Cross-dimension Affinity Distillation for 3D EM Neuron Segmentation

Xiaoyu Liu[1] Miaomiao Cai[1] Yinda Chen[1] Yueyi Zhang[1,2] Te Shi[2]
Ruobing Zhang[3,2] Xuejin Chen[1,2] Zhiwei Xiong[1,2,*]
[1]MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
[3]Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences
liuxyu@mail.ustc.edu.cn, zwxiong@ustc.edu.cn

Abstract

Accurate 3D neuron segmentation from electron microscopy (EM) volumes is crucial for neuroscience research. However, the complex neuron morphology often leads to over-merge and over-segmentation results. Recent advancements utilize 3D CNNs to predict a 3D affinity map with improved accuracy but suffer from two challenges: high computational cost and limited input size, especially for practical deployment for large-scale EM volumes. To address these challenges, we propose a novel method to leverage lightweight 2D CNNs for efficient neuron segmentation. Our method employs a 2D Y-shape network to generate two embedding maps from adjacent 2D sections, which are then converted into an affinity map by measuring their embedding distance. While the 2D network better captures pixel dependencies inside sections with larger input sizes, it overlooks inter-section dependencies. To overcome this, we introduce a cross-dimension affinity distillation (CAD) strategy that transfers inter-section dependency knowledge from a 3D teacher network to the 2D student network by ensuring consistency between their output affinity maps. Additionally, we design a feature grafting interaction (FGI) module to enhance knowledge transfer by grafting embedding maps from the 2D student onto those from the 3D teacher. Extensive experiments on multiple EM neuron segmentation datasets, including a newly built one by ourselves, demonstrate that our method achieves superior performance over state-of-the-art methods with only $\mathbf{1/20}$ *inference latency. We release our code and dataset at https://github.com/liuxy1103/CAD.*

1. Introduction

The reconstruction of neuron wiring diagrams plays a significant role in unlocking the secret of the brain and inspiring the next generation of artificial intelligence from neuroscience research [1, 3, 28, 30]. In recent years, 3D electron microscopy (EM) has become a pivotal technique for acquiring images at nanoscale resolution to trace delicate neuronal processes and synapses [11, 19, 40]. However, instance segmentation of neurons from EM volumes remains challenging due to their complicated morphologies, ambiguous boundaries, and dense distribution. Moreover, a single neuron could span the whole 3D volume, which is much larger than the field of view of existing models. Therefore, the direct application of existing instance segmentation methods [5, 12, 22, 25, 36] in natural images often leads to errors like over-merged and over-segmented neuron instances.

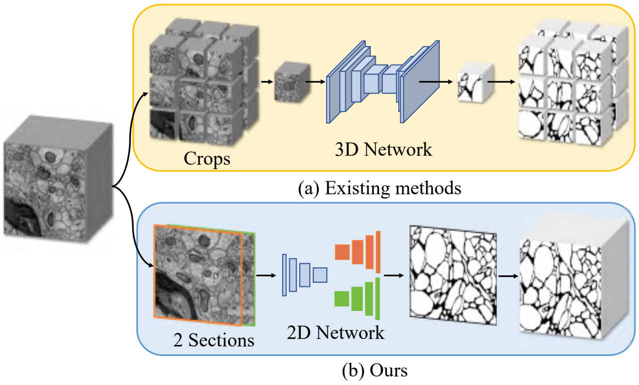

Figure 1. Comparison of different inference pipelines in the whole EM volume between existing 3D affinity modeling using a 3D network (a), and our proposed modeling using a 2D network (b).

Recently, advanced neuron segmentation approaches [7, 10, 14, 20, 21, 24, 33] based on 3D convolutional neural networks (CNNs) have achieved remarkable progress by predicting a 3D affinity map which can be converted into neuron instances with post-processing algorithms [4, 26, 35]. Although 3D CNNs can effectively capture spatial contexts in 3D and emerge as the leading neuron segmentation

[*]Corresponding author

method, the high computational and memory requirements restrict their deployment to large-scale EM datasets. Meanwhile, as illustrated in Fig. 1 (a), since 3D networks take 3D patches as input, the input size on the 2D section plane is limited. When processing large volumes, sliding windows to obtain 3D patches and stitching adjacent predictions are required, which further introduces additional errors and restricts the overall performance.

To overcome these limitations, we propose a novel method that relies on lightweight 2D CNNs to generate a 3D affinity map for efficient neuron segmentation. Specifically, we design a 2D Y-shaped network with an encoder and two decoders to simultaneously extract two embedding maps from the input of two 2D adjacent sections. We then calculate the distance between pixel embeddings from the two different embedding maps to complement the affinity information along the axial direction. As illustrated in Fig. 1 (b), this new 3D affinity modeling has three advantages: (1) It provides computational and memory savings by avoiding expensive 3D convolutions. (2) It has superior capabilities to capture spatial information of the 2D section plane due to an increased input size along the lateral direction. (3) It eliminates the need to divide and process large-scale EM volumes in multiple 3D patches, which thus avoids the additional error introduced by stitching the predicted affinities from multiple patches.

Nevertheless, the above-obtained affinity map only considers two adjacent 2D sections and lacks inter-section spatial contexts along the axial direction, resulting in ambiguities in affinity prediction. To address this, we propose a cross-dimension affinity distillation (CAD) strategy to transfer inter-section dependency knowledge from a 3D teacher network to the 2D student network by minimizing the affinity prediction discrepancy between the outputs of the two networks. Furthermore, we design a feature grafting interaction (FGI) module to enhance this knowledge transfer process. FGI grafts embedding maps from the 2D student onto those from the 3D teacher and fully calculates long-range inter-section affinities between the embeddings from the two networks. This provides complementary inter-section contextual information to refine the embeddings from the 2D CNN.

We conduct extensive experiments on multiple EM datasets which are imaged in the Drosophila melanogaster brain, the mouse somatosensory cortex, and the mouse medial entorhinal cortex. It is notable that we build a new EM neuron segmentation dataset named *Wafer4* which has a size of $125 \times 1250 \times 1250$ voxel^3 with voxel-level fine-grained annotation, to further validate the effectiveness and reliability of practical deployment of our method. Comprehensive evaluation results demonstrate that our method achieves superior performance over previous state-of-the-art methods with only $\mathbf{1/20}$ inference latency.

