



OPEN **BoxCell: Leveraging SAM for Cell Segmentation with Box Supervision**

Aayush Kumar Tyagi¹✉, Vaibhav Mishra², Prathosh A.P.³ & Mausam^{1,2}

Cell segmentation in histopathological images is vital for diagnosis, and treatment of several diseases. Annotating data is tedious, and requires medical expertise, making it difficult to employ supervised learning. Instead, we study a weakly supervised setting, where only bounding box supervision is available, and present the use of Segment Anything (SAM) for this without any finetuning, i.e., directly utilizing the pre-trained model. We propose BoxCell, a cell segmentation framework that utilizes SAM's capability to interpret bounding boxes as prompts, *both* at train and test times. At train time, gold bounding boxes given to SAM produce (pseudo-)masks, which are used to train a standalone segmenter. At test time, BoxCell generates two segmentation masks: (1) generated by this standalone segmenter, and (2) a trained object detector outputs bounding boxes, which are given as prompts to SAM to produce another mask. Recognizing complementary strengths, we reconcile the two segmentation masks using a novel integer programming formulation with intensity and spatial constraints. We experiment on three publicly available cell segmentation datasets namely, CoNSep, MoNuSeg, and TNBC, and find that BoxCell significantly outperforms existing box supervised image segmentation models, obtaining 6-10 point Dice gains.

Keywords Cell segmentation, Box-supervision, Segment anything

Cell segmentation serves as a crucial component for numerous applications, including survival prediction¹, tumor/non-tumor classification², as well as cell counting³. Our focus is on cell segmentation for histopathology images, which are obtained from tissue biopsies. Generally, cell segmentation models require pixel-level annotations, which are labor-intensive and expensive to obtain. This is because, typically, many cells are present within a single image, and annotating them requires trained pathologists. Weakly Supervised Image Segmentation (WSIS) addresses this challenge by using weak annotations, which may be present as image-level annotations⁴, scribbles⁵, point annotations⁶, or bounding boxes⁷⁻¹¹. We study bounding box supervision, as it offers a more accurate estimate of a cell boundary¹².

In this work, we explore WSIS with bounding box supervision for cell segmentation in histopathology images, utilizing Segment Anything Model (SAM)¹³. While SAM has demonstrated remarkable zero-shot performance in various segmentation tasks on natural images, its application to weak supervision, especially for cell segmentation, remains unexplored. We present BoxCell, a SAM-based method for generating segmentation masks using bounding boxes as prompts. BoxCell uses SAM in two ways – at train and test times. At train time, SAM, when prompted with gold bounding box annotations, generates pseudo-masks for training images. These (image, pseudo-mask) pairs supervise the training of a standalone image segmentation model, such as CaraNet¹⁴. At test time, BoxCell generates two segmentation masks. (1) First mask is generated by this standalone segmenter model. (2) An object detector like YOLO¹⁵, trained with box supervision, predicts bounding boxes on a test image, which are given as prompt to SAM to generate the second mask. Recognizing complementary strengths, with one excelling in localization and the other in learning cell shapes, BoxCell reconciles these strengths using a novel integer programming formulation, with intensity and spatial constraints. Our experiments on three cell segmentation datasets (CoNSep¹⁶, MoNuSeg¹⁷, and TNBC¹⁸) demonstrate BoxCell's significant gains over state-of-the-art weakly supervised segmentation systems, achieving 6-10 point better Dice scores.

¹Yardi School of Artificial Intelligence, India Institute Of Technology, New Delhi 110016, India. ²Department of Computer Science and Engineering, India Institute Of Technology, New Delhi 110016, India. ³Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru 560012, India. ✉email: aiz218615@scai.iitd.ac.in

Related work

Segment anything in image segmentation

Several studies have examined the zero-shot performance of SAM in both natural and medical image segmentation, showing strong results in common scenes but limited performance in complex or low-contrast settings, and on small or irregular objects^{19,20}. Its robustness to image corruptions has also been explored, making it applicable in real-world scenarios²¹. In medical imaging, SAM has been applied to tasks like liver tumor and brain MRI segmentation^{22,23}. However, in whole slide images (WSIs), SAM performs well on large structures but struggles with small, densely packed cells²⁴. Recent efforts like Segment Any Cell²⁵, CellSAM²⁶, MedSAM²⁷ and μ -SAM²⁸ improve performance by fine-tuning SAM or guiding it with object detectors and box prompts. A common observation across these works is that SAM benefits significantly from prompt-based supervision, particularly with bounding boxes, in challenging medical segmentation tasks.

Weakly supervised image segmentation (WSIS)

Many studies address weak supervision by scribble²⁹, class / attention guided^{30,31,31,32} or bounding box supervision for natural image segmentation. Early methods rely on the multi-instance learning (MIL). They assume that bounding boxes are tight^{8,33}, so a line connecting two opposite edges must contain at least one positive pixel. More recent approaches like BoxInst⁹, BoxTeacher⁷ use box based mask alignment, and BoxSnake¹¹ uses polygon based instance segmentation. Despite the strides made in WSIS for natural images, its progress in cell segmentation remains relatively limited³⁴. This limitation can be attributed to challenges like ambiguous boundaries and low contrast variations between foreground and background³⁵.

