BioSAM: Generating SAM Prompts From Superpixel Graph for Biological Instance Segmentation

Miaomiao Cai, Xiaoyu Liu, Zhiwei Xiong, *Member, IEEE*, and Xuejin Chen, *Member, IEEE*

***Abstract*—Proposal-free instance segmentation methods have significantly advanced the field of biological image analysis. Recently, the Segment Anything Model (SAM) has shown an extraordinary ability to handle challenging instance boundaries. However, directly applying SAM to biological images that contain instances with complex morphologies and dense distributions fails to yield satisfactory results. In this work, we propose BioSAM, a new biological instance segmentation framework generating SAM prompts from a superpixel graph. Specifically, to avoid over-merging, we first generate sufficient superpixels as graph nodes and construct an initialized graph. We then generate initial prompts from each superpixel and aggregate them through a graph neural network (GNN) by predicting the relationship of superpixels to avoid over-segmentation. We employ the SAM encoder embeddings and the SAM-assisted superpixel similarity as new features for the graph to enhance its discrimination capability. With the graph-based prompt aggregation, we utilize the aggregated prompts in SAM to refine the segmentation and generate more accurate instance boundaries. Comprehensive experiments on four representative biological datasets demonstrate that our proposed method outperforms state-of-the-art methods.**

***Index Terms*—Biological Instance Segmentation, segment anything, prompt generation, superpixel.**

I. Introduction

INSTANCE segmentation plays a crucial role in biological image analysis, by identifying and examining the morphology, distribution, and phenotyping of biological entities [1], [2], [3], [4], [5]. Existing instance segmentation methods can be categorized into proposal-based methods [6], [7], [8] and proposal-free methods [9], [10]. Proposal-based instance segmentation methods [6], [7], [8] first utilize a detection head to localize each instance by a bounding box and then use a segmentation head to predict the mask in each bounding box. However, these methods heavily rely on the quality of the detected bounding boxes. Specifically, due to the irregular shapes and dense distribution of instances in biological images, detected bounding boxes often overlap. Moreover, when the instance size is larger than the receptive field of the model, like neurons in Electron Microscopy (EM) images, these methods usually fail to detect the whole instances. To solve the aforementioned problems, proposal-free methods [9], [10], which do not rely on bounding boxes, have been proposed. Proposed-free methods first extract well-designed embeddings or morphology features and then apply a post-processing algorithm (e.g. Mean-Shift algorithm [11] or graph cut [12]) to produce the final instance segmentation result. However, since they heavily rely on non-learnable post-processing algorithms, which are hand-crafted optimizations, they lack robustness to diverse data. As a result, the proposal-free methods tend to obtain inaccurate instance boundaries, leading to suboptimal segmentation results.

The advent of the large vision model, e.g. the segment anything model (SAM) [13], brings tremendous change to the field of image segmentation due to its extraordinary ability to handle complex instance boundaries. The accurate human prompts (including points, bounding boxes, and masks) for SAM can contribute to superior segmentation [14], [15], [16], [17]. However, this approach is time-consuming and labor-intensive. Therefore, SAM proposes the "everything" mode that does not necessitate any manual annotations. Specifically, SAM automatically generates a regular grid of points as prompts and then segments everything at each point [13]. SAM utilizes Non-Maximum Suppression (NMS) to merge overlapping masks and reduce redundancy. The grid-based prompts achieve decent results in natural scenes with regular morphologies and positions. However, when applied to biological images with complex instance morphologies and dense distributions, it fails to yield satisfactory results because the grid-based prompts do not provide specific prompts for instances characterized by such morphologies and distributions. As illustrated in Fig. 1, on the one hand, it has limited capability in handling the complex spatial relationships between instances in biological images, resulting in over-segmentation and over-merging errors. On the other hand, it struggles to capture the complex morphologies and

Received 30 May 2024; revised 1 September 2024; accepted 28 September 2024. Date of publication 4 October 2024; date of current version 8 January 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62076230 and in part by the Fundamental Research Funds for the Central Universities under Grant WK3490000008 and Grant WK9100000063. *(Corresponding author: Xuejin Chen.)*

The authors are with the MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China, Hefei 230088, China, and also with the Anhui Province Key Laboratory of Biomedical Imaging and Intelligent Processing, Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China (e-mail: mmcai@mail.ustc.edu.cn; liuxyu@mail.ustc.edu.cn; zwxiong@ustc.edu.cn; xjchen99@ustc.edu.cn).

Digital Object Identifier 10.1109/JBHI.2024.3474706

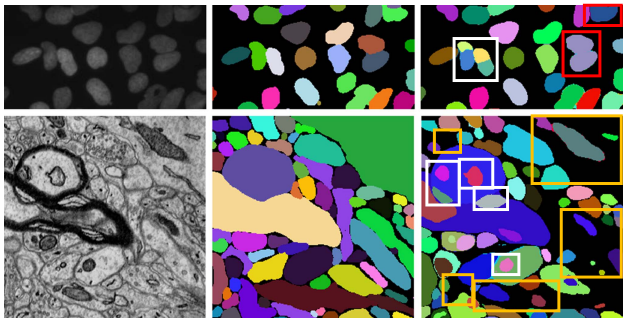

Fig. 1. Challenges encountered by SAM in processing biological images. (1) The dense and uneven distribution of biological instances leads to over-segmentation errors (indicated by white boxes) and over-merging errors (indicated by red boxes). (2) The complex instance morphologies and intricate structures result in incomplete segmentation and inaccurate boundaries (indicated by yellow boxes).

intricate structures of instances in biological images, resulting in incomplete segmentations.

In this paper, we propose BioSAM, a new biological instance segmentation framework that generates prompts from a superpixel graph and then refines the instance segmentation by SAM. These prompts contain richer information on the spatial distribution and boundary morphology of instances compared to automatically generated grid-based point prompts. The process of prompt generation consists of two stages, including superpixel graph construction and graph-based prompt aggregation. In the first stage, we generate sufficient superpixels as graph nodes to avoid over-merging errors. Then we construct a superpixel graph and predict the relationship of superpixels on the graph. To improve the feature representation capability of the graph, we leverage the prior information from SAM during the process of node feature and edge feature extraction. Specifically, we employ the SAM encoder embedding for node features and the similarity between the new superpixel predictions generated by SAM for edge features. In addition, to better extract the relation of superpixels, we add the cosine similarity between the intensity histograms as an additional edge feature to increase the distinguishability between nodes. In the second stage, we generate initial prompts from each superpixel and utilize a GNN to predict the relationship of adjacent superpixels based on the graph. We then aggregate prompts of superpixels to generate the final prompts for SAM. This graph-based prompt aggregation can effectively eliminate over-segmentation errors. Finally, the aggregated prompts are fed into SAM to obtain the final segmentation, leveraging the extraordinary capabilities of SAM in handling instance boundaries to avoid instance incompleteness.

Compared to our preliminary work [18] that employs a superpixel graph for instance segmentation without SAM, this work makes the following contributions:

- We propose a novel biological instance segmentation framework based on SAM. To address challenges encountered by SAM in biological images that exhibit dense instance distributions and diverse morphologies, we generate SAM prompts based on the superpixel graph to address the over-segmentation, over-merging, and incomplete segmentation problems.
- To improve the representation capability of the GNN, we leverage the prior information from SAM by utilizing the SAM encoder embedding and the SAM-assisted superpixel similarity. In addition, we add the cosine similarity between the intensity histograms as an additional edge feature to increase the distinguishability between nodes.
- We compare our method with various advanced methods on four representative biological datasets. The experimental results demonstrate the superior performance of our method for instance segmentation in biological images.

II. Related Work

A. Instance Segmentation

Instance segmentation in computer vision can be divided into proposal-based and proposal-free methods. Proposal-based methods [6], [19], [20], such as the Mask R-CNN [6], predict bounding boxes for each object and then compute the segmentation mask in each detected bounding box. Proposal-free methods leverage the semantic segmentation framework to group pixel-level semantic labels into distinct instances. For example, YOLACT [21] divides the segmentation pipeline into two concurrent branches, one for semantic segmentation and the other for object detection, and obtains the instance masks through these separate branches. SOLO [22] partitions the input image into various grids and computes the category probabilities directly on the convolutional feature map. Later on, many approaches [23], [24] have been proposed in the area of instance segmentation in natural images.

