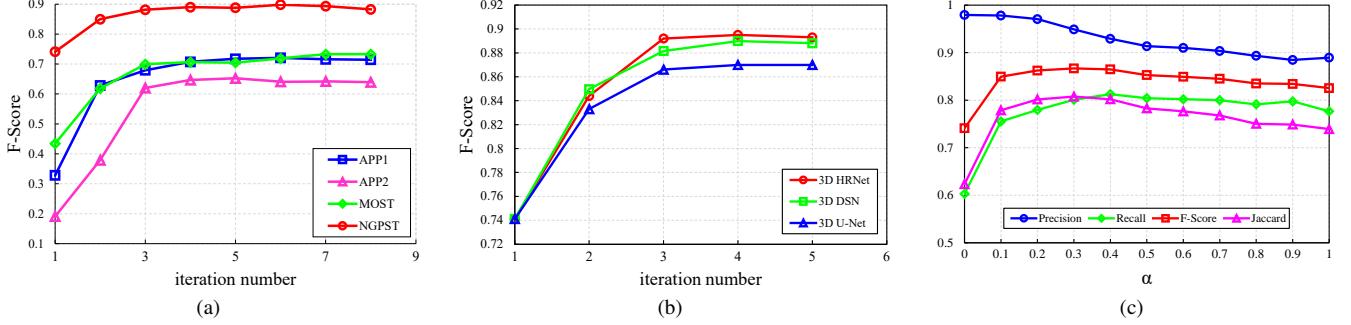


Fig. 9. Comparison of different parameters in our PLNPR framework on the VISoR-40 dataset. (a) Comparisons of F-scores of the neuronal population reconstruction at different iterations with four base tracers respectively. (b) F-Scores of neuron reconstruction at five iterations using three deep segmentation networks. Combining any base tracer and any one of the three neuron segmentation DNNs, our PLNPR scheme progressively improves the reconstruction performance. (c) Neuron reconstruction performance with different  $\alpha$  in Eq. (2) for image enhancement.

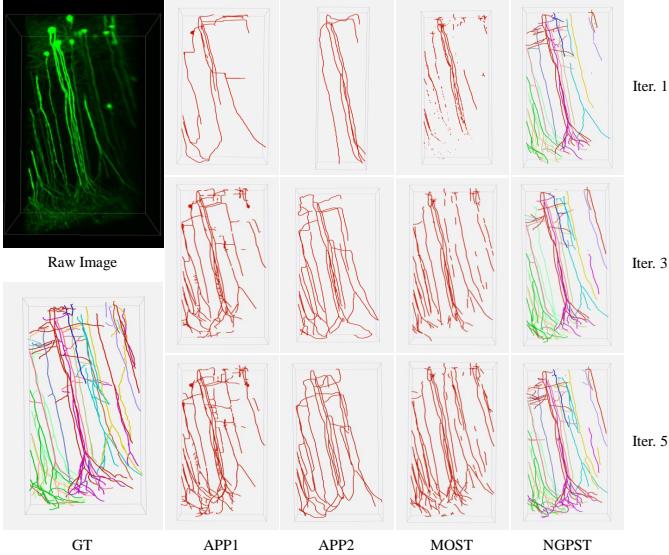


Fig. 10. Neuronal population reconstruction results of a test block at different iterations in our PLNPR framework using four neuron tracing methods.

improvement of reconstruction results is shown in Fig. 9 (b). It can be seen that our PLNPR algorithm effectively improves the neuron reconstruction performance by combining any one of the three neuron segmentation DNNs. Consequently, the segmentation network and traditional tracing method can complement and promote each other, leading to more complete neuron reconstruction.

5) *Enhancement Parameter*: In order to explore the influence of parameter  $\alpha$  in Eq. (2) for image enhancement, we adopt different values for  $\alpha$ . The results are shown in Fig. 9 (c).  $\alpha = 0$  means that the raw blocks are directly used as input for neuron tracing.  $\alpha = 1$  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 image signal, mainly because that the probability map reflects the long-range trajectory structures while the raw image signal carries more details of subtle neurites. 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

TABLE I  
PERFORMANCE COMPARISON WITH DIFFERENT METHODS FOR NEURONAL POPULATION RECONSTRUCTION ON THE VISOR-40 TEST DATASET.