The contributions of this paper are as follows:

- We propose a novel method using a lightweight 2D Y-shape network to efficiently generate a 3D affinity map, alleviating the immense computational costs and the limited input size of 3D networks.
- We propose a cross-dimension affinity distillation (CAD) strategy to transfer inter-section dependency knowledge from a 3D teacher to a 2D student by enforcing affinity map consistency.
- We design a feature grafting interaction (FGI) module to enhance knowledge transfer by grafting 2D embedding maps onto 3D embedding maps.
- We establish a new EM neuron segmentation dataset with fine-grained voxel-level annotations for over 1.9×10^8 voxels. This dataset will be released as a benchmark to facilitate future research in this area.
- We conduct extensive experiments on multiple EM neuron segmentation datasets, validating superior performance and $\times\mathbf{20}$ inference efficiency improvement of the proposed method over state-of-the-art methods.

2. Related Work

2.1. Neuron Segmentation

Neuron segmentation is an extremely challenging task compared with general instance segmentation for natural images. Deep learning-based methods have provided feasible solutions, which can be roughly divided into two categories: object tracking based methods and boundary detection based methods.

Object tracking based methods [16, 28] iteratively segment and trace individual neurons using a recurrent CNN. However, reconstructing neurons one at a time is inherently inefficient. Moreover, the complex recurrent training procedure for these approaches poses difficulties. As such, object tracking methods are impractical for connectomic reconstruction which requires segmenting numerous densely packed neurons. In contrast, boundary detection based methods [8, 10, 21, 38] adopt 3D convolutional neural networks to predict an affinity map for neuron segmentation. The predicted affinities encode key instance boundaries and are post-processed into final segmentations. Designing advanced 3D CNN architectures has become a popular approach to improve affinity prediction and segmentation accuracy. For instance, Superhuman [21] incorporated long-range affinity prediction to refine nearest-neighbor affinities. MALA [10] introduced a 3D UNet with large parameters and a MALIS loss [34] during training for topologically correct segmentations. PEA [15] proposes to explicitly model affinities by measuring instance-aware embedding distance. More recently, APViT [33] introduces an appearance prompt vision transformer network for this task. However, the computational demands of these 3D networks

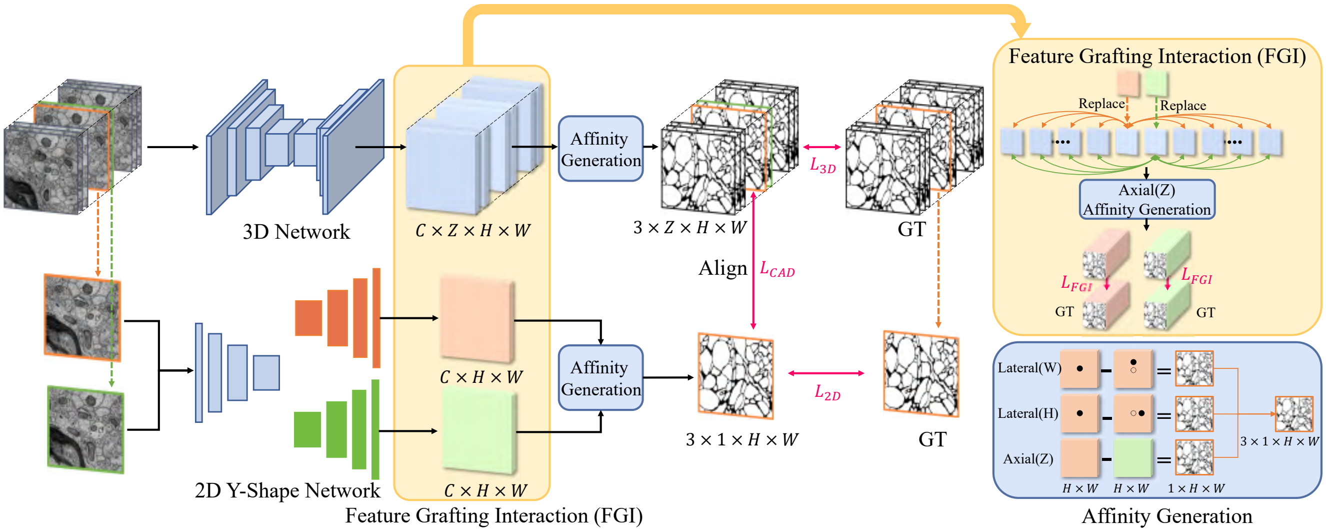

Figure 2. The workflow of our proposed framework for modeling 3D Affinity from 2D networks via cross-dimension affinity distillation (CAD) for neuron segmentation. The framework consists of two parallel networks. A 2D Y-shape network with an encoder and two decoders generates two embedding maps from adjacent 2D sections. These embedding maps are then converted into a 3D affinity map by measuring their embedding distances. In parallel, a 3D network predicts a 3D affinity map. To transfer sufficient inter-section dependency knowledge from the 3D network to the 2D network, a CAD strategy is employed to align the outputs of these predictions. Furthermore, a feature grafting interaction (FGI) module enhances the knowledge transfer process in this framework. The red and green colors indicate that the two input sections are adjacent to each other. $-$ represents the operation to measure the embedding distance.

limit their applicability to large-scale EM volumes.

In contrast to the boundary detection based methods mentioned earlier, where large-scale EM volumes are divided into multiple 3D sub-volumes and processed by 3D networks to directly predict a 3D affinity map, our proposed method introduces a novel manner for modeling 3D affinities using 2D networks. The advantage of 2D networks lies in their ability to capture pixel dependencies inside individual sections more effectively. By incorporating online knowledge distillation, our method further enhances the modeling of inter-section dependencies. As a result, our approach achieves both high performance and efficiency.

2.2. Online Knowledge Distillation

Knowledge distillation [13, 32] aims to distill knowledge from a teacher model to a student model to improve the performance of the student model. Existing knowledge distillation methods can be divided into offline distillation [23, 29, 37] and online distillation [2, 6, 41]. Earlier distillation methods often take an offline learning strategy, requiring at least two phases of training, *i.e.*, teacher model pre-training and student model distillation. The more recently proposed deep mutual learning [39] overcomes this limitation by conducting an online distillation in one-phase training between two peer student models. Online distillation can transfer the knowledge of the teacher model to the student model in real-time, and optimize end-to-end during training. This method can help the student model learn more quickly and adapt to the knowledge of the teacher model.