B. Biological Instance Segmentation

Biological instance segmentation follows the development of general instance segmentation. Related methods can also be divided into two types: proposal-based methods and proposal-free methods. Proposal-based methods [6], [7], [8] utilize object detection to find different instances and then predict masks within the region of interest. The effectiveness of these methods greatly relies on the quality of the detected bounding boxes, which often fail to distinguish overlapping instances. Additionally, when instances, such as neurons in Electron Microscopy (EM) images, exceed the receptive field of the detector, it is difficult to obtain complete segmentation using proposal-based methods. Proposal-free methods in biological image segmentation mainly consist of two types: embedding-based and affinity-based. Embedding-based methods [25], [26] typically employ an image encoder to generate high-dimensional embeddings and utilize clustering post-processing techniques (e.g. Mean-Shift algorithm [11]) to cluster these embeddings into instances. Compared to extracting pixel-wise feature embeddings, affinity-based methods [10], [12], [18], [27], [28] focus on creating an affinity map, emphasizing region boundaries. Some post-processing algorithms, such as the watershed [29] or graph cut [30], are applied on the affinity map to generate separate instances. However, the post-processing techniques

employed in the proposal-free methods are non-learnable and hand-crafted, requiring precise manual control of hyperparameters and demonstrating significant sensitivity to hyperparameter settings. Therefore, proposed-free methods tend to obtain inaccurate instance boundaries. In comparison, our method makes full use of prior information from the large model SAM and leverages its robust boundary processing and generalization capabilities.

C. Foundation Models

Large foundation models have made breakthrough progress in the field of natural language processing, demonstrating strong generalization capability on new tasks and datasets. In the computer vision field, SAM [13] becomes a commonly used foundation model for image segmentation. SAM exhibits impressive capability of generalization by using the dataset that contains eleven million images and over one billion masks [13]. However, SAM is widely employed as a semi-automatic segmentation model that requires prompts (such as points, bounding boxes, or masks), thus having difficulty promoting automatic segmentation. Subsequently, FastSAM [31] achieves automatic segmentation with SAM by incorporating a CNN-based detector and an instance segmentation branch. RSPrompter [32] enables SAM to perform automatic segmentation by training a prompt generator related to the target instances without requiring manual input. In this paper, we generate accurate superpixel prompts using graph-based aggregation while integrating SAM prior knowledge into the graph to achieve efficient automatic instance segmentation with SAM for biological images.

D. SAM-Based Medical Image Segmentation

Many research works utilize SAM for medical image analysis. Several works [33], [34], [35] evaluate the SAM's performance across various medical image segmentation tasks and modes using manual prompts. The evaluations show that the performance of SAM varies across different datasets and degrades particularly in low-contrast medical images with irregular instance shapes and weak boundaries. As a result, many works attempt to adapt SAM to medical images. Some approaches [36], [37] fine-tune the encoder and decoder of SAM but require significant computational resources and substantial training time. Other methods [38], [39] abandon or redesign the prompt encoder or mask decoder of SAM, but these approaches require manual prompts, such as points, bounding boxes, or masks, which can not achieve automated segmentation perception. In this paper, we focus on automatically providing accurate prompts to SAM by leveraging the pre-trained SAM in several stages for efficient end-to-end biological segmentation without expensive training.

III. Methodology

A. Overview

As a promptable segmentation pipeline, SAM needs high-quality prompts to produce accurate instance segmentation. Our BioSAM model generates more effective prompts instead of the regular grid points for SAM, by generating the superpixel-guided prompt. Fig. 2 shows the overall framework of our BioSAM. In the superpixel-guided prompt generation process, we construct a graph by generating sufficient superpixels as graph nodes. Then we generate prompts from each superpixel and utilize a Graph Neural Network (GNN) to predict the relationship between superpixels. The prompts of superpixels are aggregated by the results of the GNN. The aggregated prompts are then encoded as the SAM prompts to produce the final instance segmentations using the SAM models with frozen parameters.

B. Preliminaries of Segment Anything Models

SAM has significantly improved the image segmentation field, by proposing a new image segmentation pipeline named promptable segmentation. Specifically, SAM comprises three key components: (1) an image encoder that generates image embeddings from the input image; (2) a prompt encoder that generates prompt embeddings from prompts that can be automatically generated or manually provided by users; (3) a mask decoder that generates the segmentation mask from the image and prompt embeddings.

Given an input image $I \in R^{W\times H\times 3}$ and a set of prompts P, the image encoder E_{Image} and the prompt encoder E_{Prompt} transform I and P into two embeddings:$\mathbf{f}^I = E_{Image}(I)$ and $\mathbf{f}^P = E_{Prompt}(P)$. The mask decoder D predicts the segmentation mask from the embeddings $\mathbf{f}^I$ and $\mathbf{f}^P$:

$$(Z, \mathbf{c}) = D\left(\mathbf{f}^I \mid \mathbf{f}^P\right) \tag{1}$$

where $Z \in \{0,1\}^{W\times H\times 3}$ represents three predicted binary segmentation masks in three scales with different predicted confidence scores of $\mathbf{c} \in [0,1]^3$.

SAM supports three types of prompts, including point $p_{point} = (x, y)$, box defined with two diagonal corners $p_{box} = (x_{\min}, y_{\min}), (x_{\max}, y_{\max}))$, and mask prompt p_{mask} defined as binary matrix $\in \{0,1\}^{W\times H}$. The automatic SAM model initially samples points on a regular grid across the input image and segments everything at each point [13]. For each point prompt, SAM predicts a set of masks with different confidence scores that correspond to a number of valid objects. Then SAM selects masks with the highest predicted confidence score and utilizes non-maximal suppression (NMS) to filter out overlapping masks generated from different point prompts. However, since SAM is trained on large-scale natural image datasets, it fails to yield satisfactory results when directly applied to biological images with similar instance appearance textures, complex morphologies, and dense distributions.

C. Superpixel-Guided Prompt Generation

We propose to generate superpixel prompts for SAM instead of points sampled on a grid. We follow a graph-based instance segmentation approach [18] to produce the graph. We adopt its main architecture to generate a set of superpixels and utilize the GNN to predict the relationship of superpixels. Then we generate prompts from each superpixel and aggregate the prompts of

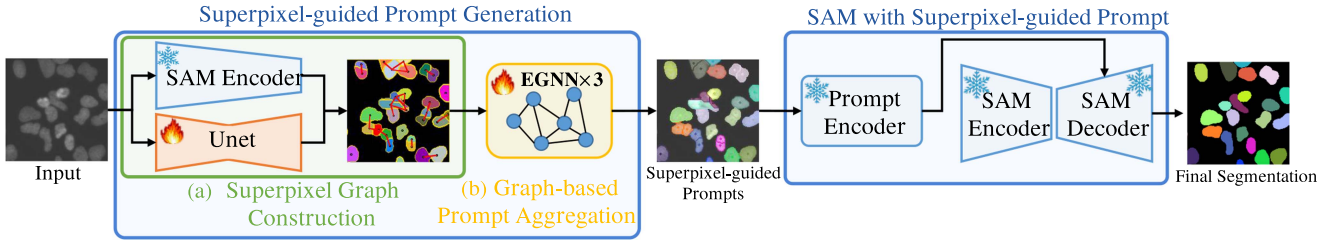

Fig. 2. The framework of our BioSAM for biological instance segmentation. BioSAM first generates superpixel-guided prompts and utilizes SAM with the prompts to generate the final segmentation. The Superpixel-guided Prompt Generation step consists of two stages: (a) Superpixel Graph Construction, which generates sufficient superpixels as graph nodes and extracts node and edge features to establish an initialized graph, and (b) Graph-based Prompt Aggregation which generates prompts from each superpixel and aggregates them to generate superpixel prompts using GNN by predicting the relationship of superpixels. Finally, we utilize SAM with the superpixel-guided prompts to obtain the final segmentation.

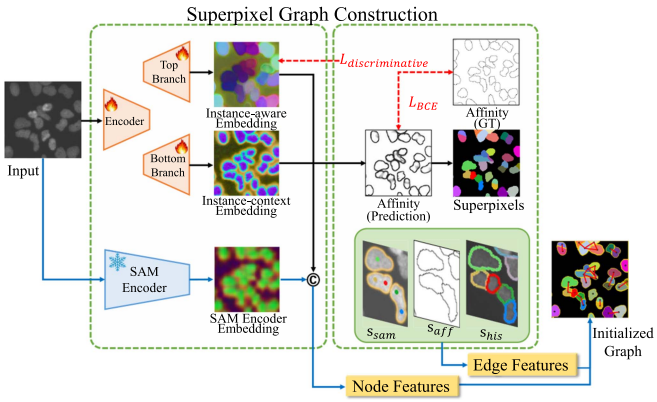

Fig. 3. Superpixel-guided graph construction. We first generate sufficient superpixels and then transform them into a graph by viewing each superpixel as a node and adding edges among all adjacent superpixels.

