# Weakly Supervised Cell Segmentation by Point Annotation

Tianyi Zhao, *Graduate Student Member, IEEE*, Zhaozheng Yin, *Member, IEEE*

***Abstract*— We propose weakly supervised training schemes to train end-to-end cell segmentation networks that only require a single point annotation per cell as the training label and generate a high-quality segmentation mask close to those fully supervised methods using mask annotation on cells. Three training schemes are investigated to train cell segmentation networks, using the point annotation. First, self-training is performed to learn additional information near the annotated points. Next, co-training is applied to learn more cell regions using multiple networks that supervise each other. Finally, a hybrid-training scheme is proposed to leverage the advantages of both self-training and co-training. During the training process, we propose a divergence loss to avoid the overfitting and a consistency loss to enforce the consensus among multiple co-trained networks. Furthermore, we propose weakly supervised learning with human in the loop, aiming at achieving high segmentation accuracy and annotation efficiency simultaneously. Evaluated on two benchmark datasets, our proposal achieves high-quality cell segmentation results comparable to the fully supervised methods, but with much less amount of human annotation effort.**

***Index Terms*— Cell segmentation, Weakly supervised learning, Point annotation, Neural networks, Human in the loop**

## I. Introduction

Cell segmentation masks can provide lots of information on cells [1] such as locations, shapes, and intensity distributions. The current supervised-learning-based cell segmentation algorithms such as those based on deep learning, require full cell segmentation masks as the annotation labels (e.g., Fig. 1(a)) to train the segmentation algorithms. But, annotating the segmentation masks is very time-consuming and costly for biologists or doctors. Furthermore, biomedical images exhibit great appearance variations due to different imaging modalities and specimen types [2]. It is almost infeasible to train a general segmentation algorithm and fine-tune it to all different application domains. Motivated by the promising prospect of deep-learning-based segmentation and hindered by the lack of fully annotated training data, we propose weakly supervised methods to training neural networks for cell segmentation, which only require a point annotation as the training label for each cell (Fig. 1(d)). Compared to labeling a complete mask for a cell, one point annotation for one cell, as the weakest supervision, can greatly save human annotation efforts, and we aim to achieve the segmentation performance close to those fully supervised methods using mask annotations.

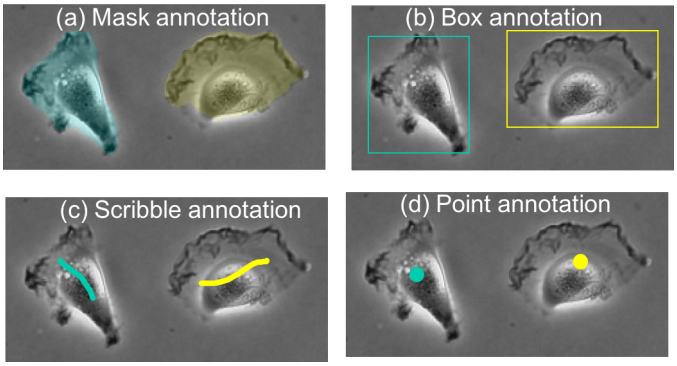


Fig. 1. Examples of fully supervised mask annotation and weakly supervised box annotation, scribble annotation and point annotation.

### A. Related Work

*1) Traditional Cell/Nuclei Segmentation:* Cell segmentation and nuclei segmentation have attracted the attention of the research community for decades [3]. Lots of cell segmentation methods were proposed before the era of deep learning, such as level sets [4], [5], graph cuts [6], [7], random walk [8], [9], correlation clustering [10], object shape prior modeling [11], and active contour [12]. Those methods are usually designed for a certain type of images and may not work well on images with large object variations. Particularly, some hand-crafted features or parameters need to be tuned for different image modalities and specimen types. Nuclei segmentation is similar as the cell segmentation but it usually has high contrast and high sparsity (e.g., benchmark datasets [13], [14]). Thresholding [15], dynamic programming [16], Bayesian classification [17], and sparsity-constrained convolutional regression [18] have been used in nuclei segmentation.

*2) Deep Learning for Cell Segmentation:* Recently, fully supervised deep learning methods have been proposed to solve the cell segmentation problem [19], [20] in phase-contrast microscopy images. UNet [21], named after its U-shape network architecture, is a representative cell segmentation method that can converge with limited training data. A pyramid based method [22], using Fully Convolutional Network [23] on

The paper was submitted on Aug 1st 2020. This project was supported by NSF CAREER award IIS-2019967 and SUNY Empire Innovation Program (EIP).

Tianyi Zhao is a PhD student in the Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA. (e-mail: zhao12@cs.stonybrook.edu).

Zhaozheng Yin is with the AI Institute, Department of Computer Science, Department of Biomedical Informatics, and Department of Applied Mathematics & Statistics (Affiliated), Stony Brook University, Stony Brook, NY 11794, USA. (e-mail: zyin@cs.stonybrook.edu).

each level of the pyramid, is developed to tackle the low contrast and irregular object boundary problem in microscopy images. Gated recurrent neural networks [24] are integrated to capture the global prior (e.g., shapes) and iteratively refine the segmentation network.

In addition to the phase-contrast microscopy images, many deep segmentation methods are proposed in other microscopy modalities such as histopathology images [25]–[28]. To segment clustered object boundaries is still a challenging problem in this task. The fully convolutional networks are used in [25] to address the problem of segmenting touching nuclei by formulating it as a regression task of the distance map. ResNet [29] is used in [26] with more instance branches to integrate global and local features. A Contour-aware Informative Aggregation Network (CIA-Net) is proposed in [27]. This network includes dense module, transition module and aggregation module to aggregate information from nuclei and contour detectors. In [28], features learned from each convolution layer are given to the sparse regression chain to detect the boundaries of the clustered objects.

In addition to biological cell images, deep segmentation networks are also widely used for different types of medical images, such as histopathological breast cancer images [30] for region segmentation, Computed Tomography (CT) images for lung segmentation [31] and airway segmentation [32], multi-phase CT scans for pancreas and pancreatic ductal adenocarcinoma segmentation [33], and electron microscopy images for neuronal membranes segmentation [34].

*3) Weakly Supervised Deep Learning Methods:* In natural image segmentation, weakly supervised segmentation algorithms have been proposed based on different kinds of weak labels. Bounding box annotation has been used in [35]–[37] for semantic segmentation networks. Scribble supervised convolutional networks are proposed in [38]. The point supervision problem is proposed in [39]. Weakly supervised segmentation algorithms for biomedical images have also been proposed based on different kinds of weak labels. For example, object detection response is propagated as the weak label to generate the segmentation mask [40]. Bounding box annotation (e.g., Fig. 1(b)) is used in [42]. Scribble annotation (e.g., Fig. 1(c)) is employed for cell segmentation in [43]. Incomplete and inaccurate annotations are used in [41]. Compared with other weak annotations (box, scribble, etc.), the incomplete and inaccurate annotations in [41] are almost the full cell masks. A contour-aware method is proposed in [41] to enhance the segmentation on the boundary.

Point annotation is used in pathology images [44] for nuclei segmentation, in immunohistochemistry cancer and tonsil tissue slides [45] for cell segmentation, in histopathological image [46]–[48] for cell segmentation, in retinal fundus images [49] for optic disc and cup segmentation, and in magnetic resonance imaging [50] for cardiac, vertebral body and prostate segmentation. In [44], [45], some unsupervised methods, such as Voronoi transformation and k-means clustering, generate sub-regions from the point annotation as pseudo labels. However, the unsupervised method works well on limited types of images with regularly distributed cells. When the cells are distributed sparsely or cells exhibit large size/shape variations, the pseudo labels generated by Voronoi transformation may not be accurate.

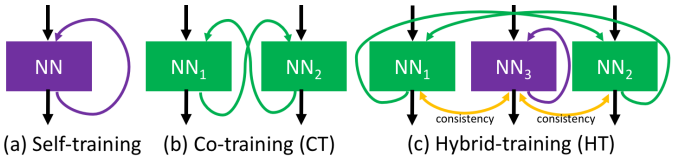


Fig. 2. Training schemes of self-training, co-training and hybrid-training. a) The self-training scheme updates the neural network by the output of itself. b) The co-training scheme has a pair of neural networks which are trained by different subsets of the data and co-trained by each other. c) The hybrid-training has three networks. Two networks are co-trained by each other and one network is self-trained. A consistency regularization is applied between the co-trained networks and the self-trained network.