In this paper, we regard the 2D network and 3D network as student and teacher models and then transfer the inter-section dependency knowledge from a 3D teacher network to the 2D student network in real-time. To the best of our knowledge, we are the first to adapt the online knowledge distillation to the pair of teacher and student networks across different dimensions for EM neuron segmentation.

3. Method

In this section, we introduce the proposed framework to model 3D affinity from 2D networks via cross-dimension affinity distillation for neuron segmentation. As shown in Fig. 2, our framework contains two parallel networks, *i.e.*, a 3D network, and a 2D Y-shape network. The 3D network adopts normal affinity modeling to predict the 3D affinity map from an input 3D EM volume. The 2D network utilizes the proposed affinity modeling (in Sec. 3.2) to generate two 2D affinity maps from the input of two consecutive sections. These two embedding maps are converted into a 3D affinity map by measuring their embedding distance. To transfer inter-section dependency knowledge from the 3D network to the 2D network, we employ cross-dimension affinity distillation (CAD) to align the outputs of the two networks (in Sec. 3.3). Furthermore, a feature grafting interaction (FGI) module is introduced to enhance this knowledge transfer process (in Sec. 3.4).

3.1. Preliminary

For the 3D network, denoted as f_{3D}, an input volume is composed of Z sections, represented as $\{I_{2D}^z\}_{z=1,\ldots,Z}$, where each section has a size of $H \times W$. This input volume can be also considered as a 3D input, denoted as I_{3D}, with a size of $Z \times H \times W$. The annotated masks corresponding to I_{3D} are represented as $Y_{3D} \in \mathbb{R}^{Z \times H \times W}$.

Affinity Definition. Conventionally, we directly employ a 3D network f_{3D} to predict a 3D affinity map $A_{3D} \in \mathbb{R}^{3 \times Z \times H \times W}$, *i.e.*, $A_{3D} = f_{3D}(I_{3D})$, which represents affinity in 3 dimensions and can be converted into instance masks by post-processing algorithms. An arbitrary voxel $I_{3D}(z,h,w)$ is mapped to a group of voxel affinities $A_{3D}(1,z,h,w)$, $A_{3D}(2,z,h,w)$, and $A_{3D}(3,z,h,w)$. These affinities indicate whether the current voxel and the adjacent voxels along the Z (axial), H (lateral), and W (lateral) dimensions belong to the same instance or not, respectively. For instance, we define $A_{3D}(1,z,h,w)$ as

$$A_{3D}(1,z,h,w) = \begin{cases} 0, & \text{if } Y(z,h,w) \neq Y(z+1,h,w) \\ 1, & \text{if } Y(z,h,w) = Y(z+1,h,w) \end{cases} \tag{1}$$

where $Y(z,h,w)$ and $Y(z+1,h,w)$ are the instance segmentation IDs of paired voxels $I_{3D}(z,h,w)$ and $I_{3D}(z+1,h,w)$. 1 means that voxel $I_{3D}(z,h,w)$ and $I_{3D}(z+1,h,w)$ belong to one instance, while 0 means the opposite.

Affinity Generation. To better leverage the semantic information of instances in the feature space, explicit affinity generation methods [15, 24] have been proposed. In line with these approaches, the proposed cross-dimension affinity distillation (in Sec. 3.3) follows a similar strategy. Instead of directly predicting a 3D affinity map, we utilize the 3D network to predict a 3D embedding map, denoted as $E_{3D} \in \mathbb{R}^{C \times Z \times H \times W}$ from the 3D input. A voxel embedding $E_{3D}(z,y,x) \in \mathbb{R}^C$ is a C-dimensional feature vector.

We then adopt a cosine distance to measure the relationship between voxel embeddings for their corresponding voxel affinity. The transformation from a paired of voxel embeddings to the voxel affinity $A_{3D}(1,z,h,w)$ is formulated as

$$A_{3D}(1,z,h,w) = \frac{E_{3D}(z,h,w)E_{3D}(z+1,h,w)}{\|E_{3D}(z,h,w)\|_2 \, \|E_{3D}(z+1,h,w)\|_2}, \tag{2}$$

where $E_{3D}(z+1,y,x)$ is the adjacent voxel embedding of $E_{3D}(z,y,x)$ along the Z dimension. $ReLU$ is used to ensure affinity values of $A_{3D}(1,z,h,w)$ are in the $[0,1]$ range. For simplicity, we do not specifically represent the $ReLU$ mapping.

3.2. Modeling 3D Affinity from 2D Networks

Based on the representational meaning of affinity, we propose to model 3D voxel affinities from a 2D network f_{2D}. This 2D network is Y-shape and has an encoder and two decoders consisting of 2D convolutional operators, which preserve high inference speed and low computation cost. A more detailed architecture of the 2D network can be found in the supplementary material. As shown in Fig. 2, the 2D network predicts two 2D embedding maps E_{2D}^z and E_{2D}^{z+1} from the input of two adjacent 2D sections I_{2D}^z and I_{2D}^{z+1}, where $E_{2D}^z \in \mathbb{R}^{C \times H \times W}$.

For an arbitrary pixel $I_{2D}^z(h,w)$, we can obtain its 3D affinity along Z,H,W dimensions by

$$\begin{aligned} A_{2D}^z(1,h,w) &= \frac{E_{2D}^z(h,w)E_{2D}^{z+1}(h,w)}{\|E_{2D}^z(h,w)\|_2 \, \|E_{2D}^{z+1}(h,w)\|_2}, \\ A_{2D}^z(2,h,w) &= \frac{E_{2D}^z(h,w)E_{2D}^z(h+1,w)}{\|E_{2D}^z(h,w)\|_2 \, \|E_{2D}^z(h+1,w)\|_2}, \\ A_{2D}^z(3,h,w) &= \frac{E_{2D}^z(h,w)E_{2D}^z(h,w+1)}{\|E_{2D}^z(h,w)\|_2 \, \|E_{2D}^z(h,w+1)\|_2}, \end{aligned} \tag{3}$$

where $A_{2D}^z(1,h,w)$, $A_{2D}^z(2,h,w)$ and $A_{2D}^z(3,h,w)$ are the 3D affinity along Z,H,W dimensions. They can form a 3D voxel affinity by

$$\begin{aligned} \widetilde{A}_{3D}(z,h,w) = &Concat(A_{2D}^z(1,h,w), \\ &A_{2D}^z(2,h,w), A_{2D}^z(3,h,w)), \end{aligned} \tag{4}$$

where $\widetilde{A}_{3D}(z,h,w)$ is the 3D affinity modeled by the 2D network, and $Concat$ is the concatenation operation.