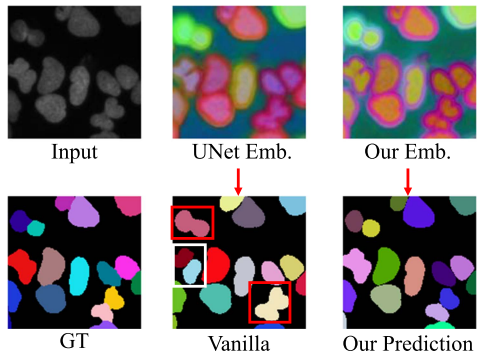

Fig. 4. Visualization of different node features and the corresponding segmentation results. 'Vanilla' represents the segmentation result using the node features extracted from UNet. 'Our Emb.' integrates the UNet embeddings with SAM embeddings as node features for segmentation. Over-merging and over-segmentation errors are highlighted in red and white boxes, respectively.

superpixels by the result of GNN. The aggregated superpixels are considered final prompts for SAM.

1) Superpixel-Guided Graph Construction: Given the input image, we first generate a set of superpixels $\{S_i|i=1,\ldots,K\}$, where K is the number of all the generated superpixels. Specifically, an affinity map is generated from one (bottom-branch) decoder and then converted to superpixels using the seeded watershed transformation algorithm [29]. An undirected graph $G=(V,E)$ is constructed from these superpixels. The node set $V=\{v_i\}$ containing N nodes, where each superpixel S_i is treated as a node v_i. The edge set $E=\{e_{ij}\}$ contains all pairs of first-order neighboring superpixels (S_i,S_j).

Node feature: The node features in [18] are extracted from the lightweight dual-branch model, resulting in insufficient representational capacity for biomedical targets with complex morphologies. To take advantage of the powerful large model SAM, we add SAM encoder embeddings as part of our node features, to capture more information of instance boundaries and contexts within the image, as Fig. 3 shows. Our node features are composed of three image embeddings. First, the SAM embeddings are extracted by a SAM encoder with frozen parameters. Secondly, we utilize the bottom branch of the trainable UNet to predict an affinity map, and then extract the penultimate layer's embeddings of the network as the instance-context embeddings to enhance the boundary information of each instance. Thirdly, following [25], the instance-aware embeddings are generated from the top branch of the UNet.

The node features extracted from the UNet only are insufficient for biomedical targets with complex morphologies and dense distribution, leading to issues such as over-segmentation or over-merging. We visualize the node features adopting principal component analysis for feature dimension reduction in Fig. 4. As it shows, our enhanced node features effectively address these problems by incorporating the embeddings from the large model SAM.

Edge feature: The edge feature of each edge e_{ij} used in [18] only comes from the affinity map. Specifically, a similarity value $s_{aff}(S_i,S_j)$ of e_{ij} is computed as the average affinities of the pixels along the overlapping boundary of two adjacent superpixels. This paradigm leads to inadequate judgment capabilities regarding the relationships between nodes in the graph. To introduce more comprehensive information about the internals of instances, we add two additional edge features, as Fig. 3 shows. First, we add the cosine similarity $s_{his}(S_i,S_j)$ between the intensity histograms of adjacent superpixels as a feature of edge e_{ij} to enhance the ability of the model to recognize differences in texture and patterns within instances. Specifically, we calculate the intensity histogram of each superpixel region in the original image and compute the cosine similarity between the intensity histograms of adjacent superpixels. Furthermore, to introduce information on the spatial and structural relationships between adjacent superpixels, we add Intersection over Union (IoU) between the new predictions from SAM as another edge feature.

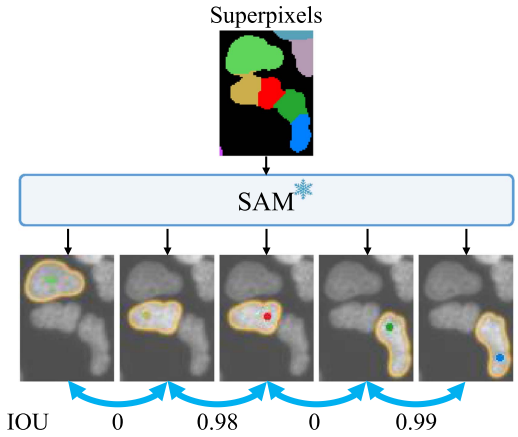

Fig. 5. The SAM-assisted superpixel similarity s_{sam}. Each superpixel is sent into SAM to predict a new segmentation mask. Then we compute the IoU among the SAM segmentation masks of adjacent superpixels. It can be seen that a pair of superpixels from the same instance can obtain a high s_{sam}, and vice versa.

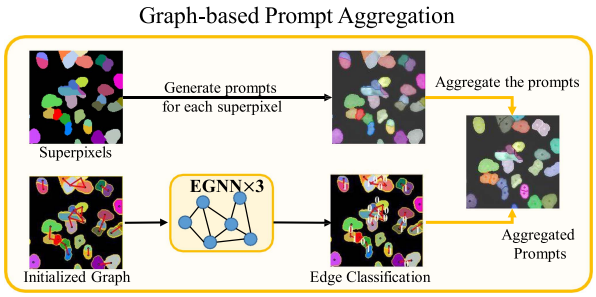

Fig. 6. Graph-based prompt aggregation. We generate prompts for each superpixel and utilize a GNN that consists of three EGNN layers to predict the relationship of superpixels. Then, we aggregate superpixel prompts according to the prediction results of the GNN.

As Fig. 5 shows, each superpixel S_i is sent into the SAM model as an individual prompt to predict a new segmentation map. Then we compute the IoU among two SAM segmentation masks from the pair of adjacent superpixels (S_i, S_j) as $s_{sam}(S_i, S_j)$. As a result, each edge e_{ij} carries three values s_{aff}, s_{his} and s_{sam}, as the edge feature for the further graph-based prompt aggregation.

2) Graph-Based Prompt Aggregation: We propose graph-based prompt aggregation to generate efficient prompts for SAM. As shown in Fig. 6, we first generate prompts from each superpixel and utilize a GNN to predict the relationship of superpixels. Then, we generate prompts by aggregating prompts of superpixels based on the results of the GNN.

Prompt of superpixel generation: For each superpixel S_i, we generate prompts P_i, including a mask prompt and several point prompts, without the use of boxes since the mask inherently includes the box information. The mask prompt p_{mask} directly comes from the superpixel segmentation results. For the point prompts, directly using the average coordinates of all points inside each superpixel cannot veritably depict the location and morphology information due to the complex shapes of biological instances. Therefore, we specially design a point prompt extraction approach. As shown in Fig. 7, we first crop out the superpixel region and apply the Euclidean distance transform. Then, for each superpixel S_i, we extract n_i point prompts by detecting n_i local peaks in the distance map while ensuring that the peaks are at least three pixels apart in the neighborhood. Finally, for each superpixel S_i, we generate prompts $P_i = \{p_{mask}, \{p^j_{point}\}^{n_i}_{j=1}\}$, where n_i represents the number of point prompts.

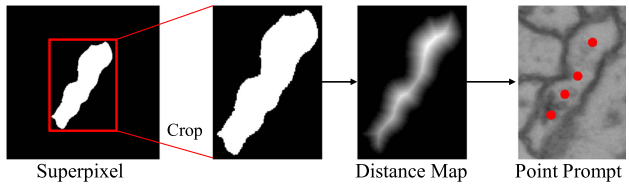

Fig. 7. Point prompt generation. We crop the superpixel and compute the Euclidean distance transform map. Then we extract point prompts by detecting local peaks in the distance map.

Prompt Aggregation: To obtain the final prompts, we determine whether the adjacent prompts of superpixels need to be aggregated by exploring the relationships of adjacent superpixels. With the improved node feature and edge features on the graph G in the superpixel-guided graph construction stage, we follow [18] to predict the relationship of superpixels on the graph G by exploiting the contextual information across all superpixels on the graph. It can be considered as a binary classification of each edge of adjacent superpixels as connected or split. A GNN is utilized to predict the relationship of superpixels through the prediction of node relationships, as Fig. 6 illustrates.

Following [40], the GNN contains three EGNN layers. Each EGNN layer is composed of two blocks for updating the node features and edge features, respectively. Note that, according to the method proposed in [40], normalization is applied for edge features, ensuring that every edge score falls within [0, 1]. Following [18], two loss functions are specially designed to supervise the GNN, including a repulsion-attraction (RA) loss to enhance the discrimination of relationships among nodes representing superpixels within the feature space, and a maximin agglomeration score loss to encourage edge classifiers by focusing more on the classification accuracy of the maximin edges.

For each edge, we compute the score $F_{gnn}(e_{ij})$ and aggregate two superpixel prompts P_i and P_j if $F_{gnn}(e_{ij}) > 0.5$. Finally, we obtain the prompts $\{P_t \mid t = 1 \dots T\}$, where T represents the number of the aggregated prompts.