Method	Precision	Recall	F-Score	Jaccard
APP2 [27]	<b>0.980</b>	0.115	0.191	0.119
SmartTracing [31]	0.877	0.291	0.400	0.242
TReMAP [49]	0.917	0.199	0.326	0.2039
MOST [29]	0.969	0.284	0.434	0.316
APP1 [26]	0.935	0.201	0.328	0.205
FMST [39]	0.884	0.208	0.335	0.211
NGPST [40]	0.978	0.603	0.741	0.623
Ours	0.971	<b>0.829</b>	<b>0.893</b>	<b>0.833</b>

labels and increase the robustness of the whole framework.

6) *Comparison with Tracing Methods*: To prove the effectiveness of our PLNPR on neuronal population reconstruction, we compare it with seven widely used neuron tracing methods, including APP1 [26], APP2 [27], MOST [29], SmartTracing (ST) [31], NGPST [40], TReMAP [49], and FMST [39]. The parameters of these tracing methods are manually adjusted for each image block to get the best performance for fair comparison. Fig. 11 shows the neuronal populations reconstructed from three test image blocks. We utilize NGPST as the base tracer with 3D DSN as the segmentation network as “Ours”. The segmentation network is trained progressively on the VISoR-40 dataset in the training stage. We use the trained model directly at the test stage for evaluation. Compared with other methods, our PLNPR is superior in both sparse and dense neurons. Table I compares the quantitative results of different methods with regard to the four metrics including precision, recall, F-Score, and Jaccard. It shows that our PLNPR makes a significant improvement on the overall performance compared with other methods. Though APP2 [27] achieves the highest precision, the reconstructed neurons are significantly sparser than others. Conventional tracing methods [26], [27], [29], [49] and learning-based methods [31], [39] 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 [40] achieves better performance of neuronal population reconstruction compared with other single-neuron tracing methods, it still remains



Fig. 11. Comparison of neuronal population reconstruction results of three image blocks. Our PLNPR method reconstructs more complete and accurate neurons compared to other methods. Separated neurons are shown in different colors.

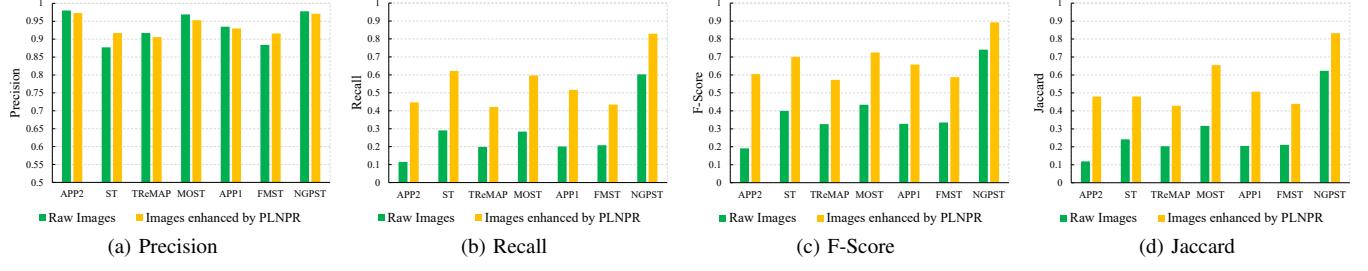


Fig. 12. Performance comparison with seven neuron tracing methods on the raw VISO-R-40 test images and on the images enhanced by our PLNPR.

difficult to extract subtle neurites for NGPST by using hand-crafted features. In comparison, our PLNPR benefits from the progressively trained segmentation network and reconstructs more complete neurons from these challenging blocks, even there exhibit noises, low contrast, and blending of fluorescence in the blocks.

Furthermore, in order to evaluate the effectiveness of our progressive learning framework to enhance images, we compare the reconstruction results of seven neuron tracing methods on the raw test images and on the same enhanced images generated by our PLNPR method in Fig. 12. For any of the seven tracing methods, the overall reconstruction performance, including F-Score and Jaccard, using our enhanced images are much better than the results performed on the raw images under the same parameter settings.

### B. Evaluation of PLNPR on BigNeuron Dataset

1) *BigNeuron Dataset*: To validate our PLNPR method on single neuron reconstruction, we employ the BigNeuron [46] dataset. This dataset consists of about 20,000 3D OM images acquired from a variety of species and optical imaging systems by different institutes. Unlike our VISO-R-40 dataset

which is built to evaluate neuronal population reconstruction, each image block in the BigNeuron dataset only contains a single neuron or fragmented neurites. Following [9], we select the same 68 images that are from a variety of species to evaluate the performance of dense neurite reconstruction. Manual reconstruction by experts is associated with each image. 51 images are used for network training in [9] and the remaining 17 images are used for evaluation. In contrast, we do not use the manual annotations in our PLNPR to train the segmentation deep neural networks.