Ensembling segmentation masks

To ensemble multiple segmentation masks, a classic approach involves outputting the average foreground probability for each pixel³⁶. Another method entails creating an ensemble with low precision and high recall, defined by the model diversity metric³⁶. Alternatively, EmergeNet³⁷ introduces a weighted average of all masks, with weights determined by their performance on the validation set. En-Seg³⁸ use multiple masks to produce an average segmentation masks. We conduct a comparative analysis of BoxCell against these methods (wherever possible), demonstrating BoxCell's superior performance.

Segmentation mask refinement

Various studies propose post-processing methods to enhance segmentation masks, including GrabCut³⁹ and conditional random fields (CRF)^{40,41}. A popular extension of these works is DenseCRF⁴², which uses a fully connected CRF that considers all pairs of pixels in an image. While this approach may be effective on some datasets, DenseCRF typically assumes that all images exhibit consistent and strong contrast between the foreground and background regions, which is often not the case in histopathology images. We conduct the comparative analysis of the BoxCell with DenseCRF and demonstrates BoxCell's superior performance.

BoxCell: Our proposed method

Weakly supervised image segmentation (WSIS) takes in training dataset $D = \{X_k^T, B_k^T\}_{k=1}^D$, where X_k^T is a training image, and B_k^T (in our setting) represents bounding box annotations for the target class. Its goal is to train a model, which, given a test image X , predicts a foreground (cell) segmentation mask M . The proposed method, *BoxCell*, consists of two components: the *Inference Time Detector* (ITD) and *Inference Time Segmenter* (ITS). Both components leverage SAM, a prompt-based general-purpose segmentation model, operating at both training and test stages (see Fig. 1).

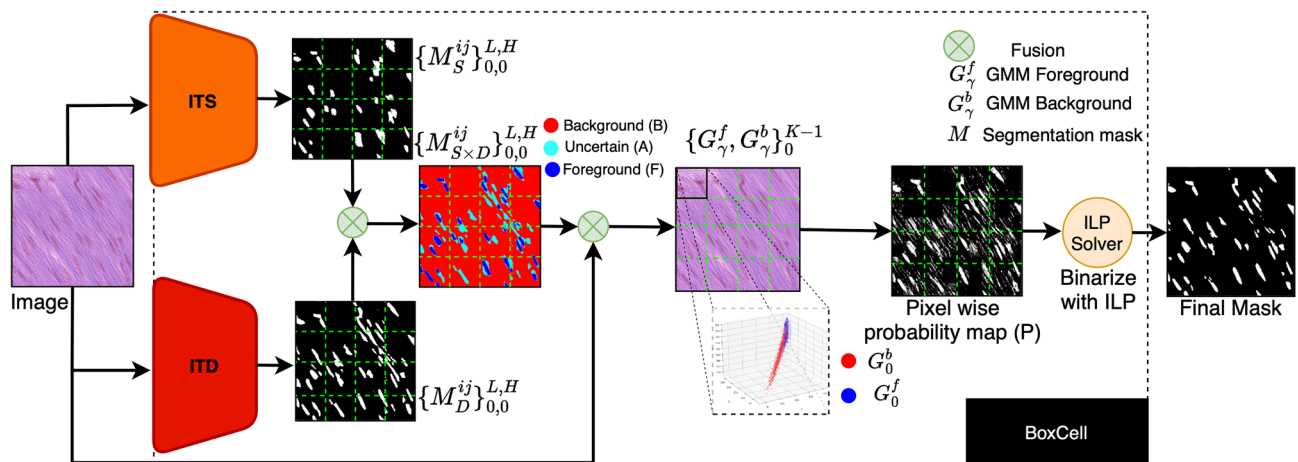


Fig. 1. Inference pipeline for BoxCell, which produces masks M_D and M_S using an ITD and ITS. These masks are split into a $K \times K$ grid, and GMMs are trained to estimate probability map (P). ILP solver refines P based on intensity and spatial constraints. This figure was created using draw.io.

ITD and ITS operate on test image X and independently generate segmentation masks, M_D and M_S . We find that the two masks possess complementary strengths, and better results can be achieved by merging the two. BoxCell achieves this via a novel Integer Linear Programming (ILP) formulation. The ILP outputs a final mask M by balancing the probability of pixel classification into classes (foreground and background), along with the goal that similar intensities at neighboring pixels should be assigned the same class.

Generating segmentation masks

Inference time detector (ITD)

ITD (see Fig. 2), trains an object detector $D(\theta)$ such as Yolov8¹⁵ using images X_k^T and the set of gold bounding boxes B_k^T . The object detector is trained with objectness, classification, and localization losses. Objectness loss (L_{obj}) is the confidence score indicating whether the box contains an object or not. Classification loss (L_{cls}) is computed as the binary cross-entropy between the predicted class and ground truth class. Localization loss (L_{loc}) is the error in predicted bounding box coordinates as compared to ground truth bounding box coordinates. The total detection loss is the sum of all three losses, given as $L_{det} = L_{obj} + L_{cls} + L_{loc}$. This object detector $D(\theta)$ predicts a set of bounding boxes \hat{B} for a test image X . Each box $b \in \hat{B}$ is used as a prompt to SAM to generate a segmentation mask within that box. All box-level masks are combined to generate the image-level segmentation mask M_D .