3.3. Cross-dimension Affinity Distillation

To capture inter-section spatial contexts along the axial direction (Z dimension), we propose a cross-dimension affinity distillation (CAD) strategy. This strategy serves as an online knowledge distillation algorithm, transferring inter-section dependency knowledge from the 3D teacher network to the 2D student network. The goal is to ensure consistency between their outputs by minimizing the mean squared error (MSE) loss between the affinity predictions of the 2D and 3D networks. The CAD loss, denoted as $\mathcal{L}_{CAD}$, is calculated as

$$\mathcal{L}_{CAD} = \frac{1}{Z \times H \times W} \sum_{z=1}^{Z} \sum_{h=1}^{H} \sum_{w=1}^{W} \left\| \widetilde{A}_{3D} - A_{3D} \right\|_2, \tag{5}$$

where $\widetilde{A}_{3D}$ represents the affinity predictions of the 2D network obtained by processing the entire volume sequence in sequence, and A_{3D} represents the affinity predictions of the 3D network.

3.4. Feature Grafting Interaction

To further enhance knowledge transfer from the 3D network to the 2D network, we propose a feature grafting interaction (FGI) module. The FGI module enables the grafting of 2D embeddings onto 3D feature maps, and this interaction involves fully calculating inter-section long-range affinities between the embeddings from the 2D CNN and the 3D CNN. For a predicted pixel embedding $E_{2D}^{z}(h,w)$ obtained from the 2D CNN, we insert it into the predicted 3D embedding map E_{3D} to replace $E_{3D}(z,h,w)$. This grafting process facilitates the incorporation of the 2D embedding information into the 3D representation. Subsequently, we measure the embedding distance between $E_{2D}^{z}(h,w)$ and other embeddings located in different sections along the Z dimension. These measurements allow us to calculate long-range affinities, capturing the relationships between the 2D and 3D embeddings across different sections in the volume. The calculation of the long-range affinity in FGI is formulated as

$$A_{FGI}^{z'}(z,h,w) = \frac{E_{2D}^{z}(h,w)E_{3D}(z',h,w)}{\left\|E_{2D}^{z}(h,w)\right\|_2 \left\|E_{3D}(z',h,w)\right\|_2}, \quad (6)$$

where z' indicates the sequence number in the input 3D volume, and $z' \neq z$.

These long-range affinities along the Z dimension are supervised by their corresponding affinity labels with an FGI loss $\mathcal{L}_{FGI}$, which is calculated as

$$\mathcal{L}_{FGI} = \sum_{z,z'=1,z\neq z'}^{Z} \sum_{h=1}^{H} \sum_{w=1}^{W} \frac{\left\| A_{FGI}^{z'}(z,h,w) - A_{gt}^{z'}(z,h,w) \right\|_2}{(Z-1)\times H \times W}, \quad (7)$$

where $A_{gt}^{z'}(z,h,w)$ is the affinity label to indicate if the voxel $I_{3D}(z',h,w)$ and $I_{3D}(z,h,w)$ belong to the same instance.

3.5. Overall Optimization

In the training stage, we initialize the 2D network f_{2D} and the 3D network f_{3D} independently. The supervision for both networks is provided by the 3D affinity label $A_{gt} \in \mathbb{R}^{3\times Z\times H\times W}$, which is generated from the annotated masks Y_{3D}. We adopt the MSE loss to supervise both f_{2D} and f_{3D} as

$$\begin{aligned} \mathcal{L}_{2D} &= \frac{1}{Z\times H\times W} \sum_{z=1}^{Z}\sum_{h=1}^{H}\sum_{w=1}^{W} \left\| \widetilde{A}_{3D} - A_{gt} \right\|_2, \\ \mathcal{L}_{3D} &= \frac{1}{Z\times H\times W} \sum_{z=1}^{Z}\sum_{h=1}^{H}\sum_{w=1}^{W} \left\| A_{3D} - A_{gt} \right\|_2, \end{aligned} \quad (8)$$

where L_{2D} and L_{3D} represent the loss functions for f_{2D} and f_{3D}, respectively.

The overall objective function, denoted as $\mathcal{L}$, is a combination of the 2D loss $\mathcal{L}_{2D}$, the 3D loss $\mathcal{L}_{3D}$, the CAD loss $\mathcal{L}_{CAD}$, and the FGI loss $\mathcal{L}_{FGI}$. Thus, the objective function can be expressed as

$$\mathcal{L} = \mathcal{L}_{2D} + \mathcal{L}_{3D} + \lambda_1 \mathcal{L}_{CAD} + \lambda_2 \mathcal{L}_{FGI}, \quad (9)$$

where the weights λ_1 and λ_2 are trade-off weights that control the importance of the corresponding loss terms.

In the inference stage, we can input the entire volume sequence to the 2D network in sequence to obtain the 3D affinity map with the same effect as the 3D network. The 3D affinity map is converted into final instance segmentation results by different post-processing algorithms.

4. Experiments

4.1. Datasets

CREMI. The CREMI dataset [9] is widely used for 3D EM neuron segmentation and is derived from adult Drosophila melanogaster brain tissue. The imaging resolution of the dataset is $4\times4\times40$ nm. It consists of three sub-volumes (CREMI-A/B/C), each containing 125 consecutive images. For training and testing purposes, the dataset is divided into 100 sections for training and 25 sections for testing.

AC3/4. AC3 and AC4 are labeled subsets extracted from the mouse somatosensory cortex dataset [17], which is a widely used EM dataset for 3D neuron instance segmentation. The images in this dataset were acquired at a resolution of $3\times3\times29$ nm. The AC3 dataset consists of 256 sequential images, while the AC4 dataset contains 100 sequential images. To evaluate our proposed method, we partitioned the data as follows: the top 80 sections of AC4 are used for training, 20 sections for validation, and the top 100 sections of AC3 for testing.

Wafer4. The Wafer4 dataset is collected from a region of the mouse medial entorhinal cortex and imaged at a resolution of $8\times8\times35$ nm with the Multi-Beam-SEM technology. We have established this new EM neuron segmentation dataset, which has a size of $125\times1250\times1250$ voxel^3 with fine-grained voxel-level annotations for over 1.9×10^8 voxels. This dataset is divided into 100 sections for training and 25 sections for testing.