2) *Experimental Settings*: To evaluate our PLNPR on single neuron reconstruction, we progressively train the DSN model on the BigNeuron dataset using pseudo labels generated by a conventional tracing method, such as NGPST [40] and APP2 [27], instead of the provided manual annotations. The learning rate was initialized as  $1 \times 10^{-4}$  and decayed using the “poly” learning rate policy with power of 0.9. The maximum iteration number is set to 24000. We cropped image patches of size  $160 \times 160 \times 8$  as input to the segmentation DNN since the axial dimensions are usually much lower in the images of the BigNeuron dataset than our VISO-R-40 dataset. Data augmentation by transposing the three dimensions of each

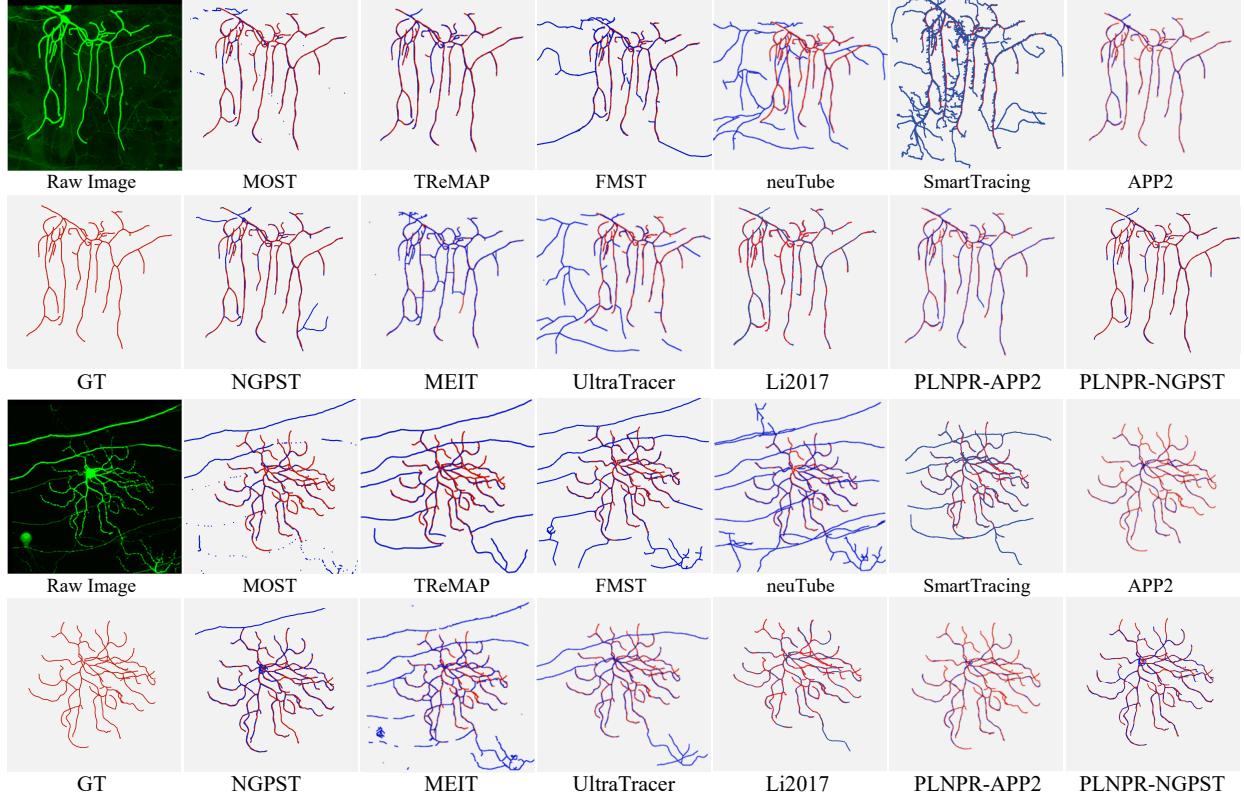


Fig. 13. Comparison of single neuron reconstruction results on two test images from the BigNeuron dataset. The reconstructed neurites are shown in blue and the corresponding ground truth (GT) are shown in red. Our method reconstructs more complete and accurate neurons compared to other methods.

TABLE II  
PERFORMANCE COMPARISON FOR SINGLE NEURON RECONSTRUCTION ON THE BIGNEURON TEST DATASET.