To address the scarcity of annotated training data, domain adaptation is deployed in [46], [47] for efficiently transferring pretrained models, which needs some extra data and labels from the same image type or synthesized images and labels [48]. An adversarial constrained-CNN (convolutional neural network) is proposed in [49] to synthesize image labels. A constrained-CNN is proposed in [50] to leverage unlabeled data by prior knowledge.

Compared to those weakly supervised methods, we propose end-to-end schemes that can be used by different image types without any pseudo labels or extra data. We also propose new divergence loss and consistency loss to regulate the neural networks during training and alleviate the problems of overfitting and unstable training process.

*4) Learning from Weak Labels:* Learning from weak labels or enhancing weak classifiers to stronger ones has attracted a lot of attention in the community. self-training, also known as bootstrapping, is proposed in [51]. A classifier is initialized by some seed data. Then, the unlabeled data are classified by the classifier trained in the previous iteration, and are used to update the classifier. This process iterates until the convergence. Bootstrapping is also a common technique to train classifiers on the image classification task with noisy labels [52]. Co-training [53] is an extension of self-training but has a pair of classifiers that can supervise each other. Inspired by the effectiveness of these methods, we investigate how to leverage self-training and co-training to train high quality cell segmentation algorithms using weak labels. Compared to the weak labels such as box annotations and scribble annotations, point annotation is more annotation-efficient for cell segmentation tasks (it is applicable to cell tracking tasks [54] and cell counting tasks [55] too). Thus, we explore the extremely-weak supervision (one point annotation per cell), targeting at the segmentation performance close to the full-supervision (mask annotation for every cell) but with the least human annotation effort.

## B. Our Proposal

High-throughput biological experiments generate large amounts of image data over time. There are a huge set of important applications that suffer from high image heterogeneity (e.g., quantification of cell size with and without drug, relative phenotype analysis, etc.). In such situations, the images to be processed vary a lot (e.g., different cell types,

different microscopes, different resolutions, etc.). Collecting a few training images with full annotations will not make the model work on all cell types. Annotating a large training dataset with full masks for all cell types can be costly and time-consuming. In contrast, doing point annotation on a lot of images in each cell type will certainly be preferred. Each cell can be labeled by one simple mouse-click, rather than tediously drawing boundaries around cells or drawing boxes and scribbles. We propose to effectively train cell segmentation algorithms from the point annotation on various cell images, achieving annotation-efficient machine learning for cell segmentation.

In our proposed method, we design self-training, co-training and hybrid-training schemes (Fig. 2) to train cell segmentation algorithms (e.g., neural network models). Self-training is to update the segmentation mask by the current prediction of the network. Co-training has a pair of networks that update each network by the currently predicted segmentation mask of the other network. The hybrid-training method makes the co-training more consistent with additional self-training. These three schemes are suitable for training various deep-learning-based cell segmentation algorithms.

Furthermore, we introduce two new loss terms, divergence loss and consistency loss, which are suitable for the self-training, co-training, and hybrid-training. One common problem during self-training is the "self-deception" problem (i.e., the network in the current iteration is cheated by the output of the previous network). The self-training process can be unstable or be prone to be overfitted. It is hard for the self-training scheme to realize its own error, thus we propose a divergence loss to deal with this problem by monitoring the first-order differences of the network between two iterative trainings in the self-training, co-training, and hybrid-training schemes. To maintain the consensus among multiple networks and make the training process more stable, we introduce a consistency loss in the hybrid-training scheme.

Since the point annotation is an extremely-weak label, the weakly-supervised method is hard to achieve segmentation masks as precisely as the full-supervision methods. In some applications where biologists really care about the accuracy of the cell boundaries (no matter how hard it is to collect the boundary annotation), we propose a mechanism of weakly supervised learning with human in the loop, i.e., our point-annotation-based weakly supervised learning can be an initialization to train the segmentation algorithms, and then it guides the human annotators to provide extra point annotations or a small amount of full annotations to fine-tune the initially trained segmentation network. Compared with the fully-supervised methods with full annotations which are trained from scratches, the proposed mechanism can train segmentation networks to segment cells as precisely as the fully-supervised methods, but with much less human annotation efforts.

In summary, our proposed method has four-fold contributions:

- We design self-training, co-training and hybrid-training schemes to train the end-to-end cell segmentation networks from point annotations.
- We introduce two new loss terms: divergence loss and consistency loss to solve the self-deception problem and unstable training problem in the self-training, co-training, and hybrid-training.
- With our weakly supervised learning with human in the loop, high quality segmentation networks can be trained efficiently with less annotations, compared to fully supervised methods.
- Our method is evaluated on two benchmark datasets, demonstrating its values on annotation-efficient machine learning for cell segmentation.

## II. Preliminaries

In this section, we briefly summarize three recent deep-learning-based cell segmentation algorithms, and we will use them as examples to demonstrate that deep learning algorithms can be weakly trained by our proposed schemes, but still obtain high performances.

### A. Fully Convolutional Network (FCN)

The fully convolutional neural network published in 2015 [23] is a neural network that mainly contains convolutional layers, down-sampling or up-sampling layers. It generates an object segmentation mask image with the same size as the input image. If a FCN does not contain any down-sampling or up-sampling layers [22], every convolutional layer of the network has the same length and width but different numbers of channels, as shown in Fig. 3. The fully convolutional neural network does not require a fixed-size input. The objective function could be pixel-wise such as the cross-entropy or mask-wise such as the dice-coefficient.

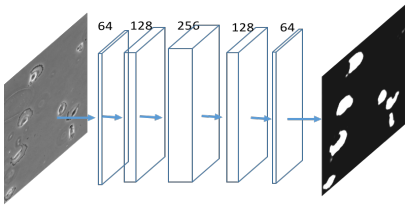


Fig. 3. A typical fully convolutional network architecture [22].

### B. U-Net

U-Net [21] was published in 2015 and has been widely used for biomedical image segmentation. It is a modified fully convolutional network, which works well with a small amount of training images on the same image type. The U-Net performs down-sampling (max-pooling) after convolutions, and then employs up-sampling with up-convolutions. Skip connections are deployed to get a more precise segmentation mask, as shown in Fig. 4.

### C. Pyramid-based Network

The pyramid-based fully convolutional network was published in 2018 [22]. A sequence of cell segmentation networks are trained in a cascaded refinement manner to tackle two challenges of cell segmentation task: (a) low contrast between cells and background; and (b) irregular shapes of cells. The

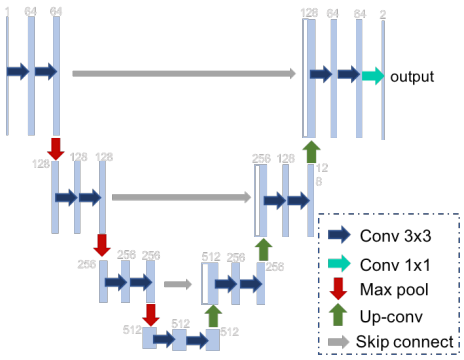


Fig. 4. A typical U-Net architecture [21].

network overview is shown in Fig. 5. The input to the series of FCNs is a Gaussian pyramid, but fusing the output from FCNs is achieved in a way similar to the sequential image reconstruction in the Laplacian pyramid. The bottom-level FCN is trained using the cross-entropy loss and generates coarse cell segmentation masks. Then, the residual error is propagated to the higher-level FCNs to add missed cell details in a cascaded way.

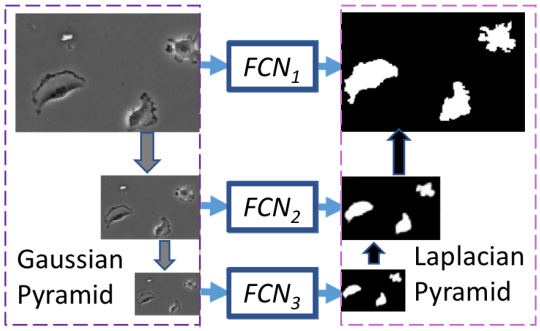


Fig. 5. A typical Pyramid-based network architecture [22].

## III. Methodology

Fig. 6 provides an overview of our training schemes for the weakly supervised cell segmentation using point annotations. Denote $X$ and $Y$ as the training image dataset and its point annotation set, respectively. The baseline method (Fig. 6(a)) is to train a Neural Network (NN) using $X$ and $Y$ directly, and obtain the trained segmentation network $F$. We describe the details of the other training schemes step-by-step in the following subsections.