D. SAM With Superpixel Prompt

Finally, for the input image I, we utilize each aggregated prompt P_t in SAM to obtain the final segmentation:

$$\begin{aligned} \mathbf{f}_t^I &= E_{Image}(I), \quad \mathbf{f}_t^P = E_{Prompt}(P_t), \\ (Z_t, \mathbf{c}_t) &= D\left(\mathbf{f}_t^I \mid \mathbf{f}_t^P\right), \end{aligned} \tag{2}$$

where E_{Image}, E_{Prompt}, and D are the image encoder, the prompt encoder, and the mask decoder of SAM, respectively. $\mathbf{f}_t^I$ and $\mathbf{f}_t^P$ represent the output embedding from the image encoder and the prompt encoder, respectively. $Z_t \in \{0,1\}^{W\times H\times 3}$ represents three predicted binary segmentation masks with different predicted confidence scores of $\mathbf{c}_t \in [0,1]^3$. We select the

mask with the highest predicted confidence score as our final segmentation.

IV. Experimental Results

To validate the effectiveness and generalization of our method, we make comparisons with the state-of-the-art methods on the AC3/AC4, CREMI, BBBC039V1, and HEK293T datasets. Then we conduct extensive ablation studies to demonstrate the importance of each key component of our method.

A. Datasets and Metrics

1) AC3/AC4: The AC3/AC4 dataset, two subsets of the mouse somatosensory cortex dataset of Kasthuri [41], is a common electron microscopy (EM) dataset for instance segmentation. The datasets contain 256 and 100 sequential EM images respectively. Each image is in size of 1024×1024 with a resolution of $3 \times 3 \times 29$ nanometers. We use the top 226 sections of AC3 as the training set, the remaining 30 sections as the validation set, and AC4 as the test set.

To assess the efficacy of segmentation in electron microscopy images, we employ two metrics by following existing work for EM instance segmentation: the Adapted Rand Error ($ARAND$) [42] and the Variation of Information (VOI) [43]. In our analysis, we calculate the two metrics for both split and merging errors. The $ARAND$ metric is from a normalized adaptation of the $RAND$ score [44]:

$$ARAND = 1 - RAND = 1 - \frac{2R_{split}R_{merging}}{R_{split} + R_{merging}}, \quad (3)$$

where R_{split} and $R_{merging}$ represent the $RAND$ scores for instance split and merging, respectively.

The metric VOI is the cumulative conditional entropies calculated between the split errors and merging errors:

$$VOI = VOI_{split} + VOI_{merging} = C(P|G) + C(G|P), \quad (4)$$

where C is the conditional entropy, P is the prediction, and G is the ground truth. Smaller $ARAND$ and VOI values represent better segmentation results.

2) Cremi: The CREMI dataset [49], imaged from the Drosophila brain at a resolution of $4 \times 4 \times 40$ nm, is another EM dataset used for 3D neuron segmentation. It consists of three sub-volumes (CREMIA, CREMIB, and CREMIC) corresponding to different neuron types. Each sub-volume consists of 125 consecutive images. We utilize the top 50 images for testing, the middle 60 images for training, and the bottom 15 images for validation. We adopt the same quantitative metrics ($ARAND$ and VOI) used for the AC3/AC4 to evaluate the results on the CREMI dataset.

3) BBBC039V1: BBC039V1 [50] is a fluorescence microscopy dataset which consists of 200 images (520×696). The instances in each image are the U2OS cells with different shapes and densities. Following [18], we adopt 100 images for the training set, 50 images for the validation set, and the remaining 50 images for the test set.

For the quantitative assessment on BBBC039V1, we utilize four metrics (AJI, $Dice$, $F1$, and PQ) commonly applied to evaluate cell segmentation. Aggregated Jaccard Index (AJI) [51] offers a comprehensive measure of segmentation overlap. It measures the agreement between the segmentation and the ground truth while accounting for the impact of both false positives and false negatives. Pixel-level Dice score ($Dice$) measures the similarity between the segmentation and the ground truth at the pixel level. The Dice score is a direct indicator of how well the segmentation results align with the actual cell boundaries. Object-level F1 score ($F1$) specifically targets the precision and recall at the level of individual cells, considering both false positives and false negatives. Panoptic Quality (PQ) [52] metric synthesizes the challenges of both instance and semantic segmentation. It evaluates the segmentation results by measuring the number of correctly identified instances and the accuracy of their semantic labels.

4) HEK293T: The HEK293T dataset [3] is a collection of confocal microscopy images of HEK293T cells, including 145 images in size of 1200×1200. 108 images are selected for training and the remaining 37 images are utilized for test images. The main challenge of segmentation on this dataset arises from the low signal regions. The segmentation performance is measured by the Average Precision metrics [53], i.e., $AP50$ and $AP75$, which respectively require a segmentation mask of at least 50% and 75% Intersection over Union (IoU) with the ground truth as the true positive. In addition, to provide a comprehensive evaluation of the model's performance, the Average Precision (AP) metric is calculated, which averages precision scores across IoU thresholds ranging from 0.5 to 0.95 with a 0.05 increment.

B. Implementation Details

To enhance the training efficiency of our method, we adopt a two-phase training procedure. In the first phase, the UNet is first pre-trained for 200 epochs. Following the instance segmentation methods [18], [25], the dual-branch UNet is trained with binary cross-entropy (BCE) loss function to supervise the affinity map generation in the bottom branch and a discriminative loss to ensure that pixels from different instances are distinguishable in the feature space. In the second phase, the GNN is trained for 200 epochs. We simultaneously update the parameters of both the GNN and the top decoder branch of the UNet, while fixing the parameters of the encoder and bottom decoder branch of the UNet. For the UNet and GNN, we initiate training with learning rates of 10^{-4} and 10^{-3}, respectively, which are halved if there is no loss decrease throughout 30 epochs. During the first and second training phases, batch sizes are set to 4 and 1, respectively. We utilize the Adam optimizer, with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. All experiments are conducted on a single NVIDIA TitanXP GPU.

C. Results on AC3/AC4 and CREMI

First, for the two EM datasets, we compare our approach against two advanced proposal-based methods, i.e., MALA [27]

and LMC [12]. MALA and LMC are two-stage methods, utilizing different post-processing techniques to cluster superpixels generated from the first stage. The same UNet backbone is utilized for the first stage as is employed in our method. Additionally, we compare the results of previous work [18], naming it 'BISSG'. We also compare with the Cellpose series, including Cellpose v1 [45] that adopts diverse datasets to enable universal segmentation, Cellpose 2.0 [46] that is adapted to a wider variety of cell images by an expanded training dataset, and Cellpose3 [47] that achieves better segmentation by enhancing the image quality. Meanwhile, we compare our method with the SMILE [48], which introduces new post-processing techniques that achieve excellent outcomes in instance segmentation for histopathology.

Furthermore, we utilize the automatic SAM method with grid point prompts, termed as 'SAM-grid'. Meanwhile, to validate the effectiveness of the prompts we generated, we devise a baseline experiment named 'SAM-box', which first uses a Mask RCNN [6] to detect bounding boxes and utilizes each box as a prompt of SAM to generate masks. 'SAM-box' has been widely embraced by the community. In addition, we compare our method with the advanced SAM-based methods (FastSAM [31] and RSPrompter [32]). Specifically, FastSAM [31] enhances the SAM by incorporating a CNN-based detector and an instance segmentation branch. RSPrompter [32] train a prompt generator to improve the SAM. We also compare the intermediate results in the initial superpixel generation stage, ending with '-s'.

Table I illustrates that our method achieves superior quantitative results for VOI and ARAND on all EM datasets. (1) Our method and the 'BISSG' method outperform the proposal-based methods, demonstrating the superiority of adopting the superpixel graph, as it can better extract the morphological features of biological images with complex shapes. (2) Compared to the 'BISSG' method, our method improves the segmentation performance by utilizing SAM, due to SAM's powerful ability to extract more accurate instance boundaries. Specifically, our method improves the ARAND metric on the AC4 dataset by 12.7%, and the VOI metric on the CREMIA dataset by 4.8%. (3) Our method significantly outperforms the Cellpose methods on all test datasets. Specifically, compared with the three versions of Cellpose, our method improves the ARAND metric on the AC4 dataset by 80.6%, 78.9%, and 78.4% respectively, and the VOI metric on the CREMIA dataset by 68.1%, 65.8%, and 63.4% respectively. The inferior performance of the three versions of Cellpose indicates that general cell instance segmentation methods face challenges in achieving satisfactory results when applied to biological images with dense distributions and complex morphological structures. (4) The performance of 'SAM-grid' is inferior, indicating that directly applying grid points as prompts of SAM for biological images struggles to achieve desirable results when faced with biological images containing intricate morphological structures. (5) The performance of 'SAM-box' is inferior for the following reasons: The neuron size in EM images is larger than the receptive field of the model, resulting in the box failing to detect the whole neuron. Consequently, the detected box from 'SAM-box' is inferior. Additionally, using only the bounding box as the prompt for SAM makes it challenging to accurately depict the morphology of the neurons, which results in inferior segmentation performance. (6) Our method surpasses advanced SAM-based methods FastSAM [31] and RSPrompter [32], demonstrating the effectiveness of utilizing the powerful capabilities of SAM for biological instance segmentation.