Method	Precision	Recall	F-Score	Jaccard	ESA	DSA	PDS
MOST [29]	0.629	0.508	0.541	0.593	31.730	38.211	0.633
TReMAP [49]	0.771	0.508	0.578	0.525	11.269	17.941	0.539
FMST [39]	0.575	0.658	0.591	0.532	17.878	23.459	0.558
neuTube [38]	0.742	0.738	0.710	0.648	12.741	21.379	0.434
SmartTracing [31]	0.701	0.722	0.613	0.465	10.430	13.430	0.547
APP2 [27]	0.875	0.582	0.681	0.667	6.063	9.079	0.496
NGPST [40]	0.710	0.757	0.728	0.592	8.830	17.489	0.463
MEIT [7]	0.689	0.729	0.702	0.396	11.635	15.772	0.544
UltraTracer [6]	0.796	0.685	0.705	0.682	10.156	17.355	0.426
Li2017 [9]	-	-	-	-	4.917	7.972	0.461
PLNPR-APP2	<b>0.877</b>	0.616	0.713	<b>0.706</b>	5.020	<b>7.753</b>	0.492
PLNPR-NGPST	0.783	<b>0.789</b>	<b>0.778</b>	0.691	<b>4.714</b>	9.458	<b>0.428</b>

training image is also performed.

3) *Comparison on BigNeuron Dataset:* On the BigNeuron dataset, we compare with ten widely-used tracing methods to validate the effectiveness of our proposed method. They are MOST [29], TReMAP [49], FMST [39], neuTube [38], SmartTracing [31], APP2 [27], NGPST [40], Li2017 [9], MEIT [7], and UltraTracer [6] respectively. While APP2 [27] performs well on the BigNeuron dataset, we also test our PLNPR with APP2 as the base tracer. “PLNPR-NGPST” means that we progressively train a DSN using the pseudo labels generated by NGPST [40]. “PLNPR-APP2” means that the DSN model is progressively trained using the pseudo labels generated by APP2 [27], which is same as the base tracer used in Li2017 [9].

The four metrics, including precision, recall, F-Score, and Jaccard, are used for comparisons for most methods. However, the implementation of the DNN-based method [9] and its reconstruction results are not available. In order to compare with [9], we only compare the three evaluation metrics reported in [9] on the same test data. The three metrics include the entire structure average (ESA), different structure average (DSA), and percentage of different structures (PDS), which are defined in [16]. While many conventional tracing methods require careful parameter tuning, we tuned parameters for each tracing method more than 10 times for each image block to find their optimal reconstructions for comparison. The quantitative comparison is shown in Table II. The weighted averages of the ESA, DSA and PDS are calculated by weighting each

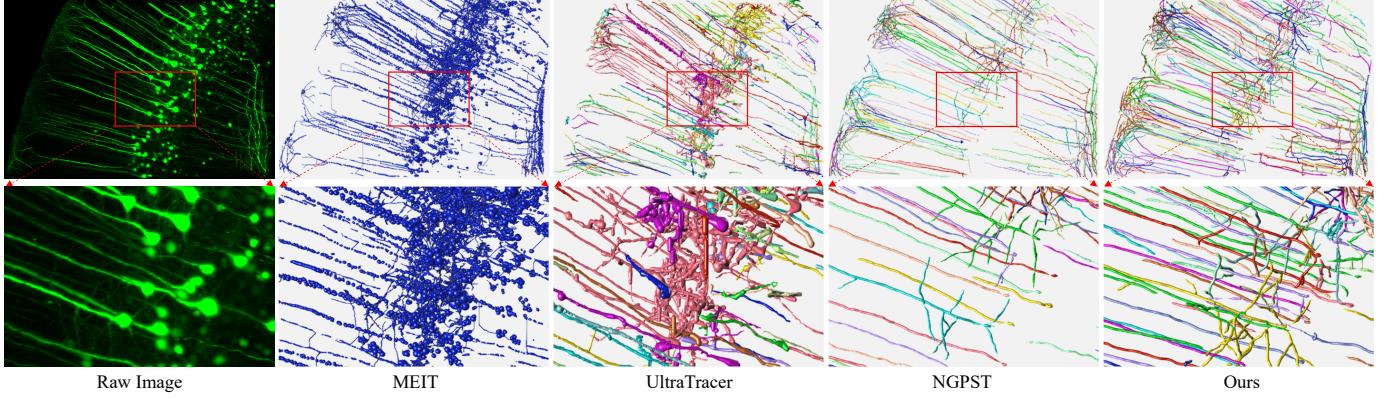


Fig. 14. Reconstruction results of dense neuronal populations from four adjacent large-scale blocks using UltraTracer [6], MEIT [7], NGPST [40] and our UltraNPR. The second row shows close-up views for a local region with dense neurites. Our method reconstructs more complete and distinguishable neurons.