4.2. Metrics

We employ two commonly used metrics to quantitatively evaluate the segmentation results: the Variation of Information (VOI) and the Adapted Rand Error ($ARAND$). The VOI metric [27] measures the dissimilarity between two segmentation masks, taking into account both over-merging and over-segmentation errors. It can be further decomposed into two components, VOI_S and VOI_M, which evaluate the extent of over-segmentation and over-merging errors, respectively. On the other hand, the ARAND metric is derived from the Rand Index and incorporates adjustments to compensate for the uneven distribution of object sizes in EM image segmentation [31]. The ARAND metric quantifies the agreement between the ground truth and the segmented results. It is worth noting that lower values of both metrics indicate better segmentation performance.

Post.	Method	CREMI-A				CREMI-B				CREMI-C			
		$VOI_S\downarrow$	$VOI_M\downarrow$	$VOI\downarrow$	$ARAND\downarrow$	$VOI_S\downarrow$	$VOI_M\downarrow$	$VOI\downarrow$	$ARAND\downarrow$	$VOI_S\downarrow$	$VOI_M\downarrow$	$VOI\downarrow$	$ARAND\downarrow$
Waterz	Superhuman [21]	0.3991	0.2405	0.6396	0.0887	0.5540	0.2215	0.7755	0.0482	0.8204	0.3375	1.1579	0.1793
	MALA [10]	0.3980	0.2356	0.6336	0.0846	0.5892	0.2608	0.8501	0.0407	0.8415	0.3324	1.1739	0.1621
	PEA [15]	0.3287	0.2977	0.6264	0.0909	0.4106	0.3741	0.7847	0.0407	0.7449	0.4464	1.1914	0.1689
	APViT [33]	0.4447	0.2595	0.7041	0.1169	0.5793	0.2014	0.7807	0.0319	0.8839	0.2341	1.1181	**0.1102**
	Ours w/o KD	0.3259	0.2986	0.6245	0.1067	0.4017	0.3472	0.7489	0.0445	0.7384	0.4547	1.1931	0.1695
	Ours	0.3132	0.2521	**0.5653**	**0.0788**	0.3793	0.3051	**0.6844**	**0.0297**	0.7381	0.3216	**1.0597**	0.1487
LMC	Superhuman [21]	0.5243	0.2429	0.7672	0.1177	0.7612	0.1859	0.9471	0.0394	1.0102	0.2330	1.2432	0.1579
	MALA [10]	0.5094	0.2483	0.7577	0.1064	0.8635	0.2157	1.0793	0.0494	1.0485	0.2622	1.3107	0.1660
	PEA [15]	0.4117	0.2515	0.6633	0.0964	0.5597	0.2347	0.7943	0.0353	0.9078	0.2510	1.1589	0.1517
	APViT [33]	0.4336	0.2914	0.7249	0.1304	0.5777	0.2162	0.7939	**0.0340**	0.8719	0.2527	**1.1247**	**0.1116**
	Ours w/o KD	0.3835	0.2418	0.6253	0.0759	0.5795	0.2202	0.7997	0.0404	0.8725	0.3027	1.1752	0.1581
	Ours	0.3719	0.2345	**0.6063**	**0.0691**	0.5394	0.2326	**0.7720**	0.0349	0.8605	0.2796	1.1401	0.1563

Table 1. Quantitative comparison of segmentation results on CREMI datasets. 'Ours w/o KD' represents the proposed 3D affinity modeling from the 2D network without using the CAD strategy and the FGI module. 'Post.' represents the post-processing algorithms. The best results and the second-best results are highlighted in bold and underlined.

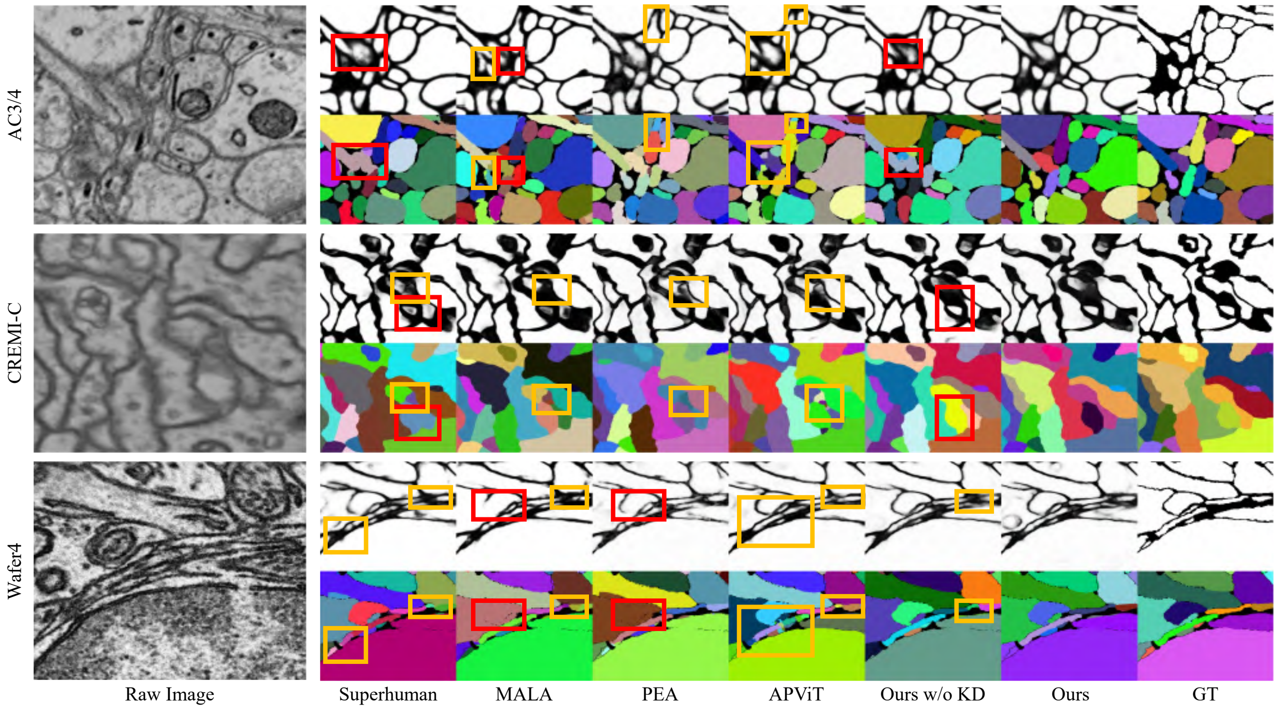

Figure 3. The 2D visual results on various datasets. In each dataset, the first row displays the affinity map, while the second row shows the corresponding instance segmentation result. Red and orange boxes indicate merge and split errors, respectively.