Inference time segmentor (ITS)

During training, SAM generates masks M_k^T for the training images (X_k^T) using ground truth bounding boxes B_k^T as prompts. Despite SAM's errors, these masks serve as pseudo-masks for training a standalone segmentation model $Sg(\phi)$ like CaraNet¹⁴ – it is trained using a sum of Dice loss and BCE: $L_{seg} = L_{bce} + L_{dice}$ (see Fig. 2). Binary cross-entropy (L_{bce}) improves the pixel-level classification of the segmentation mask, and Dice loss (L_{dice}) guides the intersection of the prediction with the ground truth masks, thereby improving the localization of the predicted masks. At test time, $Sg(\phi)$ runs on X to directly generate an image-level segmentation mask, M_S .

Integer programming for reconciling segmentation masks

We find that M_D excels in localization, whereas M_S is better at shapes; BoxCell merges the two for better performance. We make two key observations. First, for histopathological images, the intensity values within pixels of one class (foreground or background) vary significantly, due to variations in tissue structure and amount of staining from one part of image to another. So, any intensity distribution learned over the *whole* image is likely to be noisy, but could be meaningful if learned over a small patch of the image. Second, there still exists perceptible contrast between pixel intensities in the vicinity of the boundary of a segmentation mask. Following these observations, BoxCell learns Gaussian Mixture Models (GMMs) to model *patch-level* intensity distributions for foreground and background. It then casts an ILP that maximizes the GMM prediction probability along with a soft constraint that neighboring pixels are assigned different classes only if their intensity difference is high. To do so, BoxCell divides each pixel (i, j) for a test image X into three sets: F , B and A . Here, F (and B) is the set where masks M_D and M_S agree on the pixel to be in foreground (resp., background); and A is where they disagree, i.e., $A = \{(i, j) \mid M_D(i, j) \neq M_S(i, j)\}$. BoxCell accepts the pixel labels for F and B for final mask M and only attempts to reassign labels in A .

Learning GMMs

BoxCell splits the image X of size $L \times H$ into K^2 mutually exclusive and collectively exhaustive patches of size $L/K \times H/K$ each. For each patch γ , it learns two GMMs, one for foreground pixel intensities, and one for background. It uses pixels in $F \cup B$ to learn these GMMs and ignores ambiguous pixels in the patch. More

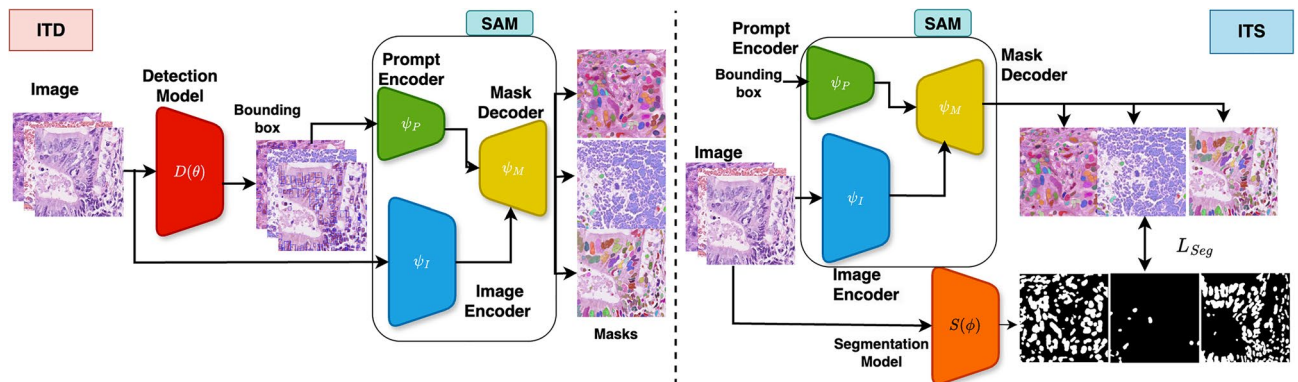


Fig. 2. Workflows with SAM in weak supervision. ITD uses the detection model $D(\theta)$ to predict bounding boxes. The detection model is trained on the training data and is used to predict bounding boxes, which are used as box prompts for SAM during inference. ITS uses segmentation masks predicted by SAM as pseudo ground truth to train $S(\phi)$. We only call $S(\phi)$ during inference.

formally, G_γ^f and G_γ^b are N -component 3-dimensional (RGB) GMMs over the foreground and background pixels in a patch γ . Let $w = \{w_1, w_2 \dots w_N\}$ be the mixture weights such that $\sum w_n = 1$ and $0 \leq w_n \leq 1$. Let $\mu = \{\mu_1, \mu_2 \dots \mu_N\}$, $\mu_n \in \mathbb{R}^3$ and $\Sigma = \{\Sigma_1, \Sigma_2 \dots \Sigma_N\}$, $\Sigma_n = [\sigma^2]_{3 \times 3}$, respectively, denote the means and co-variances. The likelihood density of an RGB pixel $c = (c^1, c^2, c^3)$ belonging to a mixture G is given by $p'(c | G; \mu, \Sigma) = \sum_{n=1}^N w_n N(c, \mu_n, \Sigma_n)$, where N is the Gaussian function.