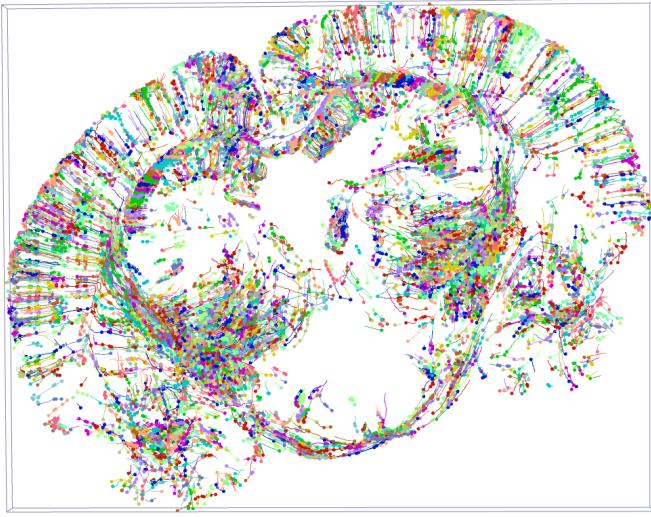


Fig. 15. The reconstruction result of neuronal populations in a large-scale 3D mouse brain slice using our UltraNPR method.

test block with the neuron length identified in the corresponding manual annotation. For these three scores, lower values presents higher coincidence between the tracing results and the manual reconstruction. From Table II, we can see that our method outperforms other methods on the BigNeuron test dataset. Comparing with the deep learning-based method [9], both our PLNPR-APP2 and PLNPR-NGPST achieve comparable performance. However, our method does not require any manual annotations to train the deep segmentation network. With ever-increasing number of unlabeled neuron datasets are collected, our method could utilize them to further improve the performance of neuron reconstruction. Fig. 13 shows the reconstructed neurons in two test images using different tracing methods.

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

To reconstruct the dense neuronal population from an ultra-scale mouse brain image, we divide the entire image into blocks in size of  $1120 \times 2048 \times 869$  considering the memory and computational cost of our PLNPR. The overlap between

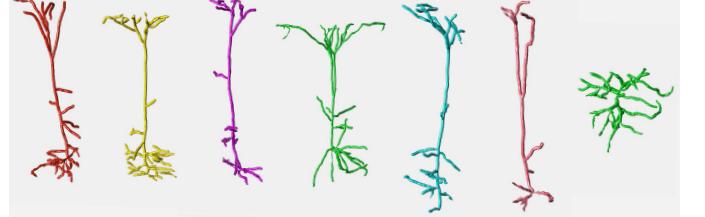


Fig. 16. Single neurons selected from the reconstructed neuronal populations in a mouse brain slice using our UltraNPR method.

adjacent blocks is set to be 300 voxels along each dimension. The  $\delta_{ovlp}$  is set to be 10 voxels, the  $\delta_{bound}$  is set to be 70 voxels and the  $\delta_{pt}$  is set to be 10 voxels for neurite fusion in Sec. III-B2. It took our UltraNPR 34 hours for image segmentation with 3D DSN, one hour for neuron reconstruction in blocks, and 10 hours for neurite fusion to reconstruct the dense neuronal population from the entire image on a cluster computer with 64 GB working memory and 20 NVIDIA 1080Ti GPUs. As shown in Fig. 15, a neuronal population consisting of 5348 neurons is successfully reconstructed by our UltraNPR from the ultra-scale noisy OM image.

Since it is infeasible to manually annotate the dense neuronal population in an ultra-scale image, quantitative evaluation of the reconstruction performance is not supported. However, we select four adjacent large-scale image blocks to qualitatively compare our UltraNPR with two state-of-the-art methods, UltraTracer [6] and MEIT [7], for large-scale neuron reconstruction. We also compare our UltraNPR with NGPST [40], which is the state-of-the-art method for neuronal population reconstruction from local image blocks. Since the sheer volume of these four image blocks is far beyond the processing capability of NGPST, we extend NGPST by using the raw image blocks as input to trace neurons and utilizing our block propagation and neurite fusion strategies to achieve the large-scale neuronal population reconstruction.