Furthermore, checking the intermediate superpixel results in the superpixel generation stage ('VOI-s', 'ARAND-s', 'VOI-split-s', and 'VOI-merging-s'), we can see that the segmentation results of all methods are inferior at this stage. A high 'VOI-split-s' value indicates that over-segmentation errors frequently occur. Comparing the performance of each method before and after aggregation, our graph-based prompt aggregation stage achieves the largest performance gain.

We show some visual comparison in Fig. 8. Compared with the competitors, the neuron segmentation results of our method contain fewer over-merging, over-segmentation, and incomplete segmentation errors. Moreover, our method achieves more accurate boundaries due to SAM's powerful ability to extract instance boundaries. Compared to 'BISSG', feeding our generated prompts into SAM improves the segmentation performance, particularly achieving more accurate boundaries while also addressing issues of over-merging and over-segmentation. Cellpose3 fails to segment complete instances with accurate boundaries due to the complex neuron morphologies and dense distribution in the EM datasets. The SAM-based methods also exhibit inferior performance due to the intricate neuron morphology.

D. Results on BBBC039V1

We compare our method with existing methods on the BBBC039V1 dataset in Table II. Cellpose3 achieves the best results, while ours is the second-best on this dataset. It is because Cellpose3 has been trained on the **test set** of BBBC039V1, providing sufficient prior information during inference on the test set. In contrast, our method demonstrates comparable performance to Cellpose3 on the BBBC039V1 dataset even without prior information from the test set. For the BBBC039V1 dataset, 'SAM-grid' performs better than BISSG on the AJI metric but worse on the others. This is because AJI focuses more on instance-level matches than pixel-level masks. Compared to EM images that have much more complex neuron morphology, the cells in the BBBC039V1 dataset show more regular shapes and are easily detected. The 'SAM-grid' method effectively detects all instances by sampling point prompts on a regular grid, resulting in better performance on AJI.

The corresponding visual comparison on the BBBC039V1 dataset is shown in Fig. 9. Our method generates fewer over-merging and over-segmentation errors. Additionally, our method shows superior performance at the instance boundaries. In comparison, Cellpose3 suffers from over-segmentation errors due to its strong priors. 'SAM-grid' and 'SAM-box' methods suffer from over-segmentation errors because of their dense prompts. Our method effectively addresses this issue with the aid of graph-based prompt aggregation.

TABLE I

QUANTITATIVE COMPARISON ON THE REPRESENTATIVE EM DATASETS. 'VOI-S', 'ARAND-S', 'VOI-SPLIT-S', AND 'VOI-MERGING-S' REPRESENT THE INTERMEDIATE SUPERPIXEL RESULTS IN THE INITIAL SUPERPIXEL GENERATION STAGE. 'SAM-GRID' REPRESENTS THE AUTOMATIC SAM WITH GRID POINT PROMPTS. 'SAM-BOX' REPRESENTS THE AUTOMATIC SAM WITH THE BOXED DETECTED BY MASK RCNN [6] AS PROMPTS. 'CREMI-TOTAL' REPRESENTS THE MEAN SCORES OF THREE SUB-VOLUMES ON THE CREMI DATASET. BOLD INDICATES THE BEST RESULTS

Dataset	Method	VOI-split-s ↓	VOI-merging-s ↓	VOI-s ↓	ARAND-s ↓	VOI-split ↓	VOI-merging ↓	VOI ↓	ARAND ↓
AC3/AC4	Cellpose v1 [46]	-	-	-	-	0.9325	3.3593	4.2918	0.9265
	Cellpose 2.0 [47]	-	-	-	-	0.8012	2.8600	3.6612	0.8533
	Cellpose3 [48]	-	-	-	-	0.8953	2.3376	3.2329	0.8302
	SMILE [49]	-	-	-	-	0.5922	0.3022	0.8944	0.2498
	MALA [28]	1.3278	0.1482	1.4759	0.4392	0.5894	0.2836	0.8730	0.2222
	LMC [12]	1.4020	0.1378	1.5398	0.4749	0.6005	0.2964	0.8969	0.2397
	BISSG [18]	1.4126	0.1384	1.5509	0.4790	0.5877	0.2304	0.8181	0.2059
	SAM-grid [13]	-	-	-	-	0.8532	2.0985	2.9518	0.8214
	SAM-box	-	-	-	-	0.4976	2.7529	3.2505	0.6800
	FastSAM [32]	-	-	-	-	0.8244	2.0023	2.8267	0.8198
	RSPrompter [33]	-	-	-	-	0.4918	2.3243	2.8161	0.6698
	Ours	1.4132	0.1381	1.5513	0.4799	0.5509	0.2290	**0.7799**	**0.1797**
CREMIA	Cellpose v1 [46]	-	-	-	-	0.9678	0.7825	1.7503	0.6063
	Cellpose 2.0 [47]	-	-	-	-	0.8843	0.7503	1.6347	0.5507
	Cellpose3 [48]	-	-	-	-	0.8620	0.6659	1.5279	0.5005
	SMILE [49]	-	-	-	-	0.3889	0.2421	0.6310	0.1529
	MALA [28]	0.7617	0.1464	0.9081	0.2518	0.4019	0.2020	0.6039	0.1418
	LMC [12]	0.7417	0.1436	0.8853	0.2322	0.3951	0.2219	0.6170	0.1418
	BISSG [18]	0.7864	0.1425	0.9289	0.2619	0.3823	0.2051	0.5874	0.1421
	SAM-grid [13]	-	-	-	-	0.9945	1.3490	2.3435	0.7603
	SAM-box	-	-	-	-	0.8571	1.1472	2.0043	0.7293
	FastSAM [32]	-	-	-	-	0.9965	1.3201	2.3166	0.7508
	RSPrompter [33]	-	-	-	-	0.8542	1.1212	1.9754	0.7012
	Ours	0.7865	0.1421	0.9286	0.2622	0.3603	0.1988	**0.5591**	**0.1188**
CREMIB	Cellpose v1 [46]	-	-	-	-	2.3331	1.7618	4.0949	0.8207
	Cellpose 2.0 [47]	-	-	-	-	1.9692	1.5855	3.5547	0.7619
	Cellpose3 [48]	-	-	-	-	1.8463	1.1976	3.0439	0.7008
	SMILE [49]	-	-	-	-	0.5793	0.6211	1.2004	0.2103
	MALA [28]	1.6236	0.2658	1.8894	0.4801	0.5001	0.5762	1.0763	0.1981
	LMC [12]	1.6244	0.3115	1.9359	0.5104	0.4760	0.6521	1.1281	0.2059
	BISSG [18]	1.6551	0.2004	1.8555	0.5025	0.5216	0.5542	1.0758	0.1952
	SAM-grid [13]	-	-	-	-	1.0942	2.1155	3.2097	0.7711
	SAM-box	-	-	-	-	1.7741	1.1635	2.9377	0.6317
	FastSAM [32]	-	-	-	-	1.0142	2.0122	3.0264	0.7423
	RSPrompter [33]	-	-	-	-	1.7501	1.0912	2.8413	0.6123
	Ours	1.6520	0.2003	1.8523	0.5020	0.5011	0.5386	**1.0397**	**0.1785**
CREMIC	Cellpose v1 [46]	-	-	-	-	1.8292	1.3659	3.1951	0.6792
	Cellpose 2.0 [47]	-	-	-	-	1.6828	1.1724	2.8552	0.5771
	Cellpose3 [48]	-	-	-	-	1.4377	1.0250	2.4628	0.5590
	SMILE [49]	-	-	-	-	0.8232	0.3201	1.1433	0.2594
	MALA [28]	1.9459	0.1351	2.0810	0.6095	0.7794	0.3506	1.1301	0.2371
	LMC [12]	1.7616	0.1263	1.8879	0.5206	0.8429	0.2926	1.1355	0.2342
	BISSG [18]	2.0152	0.1147	2.1299	0.6002	0.8063	0.2728	1.0790	0.2227
	SAM-grid [13]	-	-	-	-	1.3523	1.8127	3.1650	0.7697
	SAM-box	-	-	-	-	1.8278	1.4520	3.2798	0.6928
	FastSAM [32]	-	-	-	-	1.3402	1.6242	2.9644	0.7402
	RSPrompter [33]	-	-	-	-	1.7723	1.0240	2.7963	0.6823
	Ours	2.0200	0.1110	2.1310	0.6001	0.7802	0.2688	**1.0490**	**0.1968**
CREMI-Total	Cellpose v1 [46]	-	-	-	-	1.7100	1.3034	3.0134	0.7020
	Cellpose 2.0 [47]	-	-	-	-	1.5121	1.1694	2.6815	0.6299
	Cellpose3 [48]	-	-	-	-	1.3820	0.9629	2.3449	0.5868
	SMILE [49]	-	-	-	-	0.5971	0.3944	0.9916	0.2075
	MALA [28]	1.4437	0.1824	1.6262	0.4471	0.5605	0.3763	0.9368	0.1922
	LMC [12]	1.3759	0.1938	1.5697	0.4211	0.5713	0.3889	0.9602	0.1940
	BISSG [18]	1.4856	0.1525	1.6381	0.4549	0.5701	0.3440	0.9141	0.1867
	SAM-grid [13]	-	-	-	-	1.1470	1.7591	2.9061	0.7670
	SAM-box	-	-	-	-	1.4863	1.2542	2.7406	0.6846
	FastSAM [32]	-	-	-	-	1.1170	1.6522	2.7691	0.7444
	RSPrompter [33]	-	-	-	-	1.4589	1.0788	2.5377	0.6653
	Ours	1.4862	0.1511	1.6373	0.4547	0.5472	0.3354	**0.8826**	**0.1647**