$$\frac{1}{(2\pi)^{1.5} |\Sigma_n|^{0.5}} \exp\left(-\frac{(c - \mu_n)^T \Sigma_n^{-1} (c - \mu_n)}{2}\right) \tag{1}$$

Since each pixel can either belong to foreground or background, we normalise probabilities as

$$p(c | G_\gamma^f) = \frac{p'(c | G_\gamma^f)}{p'(c | G_\gamma^f) + p'(c | G_\gamma^b)} \tag{2}$$

Here, $p(c | G_\gamma^f)$ is the probability of pixel c being in the foreground, and $p(c | G_\gamma^b) = 1 - p(c | G_\gamma^f)$ of it being in the background. Note that these probabilities are solely based on pixel intensities and do not incorporate any spatial information. BoxCell merges $p(c | G_\gamma^f)$ for all the patches γ to create a complete probability distribution over the entire image – we denote it as $P(c)$ for RGB pixel c .

Integer linear programming

For a pixel (i, j) , lets its color information (RGB) be denoted as c_{ij} (a 3-tuple). The ILP first defines a binary variable x_{ij} (for ambiguous pixels), which is 1, iff the pixel is assigned the foreground label. It defines a part of the objective function, O_{idf} , where *idf* stands for Intensity Distribution Factor:

$$O_{idf} = \sum_{(i,j) \in A} x_{ij} P(c_{ij}) + (1 - x_{ij})(1 - P(c_{ij})) \tag{3}$$

To ensure well-formedness of cells, ILP imposes that neighboring pixels that are assigned different labels must differ in their intensities. For this, it defines binary edge variables e_{ij0} and e_{ij1} , which encode the edges between pixels (i, j) and $(i + 1, j)$, and between (i, j) and $(i, j + 1)$, respectively. The edge variables are assigned 0 if both pixels on the edge belong to the same class, and 1 otherwise. This is encoded in constraints as $e_{ij0} = |x_{ij} - x_{(i+1)j}|$ and $e_{ij1} = |x_{ij} - x_{i(j+1)}|$. If two neighboring pixels get different labels, the objective function gets penalized based on their intensity differences:

$$O_{scf} = \sum_{i=1}^{L-1} \sum_{j=1}^H e_{ij0} S_{ij0} + \sum_{i=1}^L \sum_{j=1}^{H-1} e_{ij1} S_{ij1} \tag{4}$$

We name this part of objective as O_{scf} , where *scf* stands for Spatially Constraining Factor. Here, S_{ij0} and S_{ij1} are a function of intensity differences, for which we employ the color similarity metric⁹, with θ as a hyperparameter:

$$S_{ij0} = \exp\left(\frac{-\|c_{ij} - c_{(i+1)j}\|_2}{\theta}\right), \tag{5}$$

$$S_{ij1} = \exp\left(\frac{-\|c_{ij} - c_{i(j+1)}\|_2}{\theta}\right). \tag{6}$$

Overall, the complete ILP formulation is as follows, with x_{ij} values computing the final segmentation mask labels in M for pixels in set A :

$$\begin{aligned} & \underset{x_{ij}, e_{ij0}, e_{ij1}}{\text{maximize}} \quad O_{idf} - \lambda O_{scf} \\ & \text{subject to} \\ & e_{ij0} = |x_{ij} - x_{(i+1)j}|, \\ & e_{ij1} = |x_{ij} - x_{i(j+1)}|, \\ & x_{ij}, e_{ij0}, e_{ij1} \in \{0, 1\}. \end{aligned} \tag{7}$$

Results

The primary goal of our experiments is to compare BoxCell’s performance with existing box-supervised segmentation methods. Moreover, we wish to understand the qualitative differences between ITD and ITS masks, if any. Finally, we also compare BoxCell’s ILP formulation against existing mask merging and mask refinement approaches.

Datasets

CoNSep: Colorectal nuclear segmentation and phenotypes (CoNSep)¹⁶ is a nuclear segmentation and classification dataset of H&E stained images. Each image is of 1000×1000 dimension. The dataset deals with single cancer, and colorectal adenocarcinoma (CRA) images. It consists of a total of 41 whole slide images (WSI), which have a total of 24,319 annotated cells of 3 classes: inflammatory cells, epithelial cells, and spindle cells. A total of 27 images are used for training and validation, and the rest 14 are used for testing. Further, we split the 1000×1000 images into four sub-images of dimension 500×500. This results in a dataset of 98 train, 10 validation and 56 test images.

MoNuSeg: Multi-organ nuclei segmentation (MoNuSeg)⁴³ is a nuclei segmentation dataset of H&E images representing cell nuclei from 7 different organs like breast, liver, kidney, prostate, bladder, colon and stomach to ensure diversity of nuclear appearances. It consists of a total of 51 images containing 28846 annotated cells. A total of 37 are used for training and validation, and 14 are used as test images. We split the 1000×1000 image into four 500×500 images, resulting in 133 train, 15 validation and 56 test images.

TNBC: It consists of H&E slides of triple negative breast cancer patients taken at 40x magnification¹⁸. The dataset contains 50 images with a total of 4022 annotated cells. Each image is of 512×512 dimension. This dataset proves valuable for evaluating model performance under varying degrees of cellularity. Out of 50 images, we use 34 as training samples, 5 for validation and 11 for the test set.