Fig. 14 compares the results of large-scale neuronal population reconstruction using different methods on the four image blocks. MEIT [7] is designed for tracing single neuron. Therefore, it can not separate individual neurons. Moreover, it fails to reconstruct the subtle neurites due to the noises and low

contrast in this challenging image. UltraTracer [6] achieves better performance of neuronal population reconstruction from the large-scale image. However, for the local regions with low signal-noise-ratio, it fails to separate individual neurons and trace complete neurites in a dense neuronal population. Although NGPST [40] can reconstruct separated neurons successfully, it still remains difficult to reconstruct complete neuronal populations from the challenging image for NGPST using hand-crafted features. In comparison, thanks to the signal enhancement by our deep network and block propagation designed for dense neurites, our UltraNPR is more robust to reconstruct a more complete neuronal population from the low-quality image while individual neurons are continuously and smoothly traced.

Several neurons selected from the reconstructed neuronal population are visualized in Fig. 16. We believe that these ultra-scale reconstructions provide detailed neuronal structures and will effectively support further neuronal morphology analysis in the whole brain. In summary, without any manual annotations and human interaction, our UltraNPR is capable of reconstructing dense neuronal populations from ultra-scale noisy OM images.

## V. CONCLUSION

In this work, we propose a novel UltraNPR framework for ultra-scale neuron population reconstruction in OM images. We first 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. Secondly, integrating PLNPR with efficient block-wise tracing and fusion strategies, our UltraNPR successfully reconstructs dense neuronal populations from an ultra-scale OM image 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 it will facilitate further brain studies, including neuron counting, neuron reconstruction, neuron morphology analysis, and so on.