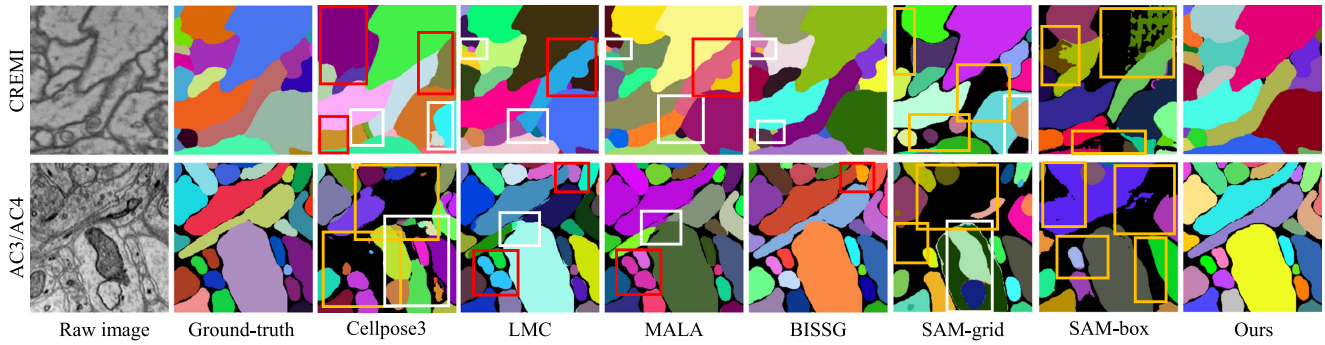

Fig. 8. Visual results on the CREMIB and AC3/AC4 datasets. Over-merging (red boxes), over-segmentation (white boxes), and incomplete segmentation (yellow boxes) are highlighted.

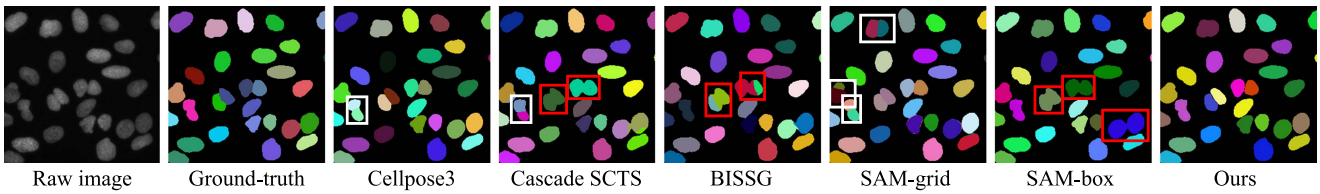

Fig. 9. Visual results on the BBBC039V1 dataset. Over-merging (red boxes) and over-segmentation (white boxes) are highlighted.

TABLE II
QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE TEST SET OF BBBC039V1

Method	AJI ↑	Dice ↑	F1 ↑	PQ ↑
Cellpose v1 [46]	0.7954	0.9489	0.9105	0.7765
Cellpose 2.0 [47]	0.8687	0.9502	0.9482	0.8538
Cellpose3 [48]	**0.8991**	**0.9599**	**0.9701**	**0.8901**
SMILE [49]	0.8702	0.9480	0.9589	0.8501
Mask RCNN [6]	0.7983	0.9277	0.9180	0.7773
Cell RCNN [8]	0.8070	0.9290	0.9276	0.7959
UPSNetN [55]	0.8128	0.9274	0.9191	0.7857
JSISNet [56]	0.8134	0.9316	0.9282	0.7913
PanFPN [53]	0.8193	0.9320	0.9275	0.7960
OANet [57]	0.8198	0.9372	0.9330	0.8085
AUNet [58]	0.8252	0.9377	0.9315	0.8090
Cell RCNNv2 [7]	0.8260	0.9336	0.9328	0.8010
PFFNet [59]	0.8477	0.9478	0.9451	0.8331
BISSG [18]	0.8680	0.9482	<u>0.9670</u>	0.8629
SAM-grid	0.8782	0.8310	0.9636	0.7585
SAM-box	0.8187	0.9322	0.9299	0.7845
FastSAM [32]	0.8213	0.8423	0.9601	0.7698
RSPrompter [33]	0.8799	0.9402	0.9323	0.7932
Ours	<u>0.8896</u>	<u>0.9514</u>	0.9639	<u>0.8751</u>

Bold indicates the best results, and underline denotes the second-best results.

TABLE III
QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE TEST SET OF HEK293T

Method	AP ↑	AP50 ↑	AP75 ↑
Cellpose v1 [46]	45.7	81.1	46.6
Cellpose 2.0 [47]	46.3	83.4	50.1
Cellpose3 [48]	48.2	84.5	51.1
SMILE [49]	50.1	85.5	56.2
Mask RCNN w/ ResNet-50 [6]	45.3	81.0	47.2
Mask RCNN w/ ResNeST-50 [60]	46.5	81.2	49.4
PointRend [19]	46.6	82.1	50.5
Mask RCNN w/ Swin-Tiny [61]	48.3	84.2	51.0
Cascade Mask R-CNN [20]	47.0	80.5	51.4
Hybrid Task Cascade [62]	48.7	83.5	51.3
MViTv2 [63]	50.0	85.2	54.0
SCTS [3]	51.6	85.9	57.6
Cascade SCTS [3]	52.3	86.1	57.4
BISSG [18]	<u>53.2</u>	<u>87.1</u>	<u>58.8</u>
SAM-grid	31.8	64.4	33.8
SAM-box	46.1	83.2	48.0
FastSAM [32]	35.1	69.0	37.1
RSPrompter [33]	48.8	84.5	48.8
Ours	**55.5**	**89.2**	**60.9**

Bold indicates the best results, and underline denotes the second-best results.

E. Results on HEK293T

Our method also shows superiority on the challenging HEK293T dataset. As Table III shows, our method achieves the best performance across all the evaluated metrics, i.e., 4.3% AP and 3.6% $AP75$ performance gain compared with the second-best method. In addition, our method outperforms the best version of Cellpose, achieving a 19.2% improvement in AP75. This demonstrates the superiority of our method compared with the general cell instance segmentation method in dealing with biological images that contain instances with complex morphological structures and uneven distributions. We visualize the segmentation results of our method and five other methods (Cellpose3 [47], Cascade SCTS [3], BISSG [18], SAM-grid [13], and SAM-box) for comparison in Fig. 10. In comparison with Cellpose3 which usually generates over-segmentation when

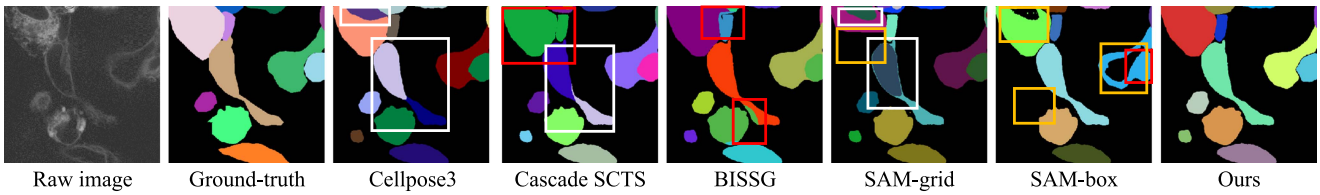

Fig. 10. Visual results on the HEK293T dataset. Over-merging (red boxes), over-segmentation (white boxes), and incomplete segmentation (yellow boxes) are highlighted.