### A. Self-training

Let $x \in R^{W \times H}$ denote an input image with width $W$ and height $H$. Its segmentation label is denoted as $y \in \{0,1\}^{K \times W \times H}$ where $K$ is the number of classes. In our segmentation task, $K = 2$, i.e., background pixel and cell pixel. Let $F(x) \in R^{K \times W \times H}$ denote the network's output (probability map from the last softmax regression layer). The network is firstly trained by the initial point annotation using the cross entropy (ce) loss:

$$L_{ce}(F(x), y) = -\sum_{w,h}\sum_{k}^{K} y_{k,w,h}\ log(F(x)_{k,w,h}) \quad (1)$$

The network $F$ trained by the initial point annotation is our baseline segmentation algorithm (i.e., Fig. 6(a)), and one sample result is shown in Fig. 7 (b), corresponding to the input image and its point annotation in Fig. 7(a). Only a small portion of cell regions can be learned from this baseline method.

The traditional bootstrapping [52] proposed a simple $\beta$-blending method, with $0 < \beta < 1$, to dynamically update the network by a convex combination of the training label and the current prediction of the model. The loss function can be modified from the cross-entropy loss (Eq. 1) as:

$$Loss = L_{ce}(F(x), \beta y + (1-\beta)F(x)) \quad (2)$$

However, since our point annotation is extremely-weak, this $\beta$-bending is not balanced. Thus, we propose a new self-training (ST) method to further improve the neural network by combining the current prediction ($F(x)$) and the initial point annotation ($y$) as:

$$\begin{aligned} L_{ST}^{t+1} &= L_{ce}(F(x)^{t+1}, \hat{y}), \\ \hat{y}_{1,w,h} &= max(y_{1,w,h}, F(x)_{1,w,h}^{t}), \\ \hat{y}_{0,w,h} &= 1 - \hat{y}_{1,w,h}, \end{aligned} \quad (3)$$

where the new combined segmentation label $\hat{y}$ substitutes the point annotation $y$ in the initial cross entropy loss in Eq. 1. $\hat{y}_{1,w,h}$ represents the label of cell class at location $(w,h)$. We use the maximum operation to combine the current prediction ($F(x)$) and the point annotation ($y$) for cell pixels. Because the point annotation is extremely-weak and will be easily overwhelmed by the prediction ($F(x)$), using the $Maximum$ is more suitable than the convex combination as Eq. 2. $\hat{y}_{0,w,h}$ is for background class, ensuring that the summed probability at each pixel $(w,h)$ is 1.

From Eq. 3, we can easily tell that: the gold-standard ground-truth cell pixels labeled by point annotations with $y_{1,w,h} = 1$ are still equal to one in the updated label $\hat{y}_{1,w,h}$; and the other initially unlabeled pixels with $y_{1,w,h} = 0$ will possibly become cell pixels after the label updating from self-training, i.e., $\hat{y}_{1,w,h} = F(x)_{1,w,h}^{t}$, and $0 < \hat{y}_{1,w,h} < 1$ (the updated label $\hat{y}_{1,w,h}$ is a soft-label). Mathematically,

$$\begin{aligned} \hat{y}_{1,w,h} &= y_{1,w,h}, when\ y_{1,w,h} = 1 \\ \hat{y}_{1,w,h} &> y_{1,w,h}, when\ y_{1,w,h} = 0. \end{aligned} \quad (4)$$

Thus, the total amount of labeled foreground pixels, in terms of $\sum \hat{y}_{1,w,h}$ , is increased from the initial gold-standard ground-truth $\sum y_{1,w,h}$.

From Eq. 3, we can also easily tell that

$$\hat{y}_{1,w,h} \geq F(x)_{1,w,h}^{t}. \quad (5)$$

At time $t+1$, the network is updated by minimizing the self-training loss $L_{ST}^{t+1}$. The cross-entropy between $F(x)^{t+1}$ and $\hat{y}$ is minimized. So, the learned network $F(\cdot)^{t+1}$ should generate $F(x)_{1,w,h}^{t+1}$ close to $\hat{y}_{1,w,h}$, i.e.,

$$F(x)_{1,w,h}^{t+1} \approx \hat{y}_{1,w,h}. \quad (6)$$

From Eq. 5 and Eq. 6, we can tell that the new prediction output $F(x)^{t+1}$ at iteration $t+1$ shall have more soft-labeled

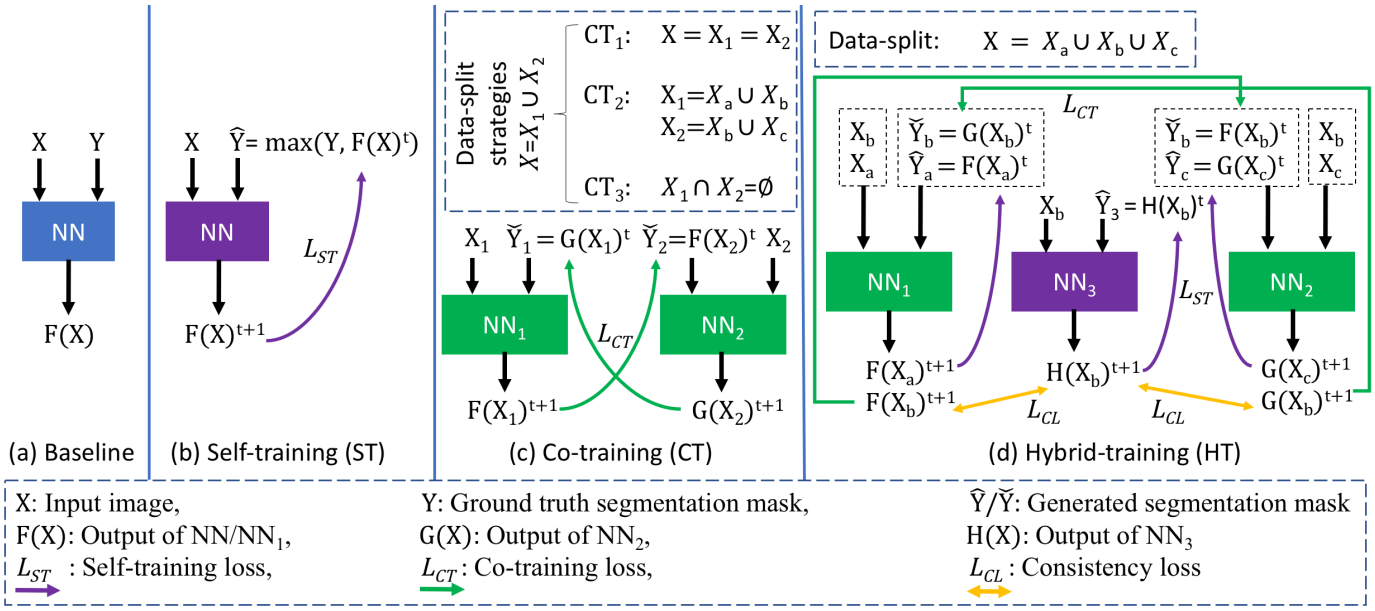


Fig. 6. The details of the training schemes: baseline, self-training, co-training and hybrid-training. a) In the baseline scheme, the neural network (NN) takes the image and the limited point annotation as the input and generates the segmentation mask as the output ($\boldsymbol{F(X)}$) b) The self-training scheme has one neural network and updates the target label during the training process by the combination of the ground truth label and self-predicted label and a divergence loss. c) The co-training scheme has a pair of neural networks which are trained by different subsets of the data and co-trained by each other to increase the segmentation ability. $\boldsymbol{CT_1}$, $\boldsymbol{CT_2}$ and $\boldsymbol{CT_3}$ represents three data-split strategies. d) The hybrid-training has three networks. Two networks are co-trained by each other and one network is self-trained. The consistency loss is applied between the co-trained networks and the self-trained network to increase the efficiency and stability of the network training.

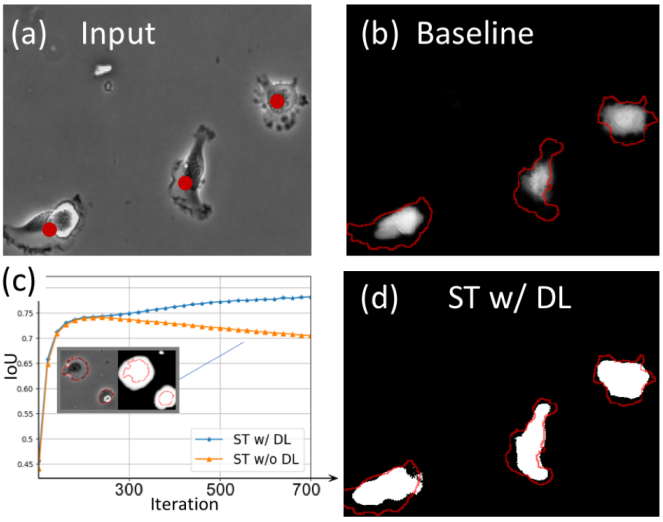


Fig. 7. The illustration for the self-training scheme. a) The input image with the point annotation (red dot). b) Segmentation results of the baseline method that is trained based on only point annotations. c) The training process of self-training (ST) with divergence loss (DL) vs. without DL. d) Segmentation results of self-training. Red contour: the boundary of ground truth mask.

foreground pixels than the previous output $F(x)^t$ at iteration t, i.e.,

$$F(x)^{t+1}_{1,w,h} \geq F(x)^{t}_{1,w,h}. \tag{7}$$

Thus, the total number of predicted foreground pixels is always increasing during the training iterations.