## REFERENCES

- [1] J. R. Petrella, R. E. Coleman, and P. M. Doraiswamy, “Neuroimaging and early diagnosis of alzheimer disease: A look to the future,” *Radiology*, vol. 226, no. 2, pp. 315–336, Feb. 2003.
- [2] A. Giorgio and N. De Stefano, “Clinical use of brain volumetry,” *J. Magn. Reson. Imaging*, vol. 37, no. 1, pp. 1–14, Jul. 2013.
- [3] S. L. Senft, “A brief history of neuronal reconstruction,” *Neuroinformatics*, vol. 9, no. 2, pp. 119–128, Sept. 2011.
- [4] H. Wang, Q. Zhu, L. Ding, Y. Shen, C.-Y. Yang, F. Xu *et al.*, “Scalable volumetric imaging for ultrahigh-speed brain mapping at synaptic resolution,” *National Science Review*, vol. 6, no. 2, pp. 982–992, Sept. 2019.
- [5] Z. Zhou, S. A. Sorensen, and H. Peng, “Neuron Crawler: An automatic tracing algorithm for very large neuron images,” in *Proc. IEEE Int. Symp. Biomed. Imag.*, Apr. 2015, pp. 870–874.
- [6] H. Peng, Z. Zhou, E. Meijering, T. Zhao, G. A. Ascoli, and M. Hawrylycz, “Automatic tracing of ultra-volumes of neuronal images,” *Nature Methods*, vol. 14, pp. 332–333, Mar. 2017.
- [7] H. Wang, D. Zhang, Y. Song, S. Liu, R. Gao, H. Peng, and W. Cai, “Memory and Time Efficient 3D Neuron Morphology Tracing in Large-Scale Images,” in *Proc. DICTA*, Dec. 2018, pp. 1–8.
- [8] K. Xu, H. Su, J. Zhu, J. Guan, and B. Zhang, “Neuron segmentation based on CNN with semi-supervised regularization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2016, pp. 1324–1332.
- [9] R. Li, T. Zeng, H. Peng, and S. Ji, “Deep learning segmentation of optical microscopy images improves 3-D neuron reconstruction,” *IEEE Trans. Med. Imag.*, vol. 36, no. 7, pp. 1533–1541, Jul. 2017.
- [10] Z. Zhou, H.-C. Kuo, H. Peng, and F. Long, “DeepNeuron: an open deep learning toolbox for neuron tracing,” *Brain Informatics*, vol. 5, no. 2, p. 3, Jun. 2018.
- [11] M. Kozinski, A. Mosinska, M. Salzmann, and P. Fua, “Tracing in 2D to reduce the annotation effort for 3D deep delineation,” *Med. Imag. Anal.*, vol. 60, pp. 1–11, Feb. 2020.
- [12] J. Zhao, X. Chen, Z. Xiong, D. Liu, J. Zeng, Y. Zhang *et al.*, “Progressive learning for neuronal population reconstruction from optical microscopy images,” in *Proc. MICCAI*, Oct. 2019, pp. 750–759.
- [13] Y. Wang, A. Narayanaswamy, C.-L. Tsai, and B. Roysam, “A broadly applicable 3-D neuron tracing method based on open-curve snake,” *Neuroinformatics*, vol. 9, no. 2, pp. 193–217, Sept. 2011.
- [14] H. Cai, X. Xu, J. Lu, J. W. Lichtman, S. P. Yung, and S. T. C. Wong, “Repulsive force based snake model to segment and track neuronal axons in 3D microscopy image stacks,” *NeuroImage*, vol. 32, no. 4, pp. 1608–1620, Oct. 2006.
- [15] E. Bas and D. Erdogmus, “Principal curves as skeletons of tubular objects,” *Neuroinformatics*, vol. 9, no. 2, pp. 181–191, Sept. 2011.
- [16] H. Peng, Z. Ruan, D. Atasoy, and S. Sternson, “Automatic reconstruction of 3d neuron structures using a graph-augmented deformable model,” *Bioinformatics*, vol. 26, no. 12, pp. 38–46, Jun. 2010.
- [17] J. Yang, P. T. Gonzalez-Bellido, and H. Peng, “A distance-field based automatic neuron tracing method,” *BMC Bioinform.*, vol. 14, no. 1, p. 93, Mar. 2013.
- [18] J. De, L. Cheng, X. Zhang, F. Lin, H. Li, K. H. Ong *et al.*, “A graph-theoretical approach for tracing filamentary structures in neuronal and retinal images,” *IEEE Trans. Med. Imag.*, vol. 35, no. 1, pp. 257–272, Jan. 2016.
- [19] T. Zhao, J. Xie, F. Amat, N. Clack, P. Ahammad, H. Peng *et al.*, “Automated reconstruction of neuronal morphology based on local geometrical and global structural models,” *Neuroinformatics*, vol. 9, no. 2, pp. 247–261, Sept. 2011.
- [20] A. Santamaría-Pang, P. Hernandez-Herrera, M. Papadakis, P. Saggu, and I. A. Kakadiaris, “Automatic morphological reconstruction of neurons from multiphoton and confocal microscopy images using 3d tubular models,” *Neuroinformatics*, vol. 13, no. 3, pp. 297–320, Jul. 2015.
- [21] S. Navlakha, P. Ahammad, and E. W. Myers, “Unsupervised segmentation of noisy electron microscopy images using salient watersheds and region merging,” *BMC Bioinform.*, vol. 14, no. 1, p. 294, Oct. 2013.
- [22] U. Sümbül, D. Roosien, D. Cai, F. Chen, N. Barry, J. P. Cunningham *et al.*, “Automated scalable segmentation of neurons from multispectral images,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1912–1920.
- [23] T. Quan, T. Zheng, Z. Yang, W. Ding, S. Li, J. Li *et al.*, “NeuroGPS: automated localization of neurons for brain circuits using L1 minimization model,” *Scientific Reports*, vol. 3, p. 1414, Apr. 2013.
- [24] S. Liu, D. Zhang, S. Liu, D. Feng, H. Peng, and W. Cai, “Rivulet: 3D neuron morphology tracing with iterative back-tracking,” *Neuroinformatics*, vol. 14, no. 4, p. 387, Oct. 2016.
- [25] P. Frasconi, L. Silvestri, P. Soda, R. Cortini, F. S. Pavone, and G. Ianello, “Large-scale automated identification of mouse brain cells in confocal light sheet microscopy images,” *Bioinformatics*, vol. 30, no. 17, pp. 587–593, Sept. 2014.
- [26] H. Peng, F. Long, and G. Myers, “Automatic 3D neuron tracing using all-path pruning,” *Bioinformatics*, vol. 27, no. 13, pp. 239–247, Jul. 2011.
- [27] H. Xiao and H. Peng, “APP2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree,” *Bioinformatics*, vol. 29, no. 11, pp. 1448–1454, Jun. 2013.
- [28] S. Liu, D. Zhang, Y. Song, H. Peng, and W. Cai, “Automated 3-D Neuron Tracing With Precise Branch Erasing and Confidence Controlled Back Tracking,” *IEEE Trans. Med. Imag.*, vol. 37, no. 11, pp. 2441–2452, Nov. 2018.
- [29] J. Wu, Y. He, Z. Yang, C. Guo, Q. Luo, W. Zhou *et al.*, “3D BrainCV: Simultaneous visualization and analysis of cells and capillaries in a whole mouse brain with one-micron voxel resolution,” *NeuroImage*, vol. 87, pp. 199–208, Feb. 2014.
- [30] M. Liu, W. Chen, C. Wang, and H. Peng, “A Multiscale Ray-Shooting Model for Termination Detection of Tree-Like Structures in Biomedical