Since all datasets are originally image segmentation datasets, we converted instance segmentation masks into bounding boxes and used them to train our object detector, keeping the golden masks held out at the time of training. Some of these datasets also assign classes to the cells, but for the sake of our problem, we are only interested in binary segmentation and ignore the class labels.

Evaluation metrics

To evaluate BoxCell's performance on the semantic segmentation task, we use the Dice coefficient and Intersection over Union (IoU) metrics to quantify the similarity between the predicted and ground truth masks.

Instance segmentation

Although BoxCell generates semantic segmentation masks, we convert these into instance segmentation masks to evaluate performance alongside instance segmentation methods. This conversion uses the outputs from both ITD and BoxCell segmentation masks. While generating ITD mask, each cell is assigned a unique instance ID. We then check for overlaps between the ITD and BoxCell masks, and in regions where overlap exists, we assign the corresponding ITD instance ID to those pixels of BoxCell mask. For regions predicted by BoxCell but not by ITD, we assign unique instance IDs to ensure they are included in the instance segmentation mask evaluation. In cases where the BoxCell mask contains connected cells, we apply k-means clustering to separate them into distinct instances. Additionally, we evaluate the instance segmentation quality using three metrics: Aggregated Jaccard Index (AJI)⁴³, Panoptic Quality (PQ)¹⁶, and Boundary F1-score (BF1)⁴⁴. Panoptic Quality (PQ) combines detection quality (DQ) and segmentation quality (SQ), offering a comprehensive measure of both the accuracy of object detection and the precision of segmentation. In our baselines, WSIS methods, except for SPN, directly produce instance segmentation masks. For ensembling and mask refinement methods, we follow the same process used in BoxCell to generate instance segmentation masks.

Implementation details

For the object detection component (ITD), we employ YOLOv8x¹⁵ trained for 300 epochs with early stopping based on validation performance. The initial learning rate is set to 0.01 and gradually reduced by a factor of 0.01 during training using a cosine decay schedule with a 5-epoch warmup. We use a batch size of 32. Data augmentations include random horizontal and vertical flips, rotations ($\pm 90^\circ$), and color jitter (brightness/contrast ± 0.2). During inference, we apply a detector confidence threshold of 0.3 and an NMS IoU threshold of 0.5. All experiments are conducted on NVIDIA RTX-5000 and Tesla A100 GPUs.

For the auxiliary segmentation stage (ITS), we adopt CaraNet¹⁴ and SegFormer⁴⁵, which utilizes a reverse axial attention mechanism effective for small or tiny object segmentation. CaraNet is trained for 200 epochs using the AdamW optimizer with an initial learning rate of 1×10^{-4} , cosine decay scheduling, and a 5-epoch warmup. We set the batch size to 16 and apply early stopping with a patience of 20 epochs based on validation Dice. The same data augmentations as ITD are used, and during inference, small objects with an area smaller than 20 pixels are removed. For comparison, we also evaluate BBTP++⁸, BoxInst⁹, BoxSnake¹¹, BoxTeacher⁷, and SPN³⁴, all trained under identical settings for fairness.

To reconcile the predictions from ITD and ITS, we propose an Integer Linear Programming (ILP)-based mask fusion strategy. We compare our ILP method against several existing mask merging strategies. (i) *AP (Averaging)* performs element-wise averaging of ITD and ITS masks, normalized by 2 and binarized at a threshold of 0.5. (ii) *LP (Low Precision Averaging)*³⁶ applies the same averaging but uses a higher binarization threshold of 0.9 to emphasize high-confidence regions. (iii) *ENet*⁴⁶ computes a weighted sum of ITD and ITS masks, where the weights are tuned on the validation set and remain balanced across datasets (e.g., CoNSep: ITD=0.522 vs. ITS=0.477; MoNuSeg: ITD=0.523 vs. ITS=0.476; TNBC: ITD=0.493 vs. ITS=0.507). (iv) *ILP (ours)* formulates the mask reconciliation as an integer linear programming problem, achieving consistent improvements across datasets, as shown in Table 5.

Following mask fusion, we refine the results using DenseCRF post-processing with Pairwise Gaussian (size 3×3) and Pairwise Bilateral (size 5×5) kernels for 10 inference iterations. For BoxCell-ILP solver, each image is partitioned into a $K \times K$ grid for RGB-GMM modeling, where we use $K = 5$ in all experiments. Hyperparameters were tuned via grid search using Gurobi⁴⁷, with $\lambda \in \{0.5, 1, 2, 5\}$ (pairwise smoothness weight), $\theta \in \{1, 5, 10, 25, 30\}$ (color similarity scaling), and the number of GMM components $\in \{2, 3, 4, 5, 6\}$.

The best results were obtained with $\lambda = 2$, $\theta = 25$, and 2 GMM components. For an image of size 500×500 , the complete processing time is approximately 67–71 seconds on an Intel Xeon processor with 32 CPU cores.