4.3. Implementation Details

We conduct all experiments using the Adam optimizer [18] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a learning rate of $1e-4$, and a batch size of 2. The experiments are performed on a single NVIDIA TitanXP GPU for a total of 200,000 iterations. To account for the faster convergence of the 2D network compared to the 3D network, we introduce the $\mathcal{L}_{CAD}$ and $\mathcal{L}_{FGI}$ loss terms after 10,000 iterations. This ensures a balanced training process. The model configurations and hyper-parameters are determined on the validation set of AC3/4. Once the hyper-parameters are determined, they are fixed for other datasets.

We utilize the same 3D network as in previous

	Post.	Method	$VOI_S \downarrow$	$VOI_M \downarrow$	$VOI \downarrow$	$ARAND \downarrow$
AC3/4	Waterz	Superhuman [21]	0.5973	0.4332	1.0305	0.1794
		MALA [10]	0.6767	0.4571	1.1338	0.1664
		PEA [15]	0.5522	0.4980	1.0502	0.2093
		APViT [33]	0.7671	0.2039	<u>0.9764</u>	**0.0775**
		Ours w/o KD	0.6008	0.4309	1.0317	0.1187
		Ours	0.5326	0.3509	**0.8835**	<u>0.0808</u>
	LMC	Superhuman [21]	1.1253	0.1891	1.3144	0.1015
		MALA [10]	1.0778	0.2435	1.3213	0.1113
		PEA [15]	0.8061	0.3052	1.1112	0.1300
		APViT [33]	0.8231	0.2054	<u>1.0285</u>	<u>0.0940</u>
		Ours w/o KD	0.8487	0.2536	1.1023	0.1094
		Ours	0.6966	0.22451	**0.9212**	**0.0771**
Wafer4	Waterz	Superhuman [21]	0.4518	0.1658	0.6176	0.0411
		MALA [10]	0.4552	0.1581	0.6133	0.0361
		PEA [15]	0.4208	0.1722	<u>0.5930</u>	<u>0.0341</u>
		APViT [33]	0.5813	0.1226	0.7039	0.0362
		Ours w/o KD	0.4039	0.2235	0.6274	0.0505
		Ours	0.4149	0.1439	**0.5587**	**0.0302**
	LMC	Superhuman [21]	0.7367	0.1411	0.8778	0.0390
		MALA [10]	0.7573	0.1403	0.8976	0.0365
		PEA [15]	0.6135	0.1578	0.7713	0.0375
		APViT [33]	0.6456	0.1114	0.7570	<u>0.0355</u>
		Ours w/o KD	0.5987	0.1580	<u>0.7567</u>	0.0373
		Ours	0.5792	0.1501	**0.7289**	**0.0344**

Table 2. Quantitative comparison of segmentation results on AC3/4 and Wafer4 datasets.

works [15, 21] to predict a 3D affinity map in our proposed training framework. During the inference stage, the affinity map predicted by the 2D network is processed into neuron instances using two different post-processing algorithms: waterz [10] and LMC [4]. It is important to note that we maintain consistent post-processing settings across all our experiments. These settings align with existing baseline methods to ensure that the conclusions drawn from our method are not influenced by the post-processing step.

4.4. Comparison with State-of-the-art Methods

Quantitative Segmentation Results. We list an extensive quantitative comparison of the CREMI datasets in Table 1 and AC3/4 and Wafer4 datasets in Table 2. Our proposed method achieves superior performance over existing state-of-the-art approaches in most cases. In Table 1, using Waterz post-processing, our method outperforms the second-best methods on the VOI metric by 9.7%, 12.3%, and 5.2% on Cremi-A, Cremi-B, and Cremi-C respectively. Notably, even without our proposed cross-dimension affinity distillation, our proposed 3D affinity modeling from a 2D network can surpass existing approaches in VOI on Cremi-A and Cremi-B. In Table 2, our method achieves the lowest VOI using both Waterz and LMC post-processing on AC3/4 and wafer4 datasets. For instance, we reduced the VOI metric by 9.5% and 5.7% compared to the second-best method on AC3/4 and wafer4 respectively. Overall, these results validate the advantages of our proposed approach for 3D neuron segmentation across diverse datasets.

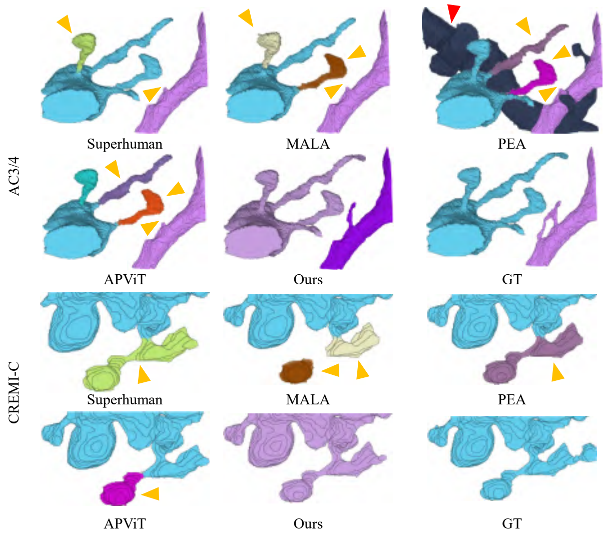

Figure 4. The 3D visual results on the AC3/4 and CREMI-C dataset. The orange and red arrows indicate the split and merge errors in the 3D structure.

Qualitative Segmentation Results. We present the 2D and 3D visual comparison results in Fig. 3 and Fig. 4, respectively. From these figures, it is evident that our proposed method outperforms other baseline methods in terms of predicting affinity maps with higher fidelity, resulting in a significant reduction in split and merge errors. As observed, our method accurately preserves the 3D structures of the neuron, surpassing the performance of other methods. Additionally, we provide supplementary material that includes more visual comparison results and the analysis of the visualization of the embedding maps from the 3D and the 2D networks.