By increasing the foreground pixels over iterations, the network can successfully learn some cell regions near annotated points at the beginning, which is very helpful to reduce the false negatives as shown in Fig. 7(b) and increase the performance during the initial training, but then it learns more from false-positives beyond cell regions, yielding a training process with decreasing accuracy. As shown in Fig. 7(c), the performance of the iteratively-trained network model (red curve) quickly reaches the highest point, then decreases gradually. This is because the network over-trusts the output from the previous iteration and the two successive networks try to generate similar results. We name this phenomenon as a "self-deception" problem, since the algorithm is fooled by its output in the previous iteration. During the iterative self-training, the prediction results in the current iteration overwhelm the initial point annotation, as the segmentation label for the next iteration. In a trivial solution, two consecutive networks can treat the entire image as cell masks. To avoid this self-deception problem and let network know when to stop at a proper stage, we add a new divergence loss (DL) to the self-training loss:

$$\begin{aligned} L^{t+1}_{ST} &= L_{ce}(F(x)^{t+1}, \hat{y}) + \eta L^{t+1}_{DL}, \\ L^{t+1}_{DL} &= \sum_{k}^{K} max(\epsilon - |\frac{||F(x)^{t+1}_k||_1 - ||\hat{y}_k||_1}{(||F(x)^{t+1}_k||_1 + ||\hat{y}_k||_1)/2}|, 0), \end{aligned} \tag{8}$$

where $|\cdot|$ is the absolute value. $||\cdot||_1$ is the entrywise $l_1$ norm of a matrix. The first term in $L_{ST}$ is the cross-entropy loss, minimizing the distance between the updated label ($\hat{y}$) and network prediction ($F(x)$). The second term is the divergence loss ($L_{DL}$), maximizing the distance between label ($\hat{y}$) and the network prediction ($F(x)$), if their normalized distance is smaller than a small value $\epsilon$. $\eta$ is a hyper-parameter to balance the two loss terms. The divergence loss penalizes two successive networks that are too similar (i.e., with a distance less than $\epsilon$).

The cross-entropy loss and divergence loss are two combat-

ing terms. The cross-entropy loss always increase the number of predicted foreground pixels. On the contrary, the divergence loss always minimize the number of predicted foreground pixels by making the prediction $F(x)^{t+1}$ and the label $\hat{y}$ dissimilar. It successfully reduces the false-positive in the self-deception problem. The training curve of self-raining with divergence loss ('ST w/ DL') in Fig. 7(c) shows that this divergence loss effectively overcomes the self-deception problem, leading to a good converged training performance. The network will converge when the two combating terms are balanced. A segmentation example is shown in Fig. 7(d). We can see that the result of self-training with divergence loss covers the major portion of cells, with a big improvement compared to the baseline method which has more false negatives (Fig. 7(b)) and self-training without the divergence loss (the segmentation result in Fig. 7(c) which has more false positives). Our next step is to explore co-training and try to obtain more precise cell boundaries.

## B. Co-training

The co-training (CT) method has a pair of networks. With the training dataset $X$ split into two subsets $X_1$ and $X_2$, two networks are self-trained by $X_1$ and $X_2$, separately. Then, the two networks start to supervise each other. Below, we describe the training procedure of the first neural network ($NN_1$). The second network ($NN_2$) will be trained in a similar way.

Let $x \in X_1$ be an input image to $NN_1$. The outputs of $NN_1$ and $NN_2$ are denoted as $F(\cdot)$ and $G(\cdot)$, respectively. The label of $x$ for $NN_1$ is the prediction from $NN_2$:

$$\check{y}_{1,w,h} = G(x)^t_{1,w,h},\ \check{y}_{0,w,h} = 1 - \check{y}_{1,w,h}. \tag{9}$$

Then, the co-training loss for for training $NN_1$ to obtain $F(\cdot)$ is defined as:

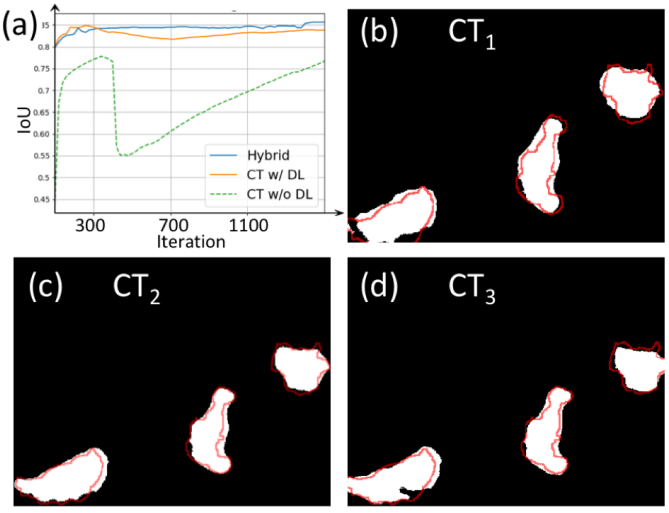


Fig. 8. The illustration for the co-training scheme.a) The training process of co-training (CT) with divergence loss (DL) vs. without DL and hybrid-training. b-d) Segmentation masks of three CT methods with different data-split strategies, as described in Fig. 6(c). The data split method used to illustrate the difference between CT w/w/o divergence loss in (a) is the $\boldsymbol{CT_3}$. Red contour: the boundary of ground truth mask.

$$L^{t+1}_{CT_F} = L_{ce}(F(x)^{t+1}, \check{y}) + \eta \sum_{k}^{K} max(\epsilon - |\frac{||F(x)^{t+1}_k||_1 - ||\check{y}_k||_1}{(||F(x)^{t+1}_k||_1 + ||\check{y}_k||_1)/2}|, 0). \tag{10}$$

The first term is the cross-entropy loss which lets the output of the two networks become more similar. However, by only using the cross-entropy loss, the two networks can learn the drawback of each other and yield an unstable training process. As shown by the green curve in Fig. 8(a), when the two networks start co-training, their training process contains the risky performance drop. Therefore, we apply the divergence loss as the second term in the co-training loss, which keeps the two networks from being too similar. The divergence loss lets the two networks learn from each other while keeps their difference, leading to a stable training process as shown by the red curve in Fig. 8(a). Similarly, we train $NN_2$. The final predicted segmentation mask on an image is the averaged mask from $NN_1$ and $NN_2$.

There are several data-split strategies in co-training, as illustrated on top of Fig. 6(c). In strategy $CT_1$, the data used for two networks are the same, i.e., $X = X1 = X2$. In strategy $CT_2$, the two networks share a portion of data, i.e., $X = X_a \cup X_b \cup X_c$, $X_1 = X_a \cup X_b$ and $X_2 = X_b \cup X_c$. In strategy $CT_3$, the data used for the two networks are exclusive, $X = X_1 \cup X_2$ and $X_1 \cap X_2 = \phi$. Some segmentation examples related to the three data-split strategies are shown in Fig. 8(b-d). Generally, strategies $C_2$ and $C_3$ are better than strategy $C_1$. More quantitative comparisons are given in the experiment section.

## C. Hybrid-training

From the previous two training schemes, we find that the self-training converges more smoothly than the co-training. The co-training, because of the supervision between two networks, can get a better final segmentation result but its training process is less stable (e.g., 'CT w/ DL' and 'CT w/o DL' training curves in Fig. 8(a) have some unstable performance drop). Thus, we investigate a hybrid-training (HT) to leverage the advantages of both self-training and co-training.