Comparison of weak supervised approaches

Semantic segmentation

Table 1 compares all models in the weakly supervised image segmentation setting. BoxCell achieves substantial 6–10 point Dice improvements compared to the strongest non-SAM competitors across datasets, demonstrating the significant merit of our approach. Even when competing against SAM-based methods, BoxCell maintains a consistent 6–7 point Dice advantage. We believe that this is because the heuristics imposed by weak supervision losses are insufficient to guide SAM. For instance, BoxInst employs intensity-dependent losses in local neighborhoods that remain static across image patches, ignoring underlying image distribution variations. Similarly, SAM-BBTP++ lacks intensity-based criteria for mask integrity and fails to penalize contradictory predictions in neighborhoods. In preliminary experiments, we also made other attempts to use weak supervision losses for training SAM and found that they generally confuse SAM (because of catastrophic forgetting). Figure 3 illustrates sample predictions from each dataset, where BoxCell demonstrates superior accuracy in capturing cell boundaries and shapes.

We also compare with mask refinement method like DenseCRF. DenseCRF uses a unary classifier to learn long-term dependencies. However, we observed that in cases where the background varies throughout the image and the contrast between the foreground and background is minimal, such as in histopathology images, DenseCRF produces suboptimal performance. Therefore, learning local features proves to be more beneficial as done in BoxCell. BoxCell achieves, 1–4 pt dice gain as compared to DenseCRF. Closest to our work, ENSeg-ILP focuses on averaging segmentation results, whereas BoxCell aims to maximize accuracy by reconciling two complementary predictions—ITD and ITS—resulting in notable Dice score improvements (ITD: 82 → 85, ITS: 80 → 85). In addition, BoxCell introduces spatial constraints based on pixel color, contributing an additional 1.1–1.5 Dice points compared to using intensity constraints alone, as employed by ENSeg-ILP (see Table 7). Although the code for ENSeg-ILP is not publicly available for a direct comparison, BoxCell's enhancements highlight its clear advantage.

As described in the dataset details, each ConSeP and MoNuSeg WSI (1000×1000) was divided into four 500×500 sub-images for evaluation, with per-WSI analysis showing negligible differences (ConSeP: 81.39 vs. 81.41; MoNuSeg: 81.74 vs. 81.73). TNBC was evaluated at full resolution. We performed paired t-test analysis to evaluate the statistical significance of BoxCell-ILP against existing bounding box supervised approaches. Statistical analysis confirms that BoxCell achieves significant improvements ($p < 10^{-4}$), with 95% confidence intervals for mean Dice gains of [5.7–5.8], [8.1–8.2], and [5.2–5.3] on ConSeP, MoNuSeg, and TNBC, respectively, validating its robustness across datasets.

Lastly, we compare BoxCell with several SAM variants (Table 2), including the recently introduced SAM2⁴⁹, designed for video segmentation, and MedSAM⁵⁰, trained specifically on medical images. We also include μ -SAM, a SAM variant optimized for microscopy images. As shown in Table 2, BoxCell consistently improves ITD-ITS performance across all SAM backbones. For instance, on the ConSeP dataset, BoxCell improves performance from 79.97/79.80 to 81.36 with SAM2, and on MoNuSeg, from 80.35/79.80 to 81.83. On TNBC, it increases from 82.24/80.54 to 84.46. Similarly, with MedSAM, the Dice score rises from 71.00 to 74.91, though its overall performance remains lower than SAM and SAM2 for cell segmentation. Quantitatively, BoxCell achieves improvements of approximately +6.5 Dice on ConSeP, +4.5 Dice on MoNuSeg, and +2.3 Dice on TNBC over μ -SAM and MedSAM. These results highlight that beyond simply adapting SAM to biomedical images, BoxCell's integer linear programming-based reconciliation of inference-time detection and segmentation yields consistent 2–3 Dice point gains across backbones and substantial absolute improvements overall.

Model	CoNSeP		MoNuSeg		TNBC	
	Dice	IoU	Dice	IoU	Dice	IoU
SPN ³⁴	64.44	60.54	50.48	35.29	68.88	53.68
BoxSnake ¹¹	71.02	55.40	74.80	60.58	69.71	56.32
BoxTeacher ⁷	63.42	43.28	69.64	54.81	74.35	59.32
BBTP ⁴⁸	70.89	55.74	72.46	60.64	68.53	58.14
BBTP++ ⁸	75.58	60.99	72.84	61.22	70.37	58.60
BoxInst ⁹	64.50	48.11	66.7	52.16	74.75	59.89
SAM-BBTP ⁴⁸	68.62	52.78	69.99	53.99	78.78	65.04
SAM-BBTP++ ⁸	74.64	60.11	73.58	60.64	79.64	67.28
SAM-BoxInst ⁹	74.08	59.43	73.05	61.22	79.43	67.43
ITD (Ours)	80.00	66.99	79.87	66.68	82.86	70.71
ITS (Ours)	79.86	65.54	79.38	65.97	80.66	66.18
BoxCell - DenseCRF (Ours) ⁴²	77.82	63.50	80.41	67.40	81.19	68.50
BoxCell - ILP (Ours)	81.39	68.82	81.74	69.25	85.01	74.06

Table 1. Comparison of Bounding Box Supervised Methods.