Model Complexity Comparison. To validate the computational advantage of our proposed method in the inference stage, we conduct experiments on the test set of the AC3/4 dataset, which has a size of $100 \times 1024 \times 1024$ voxel^3. We compare the model complexity of different methods in Table 3. Our method has the following advantages:

(1) Smaller model size and computational complexity: Compared to light 3D CNNs utilized in Superhuman [21] and PEA [15] and a heavy 3D CNN used in MALA [10], our method based on the 2D network has the smallest number of parameters and floating-point operations (FLOPs). Additionally, it exhibits the lowest GPU memory occupancy. On

Method	VOI	#Params (M)	FLOPs (GMAC)	Latency(s)
Superhuman [21]	1.0305	1.48	147520.10	113.83
MALA [10]	1.1338	84.02	298936.88	160.94
PEA [15]	1.0502	1.48	147824.30	374.02
APViT [33]	0.9764	37.25	60197.80	345.50*
Ours	**0.8835**	**0.88**	**7509.00**	**17.29**

Table 3. Quantitative comparison of model complexity and inference latency (seconds) on the AC3/4 testset. *Note that the GPU usage of APViT is high and its inference is performed on one NVIDIA 3090 GPU. VOI results are obtained by the Waterz post-processing.

the contrary, APViT [33] based on the transformer backbone requires a significant amount of GPU memory during inference. It is worth noting that we replace the NVIDIA XP GPU (12 GB) with the NVIDIA 3090 GPU (24 GB) for normal inference of APViT. The high GPU memory requirement of APViT is not practical for deployment, as it would necessitate hundreds of GPUs to process large-scale data. Our method offers a more cost-effective and deployment-friendly solution.

(2) Reduced inference latency: Our method achieves the lowest inference latency, making it highly suitable for practical deployment. The low latency is attributed to two factors: the reduced computational complexity and the ability of our method to handle larger input sizes of 2D data. This eliminates the need to divide and process large-scale EM volumes in multiple 3D patches, as required by the 3D network. While MALA [10] is designed for larger 3D inputs and fewer 3D patches divided, it still experiences significant delays when confronted with the aforementioned factors affecting inference.

4.5. Ablation Studies and Analysis

We conduct ablation studies on the AC3/4 testset and report VOI results. As shown in Table 4, we conduct an ablation study on the main components of our proposed method, including the CAD strategy and the FGI module. We compare different combinations of CAD and FGI for the 3D network f_{3D} and the 2D network f_{2D}. The results demonstrate that CAD plays a crucial role in enhancing the performance of both the 2D and 3D networks. This is attributed to the online knowledge distillation technique employed by CAD, which facilitates the transfer of cross-intersection dependency knowledge from the 3D network to the 2D network, as well as the transfer of intra-section pixel dependency knowledge from the 2D network to the 3D network. This mutual knowledge transfer contributes to the improved performance of both the 2D and 3D networks. Furthermore, the FGI module facilitates the interaction between the knowledge of the 3D network f_{3D} and the 2D network f_{2D} by calculating the long-range affinity across multiple sections. This allows us to explicitly calculate the long-range affinities and proves to be an additional factor in improving the performance of both the 3D network f_{3D} and the 2D network f_{2D}, thereby facilitating the processing of CAD. The combination of CAD and FGI exhibits the most favorable performance, as observed from the results.

Method	CAD	FGI	Waterz	LMC
f_{3D}	✗	✗	1.0502	1.1112
	✓	✗	0.9460	1.0614
	✓	✓	**0.8838**	**0.9977**
f_{2D}	✗	✗	1.0317	1.1023
	✓	✗	0.9259	0.9490
	✓	✓	**0.8835**	**0.9212**

Table 4. Ablation study on different components of the proposed cross-dimension affinity distillation strategy.

λ_1	0.1	1.0	10
Waterz / LMC	0.6572 / 0.7552	**0.5915 / 0.6705**	0.6765 / 0.6941
λ_2	0.1	1.0	10
Waterz / LMC	0.6117 / 0.6902	**0.5915 / 0.6705**	0.6045 / 0.7170

Table 5. Ablation study on loss weights of our framework.

4.6. Hyper-parameters determination

We perform experiments on the AC3/4 validation set to assess the influence of hyper-parameters λ_1 and λ_2 on balancing the impact of the CAD strategy and the FGI module in the overall optimization objective of our proposed framework. The results are presented in Table 5. Different values are tested to determine an appropriate weight. Based on the experimental findings, we empirically set both λ_1 and λ_2 to 1. This choice is made to ensure the optimal performance.

5. Conclusion

We propose a novel method to efficiently generate a 3D affinity map from the 2D network for neuron segmentation in EM volumes. We propose a cross-dimension affinity distillation (CAD) strategy to transfer inter-slice dependency knowledge from the 3D network to the 2D network by enforcing consistency between their predicted affinity maps. Furthermore, we design a feature grafting interaction (FGI) module to enhance this process by grafting embedding from the 2D network into those from the 3D network. Extensive experiments on multiple EM datasets demonstrate that our 2D affinity modeling method achieves superior neuron segmentation performance compared to the previous methods.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62021001.

References

[1] Larry F Abbott, Davi D Bock, Edward M Callaway, Winfried Denk, Catherine Dulac, Adrienne L Fairhall, Ila Fiete, Kristen M Harris, Moritz Helmstaedter, Viren Jain, et al. The mind of a mouse. *Cell*, 182(6):1372–1376, 2020. 1

[2] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. In *ICLR*, 2018. 3

[3] Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireşan, Alessandro Giusti, Luca M Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Front. Neuroanat.*, 9:142, 2015. 1

[4] Thorsten Beier, Constantin Pape, Nasim Rahaman, Timo Prange, Stuart Berg, Davi D Bock, Albert Cardona, Graham W Knott, Stephen M Plaza, Louis K Scheffer, et al. Multicut brings automated neurite segmentation closer to human performance. *Nature Methods*, 14(2):101–102, 2017. 1, 7

[5] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 1

[6] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. In *AAAI*, 2020. 3

[7] Yinda Chen, Wei Huang, Shenglong Zhou, Qi Chen, and Zhiwei Xiong. Self-supervised neuron segmentation with multi-agent reinforcement learning. *IJCAI*, 2023. 1

[8] Dan Ciresan, Alessandro Giusti, Luca Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *NeurIPS*, 2012. 2

[9] CREMI. Miccai challenge on circuit reconstruction from electron microscopy images. `https://cremi.org/`, 2016. 5