TABLE IV
ABLATION STUDY ON DIFFERENT PARTS OF NODE FEATURES. 'EMB.' STANDS FOR 'EMBEDDING'

SAM Emb.	Instance-context Emb.	Instance-aware Emb.	VOI ↓	ARAND ↓
✓			1.4167	0.3036
	✓		0.8456	0.2267
		✓	0.8323	0.2143
✓	✓		0.8082	0.1912
✓		✓	0.7921	0.1923
	✓	✓	0.7887	0.1867
✓	✓	✓	**0.7799**	**0.1797**

processing instances with complex shapes, our method generated more complete instance segmentation with accurate boundaries. The 'SAM-grid' and 'SAM-box' methods suffer from handling non-regular shapes, resulting in over-segmentation, over-merging and incomplete segmentation. 'BISSG' achieves a similar instance distribution with our method, but tends to generate inaccurate instance boundaries. Taking advantage of the powerful boundary extraction capabilities of the SAM model, our method achieves more accurate boundaries.

F. Ablation Study

To verify the effectiveness of each component within our proposed framework, we conduct ablation studies mainly on the AC3/AC4 dataset which is the representative biomedical image dataset.

1) Node Features: We conduct experiments on different parts of the node feature, including SAM encoder embeddings, instance-context embeddings, and instance-aware embeddings. As shown in Table IV, the node features are complementary to each other, while the best results are achieved when all three features are used together. Among these, the instance-aware embedding from the top branch contains more effective decision-making information. We also show some segmentation results of various biomedical images in Fig. 11 using the UNet embedding only and the integrated embeddings of UNet and SAM. It can be seen that our approach effectively addresses the over-segmentation and over-merging errors that arise in various biological images with complex morphologies and dense distribution.

2) Edge Features: We conduct experiments on different parts of the edge feature, including s_{his}, s_{sam} and s_{aff}. As shown in Table V, the best performance is achieved by integrating all features. Among these, the affinity map is more

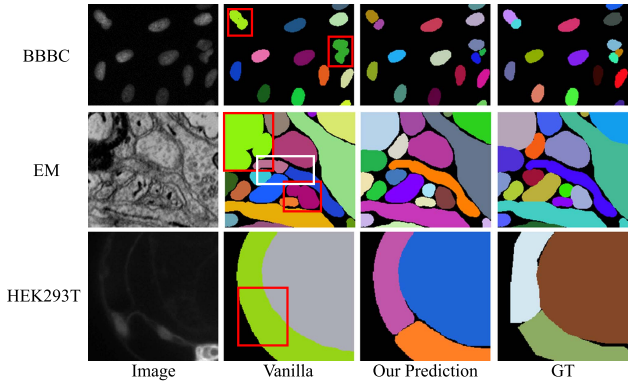

Fig. 11. Some segmentation results using different node features on three datasets that contain biological images with complex cell morphologies. 'Vanilla' represents using node features extracted from the lightweight dual-branch UNet only. 'Our Prediction' is generated by integrating the UNet embeddings with SAM embeddings as node features. Over-merging (red boxes) and over-segmentation (white boxes) are highlighted.

TABLE V
ABLATION STUDY ON DIFFERENT PARTS OF EDGE FEATURES

s_{his}	s_{sam}	s_{aff}	VOI ↓	ARAND ↓
			0.9234	0.2260
✓			0.8364	0.2035
	✓		0.8026	0.1975
		✓	0.7989	0.1896
✓	✓		0.8012	0.1889
✓		✓	0.7898	0.1844
	✓	✓	0.7825	0.1823
✓	✓	✓	**0.7799**	**0.1797**

conducive to the estimation of superpixel relationships on the graph. Based on the affinity map, the segmentation performance is further enhanced by our proposed SAM-assisted superpixel similarity s_{sam} and the cosine similarity between the intensity histograms s_{his}.

3) Effect of Improved Mechanisms: Compared to the graph-based superpixel agglomeration approach [18], we improve the node feature and edge feature for superpixel agglomeration and apply SAM in a novel biological instance segmentation framework. We conduct ablation experiments on the aforementioned improvements. As Table VI shows, each improved mechanism contributes to the segmentation performance.

To validate the effectiveness of the prompt we generated, we devise a variant, named 'BISSG w/ SAM', that leverages the instance result generated in previous work [18] directly as the SAM

TABLE VI
ABLATION STUDY ON OUR IMPROVED MECHANISMS COMPARED TO OUR PRELIMINARY WORK [18]. 'IN' MEANS THE IMPROVED NODE FEATURE. 'IE' MEANS THE IMPROVED EDGE FEATURE

Method	IN	IE	SAM	VOI ↓	ARAND ↓
BISSG [18]				0.8181	0.2059
BISSG [18] w/ SAM			✓	0.8101	0.2012
Ours-v1	✓			0.8066	0.1955
Ours-v2		✓		0.8057	0.1935
Ours-v3	✓	✓		0.7956	0.1903
Ours-v4	✓		✓	0.7989	0.1896
Ours-v5		✓	✓	0.7887	0.1867
Ours-full	✓	✓	✓	**0.7799**	**0.1797**

TABLE VII
COMPARISON OF THE SEGMENTATION PERFORMANCES (VOI (↓) AND ARAND (↓)) WHEN INTEGRATING FOUR DIFFERENT SUPERPIXEL GENERATION METHODS IN OUR BIOSAM FRAMEWORK ON THE AC3/AC4 DATASET

Method	VOI ↓	ARAND ↓
BioSAM w/ LSC [64]	0.8488	0.2192
BioSAM w/ SLIC [65]	0.8411	0.2109
BioSAM w/ FCN-based [66]	0.8167	0.1989
BioSAM w/ Affinity-based [18]	**0.7799**	**0.1797**

prompts. When the initial superpixel prompt from 'BISSG' is not good enough, directly feeding them into SAM will not yield significant benefits. In comparison, our method achieves better results with the improved mechanism for prompt generation. Specifically, using only SAM encoder embedding in 'Ours-v1' greatly improves performance over 'BISSG w/ SAM', indicating the strong representation capability of the SAM encoder for segmentation. Comparing 'BISSG w/ SAM' to 'Ours-v2', enhancing edge features with SAM-assisted superpixel similarity and histogram similarity is more effective. Then, by integrating the improved node features and edge features for prompt generation and the SAM refinement with more effective prompts in the other variants (v3, v4, v5, and ours-full), our BioSAM can progressively improve the instance segmentation in biological images. This validates the advantages of our framework, which can make full use of SAM in various stages such as feature embedding, superpixel aggregation, and boundary refinement.

4) Generalization Ability: Our BioSAM segmentation framework is general and can be combined with different superpixel generation methods. To validate its generalization ability, we integrate four different superpixel generation methods, including two traditional non-learning methods (LSC [63], SLIC [64]) and two learning-based methods (FCN-based [65] and Affinity-based [18] in our BioSAM framework. We compare the VOI or ARAND metrics of their segmentation results on the AC3/AC4 dataset in Table VII. We can see that the learning-based superpixel generation methods outperform non-learning superpixel generation methods with better superpixel results as prompts. Particularly, our BioSAM framework, with either traditional superpixel generation approaches or learning-based superpixel generation approaches, outperforms most existing methods (MALA [27], LMC [12], SAM-grid [13], FastSAM [31] and RSPrompter [32]).

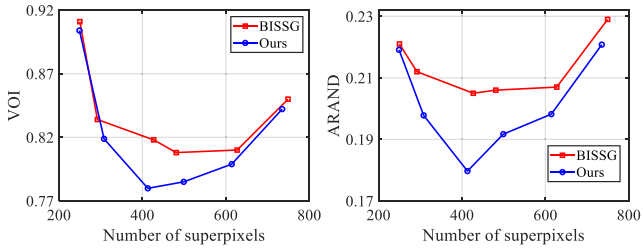

Fig. 12. Ablation study on different numbers of superpixels generated by the watershed transformation.

5) Number of Superpixels: We compare the performance of our method with 'BISSG' [18] under different numbers of superpixels. As Fig. 12 shows, our method outperforms 'BISSG' across all conditions with different numbers of superpixels. Compared to 'BISSG' which is more sensitive to the number of superpixels, our method demonstrates greater robustness across different numbers of superpixels.