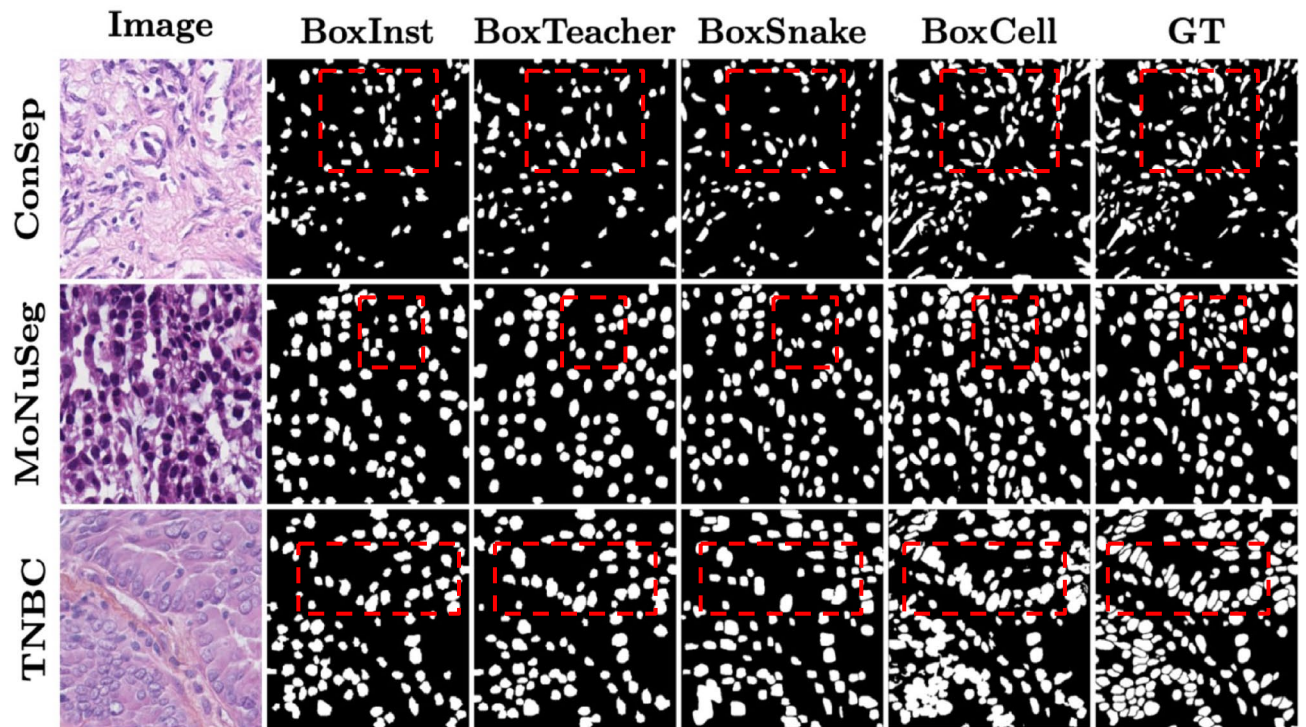


Fig. 3. Qualitative analysis of segmentation masks. Column 1 is the original image, Columns 2-5 show cropped masks (shown in red box) generated from three comparison models and BoxCell. Last column is the ground truth. BoxCell exhibits best results, providing more accurate masks with better cell boundary and shape.

Method	CoNSep		MoNuSeg		TNBC	
	Dice	IoU	Dice	IoU	Dice	IoU
with SAM ¹³	81.39	68.82	81.74	69.25	85.01	74.06
with SAM2 ⁴⁹	81.36	68.80	81.83	69.35	84.46	73.61
with MedSAM ⁵¹	74.91	60.83	77.60	64.79	82.76	70.72
with μ -SAM ⁵²	74.94	60.92	77.22	64.35	81.07	68.16

Table 2. Comparison of BoxCell with various SAM backbones.

Method	ConSep		MoNuSeg		TNBC	
	PQ	AJI	PQ	AJI	PQ	AJI
Connected Components	28.75	22.65	53.90	52.55	50.70	40.78
Watershed	30.16	27.55	57.50	58.13	53.70	49.60
BoxCell (ours)	42.75	44.72	62.51	63.75	63.90	66.26

Table 3. Comparison of detector-agnostic instance conversion methods with BoxCell.

Instance segmentation

Table 3 compares BoxCell with detector-agnostic instance conversion methods, including connected components and watershed segmentation. In our current pipeline, the number of clusters (K) for splitting touching regions is determined by the number of overlapping bounding boxes from the detector, with centroids initialized at corresponding box centers. This introduces a degree of dependence on the detector. To assess its impact, we implemented detector-agnostic alternatives (connected-component and watershed) and found that, while they achieve reasonable performance, BoxCell's detector-based instance conversion consistently yields substantially higher PQ and AJI scores across all datasets.