[10] Jan Funke, Fabian Tschopp, William Grisaitis, Arlo Sheridan, Chandan Singh, Stephan Saalfeld, and Srinivas C Turaga. Large scale image segmentation with structured loss based deep learning for connectome reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1669–1680, 2019. 1, 2, 6, 7, 8

[11] Kristen M Harris, Elizabeth Perry, Jennifer Bourne, Marcia Feinberg, Linnaea Ostroff, and Jamie Hurlburt. Uniform serial sectioning for transmission electron microscopy. *J. Neurosci.*, 26(47):12101–12103, 2006. 1

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[14] Wei Huang, Chang Chen, Zhiwei Xiong, Yueyi Zhang, Xuejin Chen, Xiaoyan Sun, and Feng Wu. Semi-supervised neuron segmentation via reinforced consistency learning. *IEEE Transactions on Medical Imaging*, 41(11):3016–3028, 2022. 1

[15] Wei Huang, Shiyu Deng, Chang Chen, Xueyang Fu, and Zhiwei Xiong. Learning to model pixel-embedded affinity for homogeneous instance segmentation. In *AAAI*, 2022. 2, 4, 6, 7, 8

[16] Michał Januszewski, Jörgen Kornfeld, Peter H Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. High-precision automated reconstruction of neurons with flood-filling networks. *Nature Methods*, 15(8):605–610, 2018. 2

[17] Narayanan Kasthuri, Kenneth Jeffrey Hayworth, Daniel Raimund Berger, Richard Lee Schalek, José Angel Conchello, Seymour Knowles-Barley, Dongil Lee, Amelio Vázquez-Reina, Verena Kaynig, Thouis Raymond Jones, et al. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015. 5

[18] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[19] Graham Knott, Herschel Marchman, David Wall, and Ben Lich. Serial section scanning electron microscopy of adult brain tissue using focused ion beam milling. *J. Neurosci.*, 28(12):2959–2964, 2008. 1

[20] Kisuk Lee, Ran Lu, Kyle Luther, and H Sebastian Seung. Learning and segmenting dense voxel embeddings for 3d neuron reconstruction. *IEEE Transactions on Medical Imaging*, 40(12):3801–3811, 2021. 1

[21] Kisuk Lee, Jonathan Zung, Peter Li, Viren Jain, and H Sebastian Seung. Superhuman accuracy on the snemi3d connectomics challenge. *arXiv preprint arXiv:1706.00120*, 2017. 1, 2, 6, 7, 8

[22] Dongnan Liu, Donghao Zhang, Yang Song, Heng Huang, and Weidong Cai. Panoptic feature fusion net: a novel instance segmentation paradigm for biomedical and biological images. *IEEE Transactions on Image Processing*, 30:2045–2059, 2021. 1

[23] Xiaoyu Liu, Bo Hu, Wei Huang, Yueyi Zhang, and Zhiwei Xiong. Efficient biomedical instance segmentation via knowledge distillation. In *MICCAI*, 2022. 3

[24] Xiaoyu Liu, Wei Huang, Zhiwei Xiong, Shenglong Zhou, Yueyi Zhang, Xuejin Chen, Zheng-Jun Zha, and Feng Wu. Learning cross-representation affinity consistency for sparsely supervised biomedical instance segmentation. In *ICCV*, 2023. 1, 4

[25] Xiaoyu Liu, Wei Huang, Yueyi Zhang, and Zhiwei Xiong. Biological instance segmentation with a superpixel-guided graph. In *IJCAI*, 2022. 1

[26] Xiaoyu Liu, Yueyi Zhang, Zhiwei Xiong, Chang Chen, Wei Huang, Xuejin Chen, and Feng Wu. Learning neuron stitching for connectomics. In *MICCAI*, 2021. 1

[27] Marina Meilă. Comparing clusterings by the variation of information. In *LTKM*. 2003. 5

[28] Yaron Meirovitch, Lu Mi, Hayk Saribekyan, Alexander Matveev, David Rolnick, and Nir Shavit. Cross-classification clustering: An efficient multi-object tracking technique for 3-d instance segmentation in connectomics. In *CVPR*, 2019. 1, 2

[29] Julia MH Noothout, Nikolas Lessmann, Matthijs C Van Eede, Louis D van Harten, Ecem Sogancioglu, Friso G Heslinga, Mitko Veta, Bram van Ginneken, and Ivana Išgum. Knowledge distillation with ensembles of convolutional neural networks for medical image segmentation. *Journal of Medical Imaging*, 9(5):052407–052407, 2022. 3

[30] Stephen M Plaza, Louis K Scheffer, and Dmitri B

Chklovskii. Toward large-scale connectome reconstructions. *Current opinion in neurobiology*, 25:201–210, 2014. 1

[31] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971. 5

[32] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 3

[33] Rui Sun, Naisong Luo, Yuwen Pan, Huayu Mai, Tianzhu Zhang, Zhiwei Xiong, and Feng Wu. Appearance prompt vision transformer for connectome reconstruction. In *IJCAI*, 2023. 1, 2, 6, 7, 8

[34] Srinivas C. Turaga, Kevin L. Briggman, Moritz Helmstaedter, Winfried Denk, and H. Sebastian Seung. Maximin affinity learning of image segmentation. In *NeurIPS*, 2009. 2

[35] Steffen Wolf, Alberto Bailoni, Constantin Pape, Nasim Rahaman, Anna Kreshuk, Ullrich Köthe, and Fred A Hamprecht. The mutex watershed and its objective: Efficient, parameter-free graph partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3724–3738, 2020. 1

[36] Jingru Yi, Hui Tang, Pengxiang Wu, Bo Liu, Daniel J Hoeppner, Dimitris N Metaxas, Lianyi Han, and Wei Fan. Object-guided instance segmentation for biological images. In *AAAI*, 2020. 1

[37] Nikos Zagoruyko, Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 3

[38] Tao Zeng, Bian Wu, and Shuiwang Ji. DeepEM3D: approaching human-level performance on 3D anisotropic EM image segmentation. *Bioinformatics*, 33(16):2555–2562, 2017. 2

[39] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, 2018. 3

[40] Zhihao Zheng, J Scott Lauritzen, Eric Perlman, Camenzind G Robinson, Matthew Nichols, Daniel Milkie, Omar Torrens, John Price, Corey B Fisher, Nadiya Sharifi, et al. A complete electron microscopy volume of the brain of adult drosophila melanogaster. *Cell*, 174(3):730–743, 2018. 1

[41] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, 2018. 3