The hybrid-training, as illustrated in Fig. 6(d) has 3 networks $NN_1$, $NN_2$ and $NN_3$. The dataset is split into 3 non-overlapped parts $X_a$, $X_b$ and $X_c$. $NN_1$ is trained by datasets $X_a$ and $X_b$. $NN_2$ is trained by datasets $X_b$ and $X_c$. Specifically, $NN_1$ and $NN_2$ are co-trained on $X_b$, and they are self-trained on $X_a$ and $X_c$, separately. $NN_3$ is self-trained by $X_b$ only. To maintain the consensus among the three networks, a consistency loss (CL) is proposed:

$$L^{t+1}_{CL} = \sum_{x \in X_b} (|F(x)^{t+1} - H(x)^t| + |G(x)^{t+1} - H(x)^t|), \tag{11}$$

where $F(x)$, $G(x)$, and $H(x)$ denote the learned network models of $NN_1$, $NN_2$ and $NN_3$, respectively. The total loss in the hybrid training is defined as

$$L^{t+1}_{HT} = L^{t+1}_{CT_F} + L^{t+1}_{CT_G} + L^{t+1}_{ST_H} + \lambda L^{t+1}_{CL}. \tag{12}$$

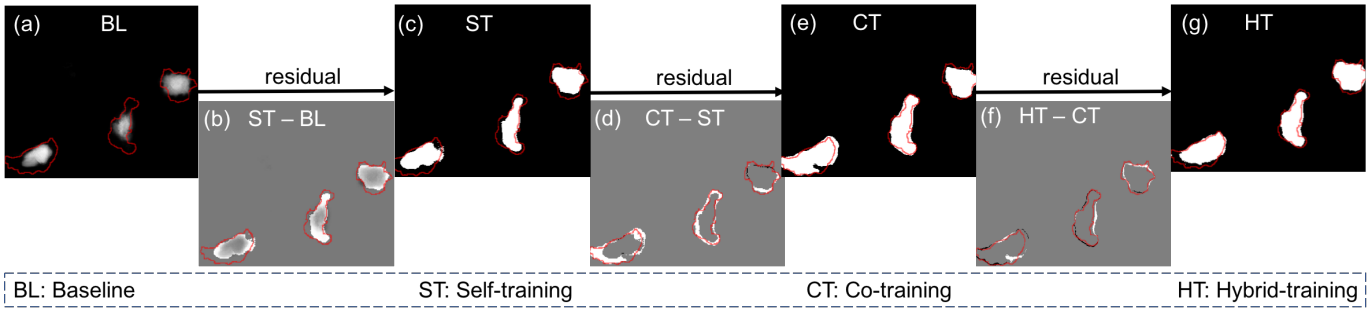


Fig. 9. The improvement from baseline (a) all the way to hybrid-training (g). b) The improvement from (a) baseline to (c) self-training (ST), visualized by ST minus BL; d) The improvement from (c) self-training (ST) to (e) co-training (CT); f) The improvement from (e) co-training (CT) to (g) hybrid-training (HT). The white pixel in the residual map represents 1. The grey pixel represents 0. The black pixel represents -1.

where $\lambda$ is a hyper-parameter to balance the loss terms. After obtaining the networks from the hybrid training, the final predicted segmentation mask is their average. One segmentation example by Hybrid-training is shown in Fig. 9(g).

In the following, we analyze the improvement from the baseline, to self-training, to co-training, and then to hybrid-training. The residual mask in Fig. 9(b) shows the difference between the result of self-training and the result of the baseline, where the baseline mask is subtracted from the self-training mask. The white pixels in Fig. 9(b) represent those pixels that are classified as cell pixels by the self-training method but misclassified as background pixels by the baseline. This residual mask clearly shows that more cell regions are correctly segmented by the self-training method, compared to the baseline.

The residual mask in Fig. 9(d) shows the difference between the co-training result and self-training result, where the self-training mask is subtracted from the co-training mask. The white pixels are classified as cell pixels by the co-training method but misclassified as background pixels by the self-training method. On the contrary, the black pixels are classified as cell pixels by the self-training method but misclassifed as background pixels by the co-training method. We can see that the co-training method can segment more regions around cell boundaries, compared to the self-training method.

Similarly, the residual mask in Fig. 9(f) shows that the hybrid-training method can further refine the result from the co-training, particularly around irregular cell boundary regions. The quantitative comparisons on the improvements are provided in the experiment section.

### D. Weakly Supervised Learning with Human in the Loop

Our weakly supervised training schemes can train cell segmentation networks in a very efficient way via point annotations. Its segmentation accuracy may not be equivalent to other fully supervised methods. But, the weakly supervised cell segmentation has its practical advantages in many important biological applications as we summarized in the motivation of our proposal (Section I-B). To pursue both efficiency and accuracy if biologists care a lot about perfect cell boundaries, we propose weakly supervised learning with human in the loop, so the high segmentation accuracy can be achieved with a little more human efforts than point annotations but much less than the full supervision.

First, we train cell segmentation networks using point-annotation-based weakly supervised methods, as a pre-train stage. The segmentation results by the pre-trained cell segmentation networks, visualized as contours or masks overlaid on input images, can show human annotators where the weakly-supervised methods make mistakes. Thus, human annotators can be guided to provide more annotations on those image regions with mistakes, either providing more point annotations there or providing full cell masks for those cells. Finally, the cell segmentation networks can be fine-tuned by the extra point annotations and/or some full cell masks using the cross-entropy loss. By keeping human in the loop of the weakly supervised learning, we can achieve precise segmentation performance as the fully supervised method, but with much less human annotation efforts. We show the effectiveness of this human-assisted weakly learning in the experiment section.

## IV. Experiments

In this section we evaluate our proposed method on two phase-contrast cell image datasets. We provide the ablation studies on self-training, co-training and hybrid-training, and then compare with the state-of-the-art unsupervised, weakly supervised and fully supervised methods.

### A. Datasets and Implementation Details

We validate our approach on two datasets: (1) PHC dataset from the ISBI cell segmentation challenge [54], [56], which contains 230 images. 196 images are used for training. 34 images are for testing. (2) Phase100 dataset from [22], which contains 100 microscopy images. 60 images are used for training and the rest is for testing. The size of the used microscopy image is 520x696 for the PHC dataset and 512x512 for the Phase100 dataset. In the training data, we only use one point annotation per cell. The testing data have the ground truth segmentation masks.

The neural networks used to validate the weakly supervised cell segmentation include four networks reviewed in Section II: FCN 11 [22], FCN16 [23], U-Net [21], and Pyramid-based FCN [22]. The FCN11 has 11 convolutional layers with batch-normalization on the first 9 layers. The FCN16 is the VGG16-based fully convolutional network. The U-Net contains 4-levels of downsampling and upsampling. The Pyramid-based FCN contains 3 levels of FCNs. Our training schemes can be

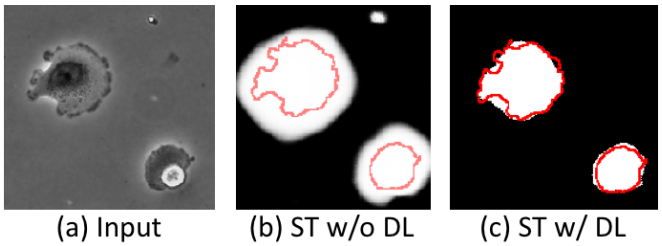


Fig. 10. Examples of self-training (ST) without or with divergence loss (DL).

TABLE I
ABLATION STUDY ON SELF-TRAINING ON PHC [54] DATASET

| Methods: | Baseline | ST w/o DL | ST |
|---|---|---|---|
| IOU: | 0.575± 0.043 | 0.718 ± 0.0056 | 0.799 ± 0.0005 |

applied to other segmentation networks as well. The learning rate is set as $10^{-4}$. $\eta = 10^{-4}$, $\epsilon = 0.03$ and $\lambda = 0.1$, by cross-validations. When training the segmentation networks, we employ the on-the-fly data augmentation, which means at every iteration, the data batch is randomly augmented. For each iteration, an image is shifted randomly within 20 pixels, scaled between 0.9-1.1, and rotated by 90, 180, or 270 degrees.

### B. Ablation Study on Self-training

In this section, we compare three methods: (1) Baseline that is trained only by the point annotation; (2) Self-training without the divergence loss ($ST\ w/o\ DL$); and (3) Self-training with the divergence loss ($ST$). We use the U-Net as the backbone for this ablation study. The evaluation metric is the IOU (pixel-wise intersection over union, between the segmentation and ground truth). The results are shown in Table I. $ST\ w/o\ DL$ works better than the baseline method, but because of the self-deception problem, it tends to segment too much cell regions (false-positives), as shown in Fig. 10(b). With the divergence loss that enforces the distance between the predicted mask $F(x)$ and the label $\hat{y}$, $ST$ generates better segmentation mask (Fig. 10(c)) and achieves better segmentation performance (Table I).