- Images," *IEEE Trans. Med. Imag.*, vol. 38, no. 8, pp. 1923–1934, Aug. 2019.
- [31] H. Chen, H. Xiao, T. Liu, and H. Peng, "SmartTracing: self-learning-based neuron reconstruction," *Brain Informatics*, vol. 2, no. 3, pp. 135–144, Sept. 2015.
- [32] S. Basu, W. T. Ooi, and D. Racoceanu, "Neurite tracing with object process," *IEEE Trans. Med. Imag.*, vol. 35, no. 6, pp. 1443–1451, Jun. 2016.
- [33] M. Radojević and E. Meijering, "Automated neuron tracing using probability hypothesis density filtering," *Bioinformatics*, vol. 33, no. 7, pp. 1073–1080, Jan. 2017.
- [34] C.-W. Wang, Y.-C. Lee, H. Pradana, Z. Zhou, and H. Peng, "Ensemble neuron tracer for 3d neuron reconstruction," *Neuroinformatics*, vol. 15, no. 2, pp. 185–198, Apr. 2017.
- [35] L. Gu, X. Zhang, H. Zhao, H. Li, and L. Cheng, "Segment 2D and 3D Filaments by Learning Structured and Contextual Features," *IEEE Trans. Med. Imag.*, vol. 36, no. 2, pp. 596–606, Feb. 2017.
- [36] H. Skibbe, M. Reisert, S. Maeda, M. Koyama, S. Oba, K. Ito, and S. Ishii, "Efficient monte carlo image analysis for the location of vascular entity," *IEEE Trans. Med. Imag.*, vol. 34, no. 2, pp. 628–643, Feb. 2015.
- [37] H. Skibbe, M. Reisert, K. Nakae, A. Watakabe, J. Hata, H. Mizukami *et al.*, "PAT—probabilistic axon tracking for densely labeled neurons in large 3-d micrographs," *IEEE Trans. Med. Imag.*, vol. 38, no. 1, pp. 69–78, Jan. 2019.
- [38] L. Feng, T. Zhao, and J. Kim, "neuTube 1.0: A new design for efficient neuron reconstruction software based on the swc format," *eNeuro*, vol. 2, no. 1, pp. ENEURO.0049–14.2014, Jan. 2015.
- [39] J. Yang, M. Hao, X. Liu, Z. Wan, N. Zhong, and H. Peng, "FMST: an automatic neuron tracing method based on fast marching and minimum spanning tree," *Neuroinformatics*, vol. 17, no. 2, pp. 185–196, Apr. 2019.
- [40] T. Quan, H. Zhou, J. Li, S. Li, A. Li, Y. Li *et al.*, "NeuroGPS-Tree: automatic reconstruction of large-scale neuronal populations with dense neurites," *Nature Methods*, vol. 13, pp. 51–54, Nov. 2015.
- [41] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *Proc. MICCAI*, Oct. 2016, pp. 424–432.
- [42] Q. Dou, L. Yu, H. Chen, Y. Jin, X. Yang, J. Qin *et al.*, "3D deeply supervised network for automated segmentation of volumetric medical images," *Med. Imag. Anal.*, vol. 41, pp. 40–54, Oct. 2017.
- [43] L. Yu, J.-Z. Cheng, Q. Dou, X. Yang, H. Chen, J. Qin *et al.*, "Automatic 3D cardiovascular mr segmentation with densely-connected volumetric convnets," in *Proc. MICCAI*, Sept. 2017, pp. 287–295.
- [44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, Jun. 2014.
- [45] R. Cannon, D. Turner, G. Pyapali, and H. Wheal, "An on-line archive of reconstructed hippocampal neurons," *J. Neurosci. Meth.*, vol. 84, no. 1, pp. 49–54, Oct. 1998.
- [46] H. Peng, M. Hawrylycz, J. Roskams, S. Hill, N. Spruston, E. Meijering *et al.*, "BigNeuron: large-scale 3d neuron reconstruction from optical microscopy images," *Neuron*, vol. 87, no. 2, pp. 252–256, Jul. 2015.
- [47] H. Peng, J. Tang, H. Xiao, A. Bria, J. Zhou, V. Butler *et al.*, "Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis," *Nature Communications*, vol. 5, no. 1, p. 4342, Jul. 2014.
- [48] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5693–5703.
- [49] Z. Zhou, X. Liu, B. Long, and H. Peng, "TReMAP: Automatic 3D neuron reconstruction based on tracing, reverse mapping and assembling of 2d projections," *Neuroinformatics*, vol. 14, no. 1, pp. 41–50, Jan. 2016.