V. CONCLUSION

We propose BioSAM, a new biological instance segmentation framework generating SAM prompts from a superpixel graph. The essence is to generate prompts for SAM using GNN. We first generate sufficient superpixels as graph nodes and establish an initialized graph. We leverage the prior information from SAM to improve the feature representation of the graph. We then generate initial prompts from each superpixel and aggregate them through a GNN by predicting the relationship of superpixels. Finally, we utilize the aggregated efficient prompts in SAM to obtain the final segmentation. Our proposed method outperforms state-of-the-art methods on four representative biological datasets.

REFERENCES

[1] X. Liu, B. Hu, M. Li, W. Huang, Y. Zhang, and Z. Xiong, "A soma segmentation benchmark in full adult fly brain," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7402–7411.
[2] Y. Chen, W. Huang, X. Liu, S. Deng, Q. Chen, and Z. Xiong, "Learning multiscale consistency for self-supervised electron microscopy instance segmentation," in *IEEE Int. Conf. Acoust., Speech Signal Process.*, 2024, pp. 1566–1570.
[3] Y. Zhou, W. Li, and G. Yang, "SCTS: Instance segmentation of single cells using a transformer-based semantic-aware model and space-filling augmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Jan. 2023, pp. 5944–5953.
[4] H. Wang, Z. Liu, and X. Ma, "Learning consistency and specificity of cells from single-cell multi-omic data," *IEEE J. Biomed. Health Inform.*, vol. 28, no. 5, pp. 3134–3145, May 2024.
[5] Z. Wan et al., "CellT-Net: A composite transformer method for 2-D cell instance segmentation," *IEEE J. Biomed. Health Inform.*, vol. 28, no. 2, pp. 730–741, Feb. 2024.
[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[7] D. Liu et al., "Nuclei segmentation via a deep panoptic model with semantic feature fusion," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 861–868.
[8] D. Zhang et al., "Panoptic segmentation with an end-to-end cell R-CNN for pathology image analysis," in *Proc. Med. Image Comput. Comput.-Assist. Interv.*, 2018, pp. 237–244.
[9] X. Liu et al., "Cross-dimension affinity distillation for 3D EM neuron segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 11104–11113.

[10] Y. Liu et al., "Affinity derivation and graph merge for instance segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 686–703.

[11] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 32–40, Jan. 1975.

[12] T. Beier et al., "Multicut brings automated neurite segmentation closer to human performance," *Nature Methods*, vol. 14, no. 2, pp. 101–102, 2017.

[13] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.

[14] J. Huang et al., "Learning to prompt segment anything models," 2024, *arXiv:2401.04651*.

[15] C. Zhang et al., "A survey on segment anything model (SAM): Vision foundation model meets prompt engineering," 2023, *arXiv:2306.06211*.

[16] C. Li, P. Khanduri, Y. Qiang, R. I. Sultan, I. Chetty, and D. Zhu, "Auto-prompting SAM for mobile friendly 3D medical image segmentation," 2023, *arXiv:2308.14936*.

[17] D. Cheng, Z. Qin, Z. Jiang, S. Zhang, Q. Lao, and K. Li, "SAM on medical images: A comprehensive study on three prompt modes," 2023, *arXiv:2305.00035*.

[18] X. Liu, W. Huang, Y. Zhang, and Z. Xiong, "Biological instance segmentation with a superpixel-guided graph," in *Proc. Int. Joint Conf. Artif. Intell.*, 2022, pp. 1209–1215.

[19] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2020.

[20] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162.

[21] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9157–9166.

[22] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "SOLO: Segmenting objects by locations," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 649–665.

[23] R. Li, C. He, S. Li, Y. Zhang, and L. Zhang, "DynaMask: Dynamic mask selection for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 11279–11288.

[24] F. Li et al., "Mask DINO: Towards a unified transformer-based framework for object detection and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 3041–3050.

[25] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function," 2017, *arXiv:1708.02551*.

[26] V. Kulikov and V. Lempitsky, "Instance segmentation of biological images using harmonic embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3843–3851.

[27] J. Funke et al., "Large scale image segmentation with structured loss based deep learning for connectome reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1669–1680, Jul. 2019.

[28] W. Huang, S. Deng, C. Chen, X. Fu, and Z. Xiong, "Learning to model pixel-embedded affinity for homogeneous instance segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 1007–1015.

[29] S. Beucher, "The morphological approach to segmentation: The watershed transformation," *Math. Morphol. Image Process.*, pp. 433–482, 1993.

[30] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2001, vol. 1, pp. 105–112.

[31] X. Zhao et al., "Fast segment anything," 2023, *arXiv:2306.12156*.

[32] K. Chen et al., "RSPrompter: Learning to prompt for remote sensing instance segmentation based on visual foundation model," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4701117.

[33] R. Deng et al., "Segment anything model (SAM) for digital pathology: Assess zero-shot segmentation on whole slide imaging," *Med. Imag. Deep Learn.*, 2023.

[34] S. Mohapatra, A. Gosai, and G. Schlaug, "SAM vs BET: A comparative study for brain extraction and segmentation of magnetic resonance images using deep learning," 2023, *arXiv:2304.04738*.

[35] A. Wang, M. Islam, M. Xu, Y. Zhang, and H. Ren, "SAM meets robotic surgery: An empirical study on generalization, robustness and adaptation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2023, pp. 234–244.

[36] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Commun.*, vol. 15, no. 1, 2024, Art. no. 654.

[37] Y. Liu et al., "Segment any medical model extended," *Proc. SPIE*, vol. 12926, 2024, Art. no. 129261M.

[38] S. Gong et al., "3DSAM-adapter: Holistic adaptation of SAM from 2D to 3D for promptable medical image segmentation," 2023, *arXiv:2306.13465*.

[39] J. Wu et al., "Medical SAM adapter: Adapting segment anything model for medical image segmentation," 2023, *arXiv:2304.12620*.

[40] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11–20.

[41] N. Kasthuri et al., "Saturated reconstruction of a volume of neocortex," *Cell*, vol. 162, no. 3, pp. 648–661, 2015.

[42] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.

[43] M. Meilă, "Comparing clusterings by the variation of information," in *Proc. Learn. Theory Kernel Mach.*, 2003, pp. 173–187.

[44] I. Arganda-Carreras et al., "Crowdsourcing the creation of image segmentation algorithms for connectomics," *Front. Neuroanat.*, vol. 9, 2015, Art. no. 152591.

[45] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: A generalist algorithm for cellular segmentation," *Nature Methods*, vol. 18, no. 1, pp. 100–106, 2021.

[46] M. Pachitariu and C. Stringer, "Cellpose 2.0: How to train your own model," *Nature Methods*, vol. 19, no. 12, pp. 1634–1641, 2022.

[47] C. Stringer and M. Pachitariu, "Cellpose3: One-click image restoration for improved cellular segmentation," *bioRxiv*, pp. 2024–02, 2024.

[48] X. Pan et al., "SMILE: Cost-sensitive multi-task learning for nuclear segmentation and classification with imbalanced annotations," *Med. Image Anal.*, vol. 88, 2023, Art. no. 102867.

[49] CREMI, "MICCAI challenge on circuit reconstruction from electron microscopy images," 2016. [Online]. Available: https://cremi.org/

[50] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, "Annotated high-throughput microscopy image sets for validation," *Nature Methods*, vol. 9, no. 7, pp. 637–637, 2012.

[51] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi, "A dataset and a technique for generalized nuclear segmentation for computational pathology," *IEEE Trans. Med. Imag.*, vol. 36, no. 7, pp. 1550–1560, 2017.

[52] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6399–6408.

[53] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[54] Y. Xiong et al., "UPSNet: A unified panoptic segmentation network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8818–8826.

[55] D. De Geus, P. Meletis, and G. Dubbelman, "Panoptic segmentation with a joint semantic and instance segmentation network," 2018, *arXiv:1809.02110*.

[56] H. Liu et al., "An end-to-end network for panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6172–6181.

[57] Y. Li et al., "Attention-guided unified network for panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7026–7035.

[58] D. Liu, D. Zhang, Y. Song, H. Huang, and W. Cai, "Panoptic feature fusion net: A novel instance segmentation paradigm for biomedical and biological images," *IEEE Trans. Image Process.*, vol. 30, pp. 2045–2059, 2021.

[59] H. Zhang et al., "ResNeSt: Split-attention networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2736–2746.

[60] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[61] K. Chen et al., "Hybrid task cascade for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4974–4983.

[62] Y. Li et al., "MViTv2: Improved multiscale vision transformers for classification and detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4804–4814.

[63] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1356–1363.

[64] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[65] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13964–13973.