To analyze how the location of point annotation will affect the performance of segmentation algorithms, we implement a sensitivity study by using the point annotation at different random-selected locations. We generate 5 different sets of point annotations for the training images (e.g., Fig. 11). The point is generated by randomly shift the original point annotations from human annotators. Some misgenerated points that are outside of the cell regions are manually corrected by moving them back to cell boundaries. As shown in Fig. 11, some points are on the boundary of the cell. The mean of the IoU scores, from the ST based on the five different sets of point annotations, are given in Table I. The variance is as low as 0.0005, which validates that the proposed training scheme is robust to the location of the point annotations. This nice property can save the human labeling effort because annotators can quickly drop a point on cell regions rather than spending time to draw a point close to cell centroids.

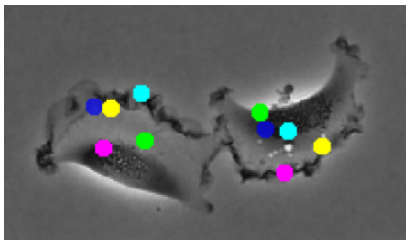

Fig. 11. Example of five different sets of randomly-selected point annotations. One color represents one set.

### C. Ablation Study on Co-training

In this section, we compare 4 co-training methods: (1) $CT\ w/o\ DL$: the co-training method without divergence loss ($DL$), where the data is split as $X = X_1 \cup X_2$ and $X_1 \cap X_2 = \phi$; (2) $CT1$: the co-training method with $DL$, where the data is split as $X = X_1 = X_2$; (3) $CT2$: the co-training method with $DL$, where the data is split as $X = X_a \cup X_b \cup X_c$ and $X_1 = X_a \cup X_b$ and $X_2 = X_b \cup X_c$; and (4) $CT3$: the co-training method with $DL$, where the data is split as $X = X_1 \cup X_2$ and $X_1 \cap X_2 = \phi$. We use the U-Net as the backbone for this ablation study. As shown in Table II, the co-training without the divergence loss ($CT\ w/o\ DL$) does not work as well as other other co-training methods with $DL$. The two co-training methods ($CT2$ and $CT3$) with overlapped training datasets or disjoint training datasets work better than $CT1$ that uses the identical training dataset in the co-training. The paired networks in the co-training methods $CT2$ and $CT3$ learn from different images, so they can bring extra information to the co-training process, leading to better performance than CT1 that uses the identical training dataset for two networks.

### D. Ablation Study on Hybrid-training

From Fig. 8(a), we observe that at the beginning of co-training, the performance firstly drops then increases gradually. With divergence loss this problem is almost fixed but with a little drop. This is because the two different networks need several iterations to assimilate. Thus, the training process in co-training is not very stable, though its performance is better than the self-training, by comparing Tables I and II. From Fig. 7(c), we observed that the 'ST w/ DL' (self-training with divergence loss) converges stably. Hence, the hybrid training ($HT$) combines co-training and self-training with consistency loss can overcome the unstable problem and achieve good accuracy. We use the U-Net as the backbone for this ablation study. Validated from Table III, the hybrid training ($HT$) achieves a better segmentation performance than co-training methods in Table II and self-training methods in Table I, and its training process is stable (the blue 'Hybrid' curve in Fig. 8(a)). In Table III, we also validate that the divergence loss and consistency loss help improve the performance of hybrid training.

### E. Comparison with Weakly Supervised Methods

We compare our weakly-supervised segmentation method with several representative weakly-supervised methods such

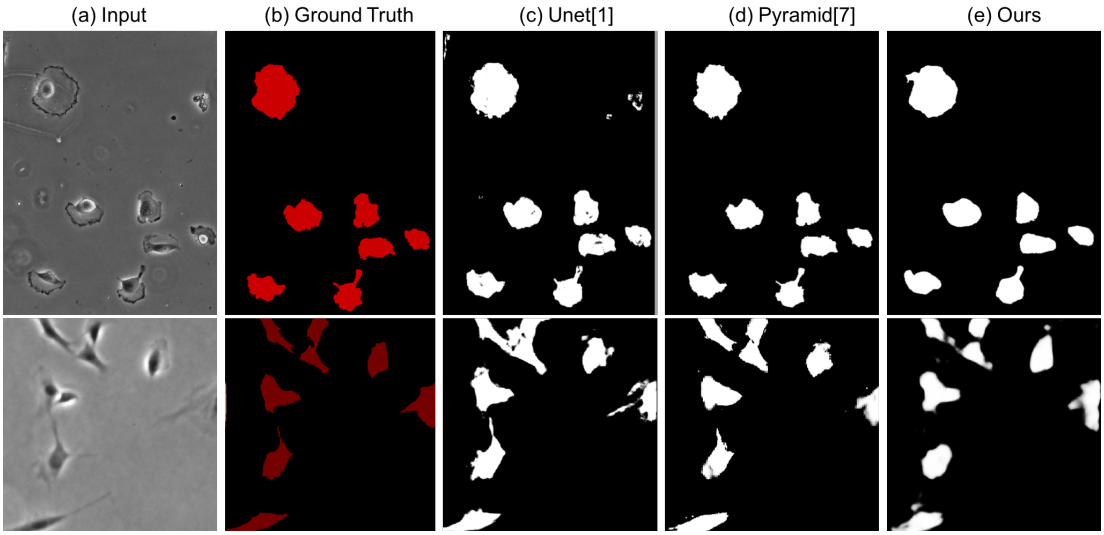


Fig. 12. Segmentation on PHC dataset [54] (1st row) and Phase100 dataset [22] (2nd row).

TABLE II
RESULTS OF ABLATION STUDY ON CO-TRAINING ON PHC [54] DATASET

| Methods: | CT w/o DL | CT1 | CT2 | CT3 |
|---|---|---|---|---|
| IOU: | 0.782 | 0.817 | 0.842 | 0.844 |

TABLE III
RESULTS OF ABLATION STUDY ON HYBRID-TRAINING ON PHC [54] DATASET

| Methods: | HT w/o DL&CL | HT w/o DL | HT w/o CL | HT |
|---|---|---|---|---|
| IOU | 0.801 | 0.823 | 0.858 | 0.865 |

as the GrabCut method [6] based on box annotations, a U-Net [21] based on scribble annotations, KTH-SE method [56] based on thresholding and watershed transform, and Qu et al. [44] based on Voronoi diagram and the k-means clustering methods to generate coarse labels and conditional random field to refine the boundary, using the IOU evaluation metric. Our training schemes can be applied to different segmentation networks, for example, the four networks reviewed in Section II: FCN11 [22] (a 11-layer fully convolutional network), FCN16 [23] (VGG16-based fully convolutional network), U-Net [21], and Pyramid-based FCN [22]. The quantitative results are in Table IV. Using pyramid-based FCN as the backbone achieves the best segmentation performance.

TABLE IV
EXPERIMENT RESULTS ON PHC [54] DATASET AND PHASE100 [22] DATASET, COMPARED WITH OTHER WEAKLY SUPERVISED METHODS

| Method | PHC [54] IOU | Phase100 [22] IOU |
|---|---|---|
| GraphCut [6] | 0.810 | - |
| Scribble w U-Net [21] | 0.826 | 0.657 |
| KTH-SE [56] | 0.795 | - |
| Qu et al. [44] | 0.757 | 0.773 |
| Ours w FCN11 [22] | 0.796 | 0.763 |
| Ours w FCN16 [23] | 0.805 | 0.773 |
| Ours w U-Net [21] | 0.865 | 0.808 |
| Ours w Pyramid-based Network [22] | 0.871 | 0.811 |

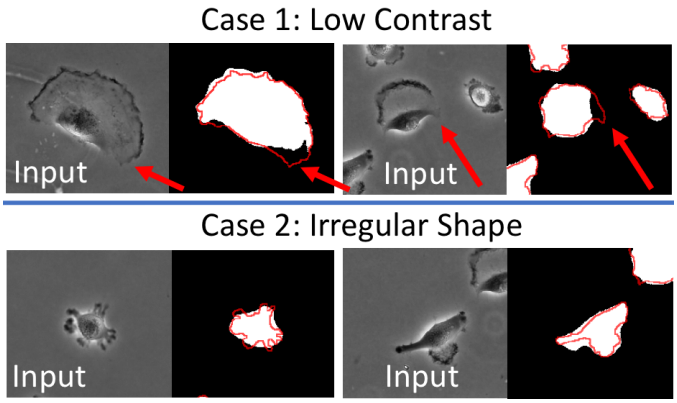


Fig. 13. Examples of failure cases. Red contour: the boundary of ground truth.

TABLE V
EXPERIMENT RESULTS ON PHC [54] DATASET, COMPARED WITH OTHER FULLY SUPERVISED METHODS

| Method | IOU | AUC |
|---|---|---|
| U-Net [21] | 0.920 | - |
| FCN11 [22] | 0.864 | 0.883 |
| FCN16 [23] | 0.921 | 0.949 |
| Pyramid [22] | 0.938 | 0.963 |
| GRUU [24] | 0.938 | - |
| Ours | 0.871 | 0.922 |

### *F. Comparison with Fully Supervised Methods*

We compare our weakly-supervised segmentation method with five fully supervised methods that are trained using completely-annotated cell masks: U-Net [21], FCN11 [22], FCN16 [23], pyramid-based FCN [22] (Pyramid), and convolutional and gated recurrent neural network [24] (GRUU). The quantitative results are summarized in Table V (PHC dataset) and Table VI (Phase100 dataset) regarding to two metrics: IOU and Area-Under-Curve (the curve of precision vs. recall). The results of the compared methods in Table V and Table VI are referred from their papers directly. The performance of our hybrid training method with pyramid-based FCN as the backbone is a little worse than the fully-supervised methods. Some qualitative results are shown in Fig. 12. Though the

TABLE VI
EXPERIMENT RESULTS ON PHASE100 [22] DATASET, COMPARED WITH OTHER FULLY SUPERVISED METHODS

| Method | IOU | AUC |
|---|---|---|
| U-Net [21] | 0.844 | 0.932 |
| Pyramid [22] | 0.912 | 0.970 |
| Ours | 0.811 | 0.903 |

quantitative score of our weakly supervised method is lower than some fully-supervised methods, our qualitative results look smoother without any post-process.

We want to emphasize that, those full-supervised networks are trained by segmentation masks (795624 annotated pixels in PHC and 902938 annotated pixels in Phase100). Our weakly-supervised methods only use 1242 annotated pixels in PHC (point annotation on 1242 cells in 190 images) and 358 annotated pixels in Phase100 (point annotation on 358 cells in 60 images). The averaged annotation time for one point-annotation is about 0.95 second (the total annotation time divided by the number of annotated images). The annotation time for one mask-annotation of one cell is 63 seconds. Our annotation-efficient method greatly reduces the human labeling cost while achieving performances close to fully-supervised methods. The annotation-efficient deep learning for cell segmentation is very suitable for high-throughput biological experiments with large variations from imaging modalities or specimen appearances, as we summarized in our proposal motivation in Section I-B.

Two major cases on which our approach currently does not work well, compared to fully supervised methods, are (Fig. 13): (1) cell regions with low contrast compared to the background, and (2) cells with irregular shapes that are difficult to get precise boundaries. If some biological applications require high accurate cell segmentation masks while prefers the least human annotation efforts, we propose to explore the point-annotation-based weakly supervised learning with human in the loop, aiming at achieving both high accuracy and annotation efficiency simutaneously.

### G. Weakly Supervised Learning with Human in the Loop

In this section we evaluate the effectiveness of the weakly supervised cell segmentation with human in the loop, whose goal is to archive high-quality segmentation performance as fully supervised methods, but with much less human annotation efforts. Our human-in-the-loop method can supply extra point annotations at locations where our weakly-supervised method makes mistakes. For example, in Fig. 13, we can observe where are major image regions with segmentation mistakes, such as those low-contrast areas. For this experiment, human annotator added extra point annotations (about 10% of the dataset) on those areas with mistakes. Then, the segmentation networks initially trained by the weakly supervised method are fine-tuned by the extra annotation points. With the increased number of point annotations, the segmentation performance (denoted as 'Ours w/ extra pts' in Table VII) increases a little but not a lot. Since more accurate annotations are needed for precise segmentation boundaries, human annotators can provide fully annotated masks for those misclassified cells. We investigate how the segmentation performance is affected by the additional fully-annotated masks. The results are summarized in Table VII, denoted as 'Ours/ w $p\%$', where $p\%$ represents the percentage of cells that are fully annotated for training. The backbone network used in this study is the Pyramid-based FCN. The fully supervised methods annotate 100% of the cells in the training data. As we can see from the table, by annotating only 30% of the cells in the training data, our method can achieve the results close to fully supervised results reported in Table V and Table VI.

TABLE VII
EXPERIMENT RESULTS ON WEAKLY SUPERVISED LEARNING WITH HUMAN IN THE LOOP

| Method | PHC Dataset [54] | | Phase100 Dataset [22] | |
|---|---|---|---|---|
| | IOU | AUC | IOU | AUC |
| Ours w/ extra pts | 0.878 | 0.927 | 0.827 | 0.901 |
| Ours w/ 10% | 0.903 | 0.945 | 0.876 | 0.932 |
| Ours w/ 20% | 0.922 | 0.957 | 0.883 | 0.934 |
| Ours w/ 30% | 0.928 | 0.968 | 0.885 | 0.935 |
| Ours w/ 40% | 0.931 | 0.969 | 0.887 | 0.940 |
| Ours w/ 50% | 0.932 | 0.969 | 0.891 | 0.941 |

## V. CONCLUSION

We present end-to-end self-training, co-training and hybrid-training methods for the weakly supervised cell segmentation with only point annotations on cells. A divergence loss and a consistency loss are proposed to avoid the overfitting, enforce the stable training process, and maintain the consensus among multiple co-trained networks. We also propose the weakly supervised learning with human in the loop to achieve annotation-efficient deep learning for accurate cell segmentation. Our proposal is validated on two public benchmark datasets with segmentation performances close to fully supervised methods, but with much less human annotation effort.

## REFERENCES


[1] E. Meijering, "Cell segmentation: 50 years down the road [life sciences." *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 140-145, 2012.

[2] E. D. Gelasca, B. Obara, D. Fedorov, K. Kvilekval, and B. Manjunath, "A biosegmentation benchmark for evaluation of bioimage analysis methods," *BMC Bioinformat.*, vol. 10, no. 368, pp. 1–12, Nov. 2009.

[3] F. Xing, and L. Yang. "Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review." *IEEE reviews in biomed. engineer.* vol. 9, pp. 234-263, 2016.

[4] M. Rousson and N. Paragios, "Shape priors for level set representations," *Eur. Conf. Comput. Vis.*, vol. 2351, pp. 78–92, 2002.

[5] X. Wu and S. K. Shah, "Cell segmentation in multispectral images using level sets with priors for accurate shape recovery," *IEEE Int. Symp. Biomed. Imag.*, pp. 2117–2120, Mar. 2011.

[6] R. Bensch, and O. Ronneberger. "Cell segmentation and tracking in phase contrast images using graph cut with asymmetric boundary costs," *IEEE Int. Symp. Biomed. Imag.*(ISBI), pp. 1220-1223, Apr. 2015.

[7] Y. Boykov, O. Veksler, R. Zabih "Fast approximate energy minimization via graph cuts," *IEEE Trans. on pattern analys. and mach. intellig.* vol. 23, no. 11, pp. 1222–1239, 2001.

[8] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[9] S. Andrews, G. Hamarneh, and A. Saad, "Fast random walker with priors using precomputation for interactive medical image segmentation," *Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, vol. 6363, pp. 9–16, 2010.

[10] C. Zhang, J. Yarkony, and F. A. Hamprecht, "Cell detection and segmentation using correlation clustering," *Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, vol. 8673, pp. 9–16, 2014.

[11] F. Xing and L. Yang, "Unsupervised shape prior modeling for cell segmentation in neuroendocrine tumor," *IEEE Int. Symp. Biomed. Imag.*, pp. 1443–1446, Apr. 2015.

[12] L. Yang, P. Meer, and D. J. Foran, "Unsupervised segmentation based on robust estimation and color active contour models," *IEEE Trans. Inf. Technol. Biomed.*, vol. 9, no. 3, pp. 475–486, Sep. 2005.

[13] J. C. Caicedo, et al. "Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl," *Nature methods* vol. 16, no. 12, pp. 1247-1253, 2019.

[14] L. P. Coelho, A. Shariff, and R. F. Murphy, "Nuclear segmentation in microscope cell images: A hand-segmented dataset and comparison of algorithms," *IEEE Int. Symp. Biomed. Imag.*, pp. 518– 521, Jun. 2009.

[15] W. Beaver, D. Kosman, G. Tedeschi, E. Bier, W. McGinnis, and Y. Freund. "Segmentation of nuclei in confocal image stacks using performance based thresholding," *IEEE Int. Symp. Biomed. Imag.*, pp. 53-56, Apr. 2007.

[16] K. Nandy, P. R. Gudla, and S. J. Lockett. "Automatic segmentation of cell nuclei in 2D using dynamic programming," *2nd Workshop on Microsopic Image Analysis with Applications in Biology*. vol. 57, no. 7, pp. 1676-1689, Feb. 2007.

[17] C. Jung, C. Kim, S. W. Chae, and S. Oh, "Unsupervised segmentation of overlapped nuclei using Bayesian classification," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 12, pp. 2825–2832, Dec. 2010.

[18] Y. Zhou, H. Chang, K. E. Barner, and B. Parvin, "Nuclei segmentation via sparsity constrained convolutional regression," *IEEE Int. Symp. Biomed. Imag.*, pp. 1284–1287, Apr. 2015.

[19] E. Moen, et al. "Deep learning for cellular image analysis." *Nature methods*, pp. 1-14, 2019.

[20] F. Xing, et al. "Deep learning in microscopy image analysis: A survey." *IEEE Trans. on Neur. Net. and Learn. Syst.* vol. 29, no. 10, pp. 4550-4568, 2017.

[21] O. Ronneberger, p. Fischer, T. Brox, "U-net: Convolutional networks for biomedical image segmentation." *Int. Conf. on Med. Image Comput. Comput.-Assisted Intervention*, pp. 234-241, 2015.

[22] T. Zhao, and Z. Yin, "Pyramid-based fully convolutional networks for cell segmentation." *Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, pp. 677-685, 2018.

[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation." *Conf. on Comput. Vis. and Pattern Recog.*, pp. 3431-3440, 2015.

[24] T. Wollmann et al. "GRUU-Net: Integrated convolutional and gated recurrent neural network for cell segmentation." *Med. imag. analys.* vol. 56, pp. 68-79, 2019.

[25] P. Naylor, M. Laé, F. Reyal and T. Walter, 'of Nuclei in Histopathology Images by Deep Regression of the Distance Map," *IEEE Trans. on Med. imag.*, vol. 38, no. 2, pp. 448-459, Feb. 20199.

[26] D. Liu, D. Zhang, Y. Song, C. Zhang, F. Zhang, L. O'Donnell, and W. Cai. "Nuclei Segmentation via a Deep Panoptic Model with Semantic Feature Fusion." *Int. Joint Conf. on Art. Intellig*, pp. 861-868. 2019.

[27] Y. Zhou, O. F. Onder, Q. Dou, E. Tsougenis, H. Chen, and P. Heng. "Cia-net: Robust nuclei instance segmentation with contour-aware information aggregation." *Int. Conf. on Inf. Process. in Med. Imag.*, pp. 682-693, 2019.

[28] J. Song, L. Xiao and Z. Lian, "Contour-Seed Pairs Learning-Based Framework for Simultaneously Detecting and Segmenting Various Overlapping Cells/Nuclei in Microscopy Images," *IEEE Trans. on Imag. Process.*, vol. 27, no. 12, pp. 5759-5774, Dec. 2018.

[29] K, He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." *Conf. on Comput. Vis. and Pattern Recog.*, pp. 770-778. 2016.

[30] H. Su, F. Liu, Y. Xie, F. Xing, S. Meyyappan, and L. Yang, "Region segmentation in histopathological breast cancer images using deep convolutional neural network," *IEEE Int. Symp. Biomed. Imag.*, pp. 55–58, Apr. 2015.

[31] T. Zhao, D. Gao, J. Wang, and Z. Yin, "Lung segmentation in ct images using a fully convolutional neural network with multi-instance and conditional adversary loss," *IEEE Int. Symp. Biomed. Imag.*, pp. 505-509, Apr. 2018.

[32] T. Zhao, Z. Yin, J. Wang, D. Gao, Y. Chen, Y. Mao "Bronchus Segmentation and Classification by Neural Networks and Linear Programming," *Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*. pp. 230-239, 2019.

[33] L. Zhang, et al."Robust Pancreatic Ductal Adenocarcinoma Segmentation with Multi-Institutional Multi-Phase Partially-Annotated CT Scans, " *Int. Conf. Med. Image Comput. Comput.-Assisted Intervention* , pp. 491-500, 2020.

[34] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. "Deep neural networks segment neuronal membranes in electron microscopy images." *Advanc. in neur. inf. processing syst.* , pp. 2843-2851, 2012.

[35] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weaklyand semi-supervised learning of a deep convolutional network for semantic image segmentation," *IEEE int. conf. on comput. vis.*, pp. 1742–1750, 2015.

[36] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," *IEEE int. conf. on comput. vis.*, pp. 1635–1643, 2015.

[37] M. Rajchl, M. C. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, and W. Bai, et al., "Deepcut: Object segmentation from bounding box annotations using convolutional neural networks," *IEEE trans. on medic. imaging*, vol. 36, no. 2, pp. 674–683, 2017.

[38] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble supervised convolutional networks for semantic segmentation," *IEEE Conf. on Comput. Vis, and Pattern Recog.*, pp. 3159–3167, 2016.

[39] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, ""What's the point: Semantic segmentation with point supervision," *Eur. Conf. Comput. Vis.*, pp. 549–565, 2016.

[40] K. Nishimura, and R. Bise, "Weakly Supervised Cell Instance Segmentation by Propagating from Detection Response." *Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, pp. 649-657, 2019.

[41] A G. Fidel, P. D M. Fernandez, T. I. Ren, and A. Cunha, "A weakly supervised method for instance segmentation of biological cells." *Medical Image Learning with Less Labels and Imperfect Data, MICCAI Workshop*, pp. 216–224, 2019.

[42] Z. Zhao, L. Yang, H. Zheng, I. H. Guldner, S. Zhang, and D. Z. Chen, "Deep learning based instance segmentation in 3d biomedical images using weak annotation," *Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, pp. 352–360, 2018.

[43] H. Lee, and J. Won-Ki. "Scribble2Label: Scribble-Supervised Cell Segmentation via Self-Generating Pseudo-Labels with Consistency." arXiv preprint arXiv:2006.12890, 2020.

[44] H. Qu, et al. "Weakly Supervised Deep Nuclei Segmentation Using Partial Points Annotation in Histopathology Images." *IEEE Trans. on Med. Imaging*, vol. 39, no. 11, pp. 3655-3666, 2020.

[45] A. Chamanzar, and Y. Nie. "Weakly Supervised Multi-Task Learning for Cell Detection and Segmentation." *IEEE Int. Symp. Biomed. Imag.*, pp. 513-516, 2020.

[46] S. Obikane, and Y. Aoki. "Weakly Supervised Domain Adaptation with Point Supervision in Histopathological Image Segmentation." *Asian Conf. on Pattern Recog.*, pp. 127-140, 2019.

[47] D. Liu, D. Zhang, Y. Song, F. Zhang, L. O'Donnell, H. Huang, M. Chen, and W. Cai. "Unsupervised Instance Segmentation in Microscopy Images via Panoptic Domain Adaptation and Task Re-weighting." *IEEE Conf. on Comput. Vis, and Pattern Recog.*, pp. 4243-4252, 2020.

[48] L. Hou, A. Agarwal, D. Samaras, T. M. Kurc, R. R. Gupta, and J. H. Saltz. "Robust histopathology image analysis: to label or to synthesize?." *IEEE Conf. on Comput. Vis, and Pattern Recog.*, pp. 8533-8542. 2019.

[49] P. Zhang, Y. Zhong, and X. Li. "ACCL: Adversarial constrained-CNN loss for weakly supervised medical image segmentation." arXiv preprint arXiv:2005.00328, 2020.

[50] H. Kervadec, J. Dolz, M. Tang, E. Granger, Y. Boykov, and I. B. Ayed, "Constrained-cnn losses for weakly supervised segmentation," *Med. imag. analys.*, vol. 54, pp. 88–99, 2019.

[51] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods." *33rd annual meeting on Association for Computational Linguistics*, pp. 189–196, 1995.

[52] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. "Training deep neural networks on noisy labels with bootstrapping." arXiv preprint arXiv:1412.6596, 2014.

[53] A. Blum and T. Mitchell. "Combining labeled and unlabeled data with co-training." *eleventh annual conference on Computational learning theory*, pp. 92–100. ACM, 1998.

[54] M. Maška, et al. "A benchmark for comparison of cell tracking algorithms." *Bioinformatics* vol. 30, no. 11, pp. 1609-1617, 2014.

[55] S. Kothari, Q. Chaudry, and M. D. Wang, "Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques," *IEEE Int. Symp. Biomed. Imag.*, pp. 795– 798, Apr. 2009.

[56] V. Ulman et al. "An objective comparison of cell-tracking algorithms." *Nature methods* vol. 14, no. 12, pp. 1141-1152, 2017.