Furthermore, Table 4 compares BoxCell with other baselines on the instance segmentation task. Although BoxCell is primarily designed for semantic segmentation, we extend it to instance segmentation and observe consistent improvements in PQ, AJI, and Boundary F1. BoxCell achieves 2–7 point PQ gains over weakly

Model	CoNSep			MoNuSeg			TNBC		
	PQ	AJI	BF1	PQ	AJI	BF1	PQ	AJI	BF1
BoxTeacher ⁷	34.66	23.95	51.09	42.65	42.05	49.92	47.24	46.67	52.33
BoxSnake ¹¹	33.7	34.87	51.60	56.43	56.51	51.18	53.71	52.64	53.02
BoxInst ⁹	41.63	28.5	52.43	55.01	55.5	53.90	56.01	56.79	51.78
SAM-BBTP ⁴⁸	35.51	27.39	51.12	48.32	48.83	50.6	51.48	52.09	52.5
SAM-BBTP++ ⁸	41.97	29.65	52.38	56.07	56.91	53.92	56.13	56.87	52.56
AP (ITS, ITD) ³⁶	39.90	28.41	50.18	54.17	54.98	52.21	55.32	55.12	51.73
LP (ITS, ITD) ³⁶	40.01	28.56	50.66	54.91	52.50	52.12	54.42	54.86	51.23
ENet (ITS, ITD) ⁴⁶	39.84	28.35	50.02	54.12	54.94	52.17	55.29	55.08	51.67
BoxCell-DenseCRF (Ours) ⁴²	35.03	40.66	53.87	59.01	60.51	53.62	50.61	58.02	52.92
BoxCell - ILP (Ours)	42.75	44.72	54.07	62.51	63.75	54.03	63.9	66.26	53.85

Table 4. Performance Comparison for Instance Segmentation.

Model	Time (sec)	ConSep		MoNuSeg		TNBC	
		Dice	IoU	Dice	IoU	Dice	IoU
AP (ITS, ITD)	1.20	80.55	67.69	80.04	66.86	82.92	70.97
LP (ITS, ITD)	1.26	80.59	66.70	79.57	66.37	78.48	70.07
ENet (ITS, ITD)	2.63	80.23	67.40	80.07	66.59	82.26	71.07
BoxCell - Sparse	15.00	80.15	67.10	80.21	67.02	85.02	74.07
BoxCell - Gurobi	69.02	81.39	68.82	81.74	69.25	85.01	74.06

Table 5. Comparison of Mask Merging Methods.

Method	ConSep		MoNuSeg		TNBC	
	Dice	IoU	Dice	IoU	Dice	IoU
CaraNet ¹⁴	81.39	68.82	81.74	69.25	85.01	74.06
SegFormer ⁴⁵	78.83	65.34	80.86	68.01	85.65	75.01

Table 6. Comparison of Segmentors for ITS.

supervised instance segmentation methods, particularly excelling in densely packed or low-contrast regions where competing approaches often struggle.

Comparison with mask merging methods

Table 5 reports experiments where masks M_D and M_S from ITD and ITS are merged. We compare several mask merging strategies from the literature and our proposed ILP-based approach. AP (Averaging) performs element-wise averaging of ITD and ITS masks, followed by normalization by 2 and binarization at a threshold of 0.5. LP (Low Precision Averaging)³⁶ uses the same averaging but applies a higher threshold of 0.9 to emphasize high-confidence regions. ENet⁴⁶ computes a weighted sum of ITD and ITS masks, where the weights are tuned on the validation set. The weights are nearly balanced across datasets, indicating stability (e.g., CoNSep: ITD=0.522 vs. ITS=0.477; MoNuSeg: ITD=0.523 vs. ITS=0.476; TNBC: ITD=0.493 vs. ITS=0.507). Finally, ILP (ours) reconciles ITD and ITS masks through integer linear programming. Across all datasets, BoxCell's ILP achieves consistent improvements over them.

Segmentation backbones

We utilized CaraNet as the backbone for training the Inference Time Segmentor (ITS) model. Additionally, we evaluated a transformer-based approach, SegFormer, for training ITS, as shown in Table 6. The table compares the performance of BoxCell with CaraNet as the ITS model (BoxCell-CaraNet) and BoxCell with SegFormer as the ITS model (BoxCell-SegFormer). BoxCell-SegFormer demonstrated performance comparable to BoxCell-CaraNet. However, due to the limited dataset size, larger transformer-based models like SegFormer may not yield significant benefits, as evidenced in the results.

Discussion: ablation and error analysis

Ablation study

We conduct an ablation study to evaluate the individual contributions of various components to the overall model performance, as summarized in Table 7. Specifically, we examine the impact of the Intensity Distribution Factor (IDF) and the Spatial Constraining Factor (SCF), alongside the ITD and ITS predictions. Incorporating

				CoNSep		MoNuSeg		TNBC	
ITD	ITS	IDF	SCF	Dice	IoU	Dice	IoU	Dice	IoU
✓				80.00	66.99	79.87	66.68	82.86	70.71
	✓			79.86	65.54	79.38	65.97	80.66	66.18
✓	✓	✓		81.03	68.06	81.20	68.58	84.53	73.33
✓	✓		✓	80.36	67.24	80.47	67.70	82.98	70.87
✓	✓	✓	✓	81.39	68.82	81.74	69.25	85.01	74.06

Table 7. Ablation Table.

Solver	CoNSep		MoNuSeg		TNBC		Note
	Time/image (s)	Dice	Time/image (s)	Dice	Time/image (s)	Dice	
CBC	204.25	78.99	219.00	75.74	190.00	81.58	open-source
α -expansion	46.20	72.74	42.77	79.13	46.40	71.16	approx.
Sparse graph	15.00	80.15	12.75	80.21	15.07	<u>85.02</u>	fast approx.
OR-Tools	87.30	<u>81.34</u>	80.94	<u>80.60</u>	85.59	85.03	open-source
Gurobi	71.06	81.39	67.27	81.74	68.67	85.01	commercial

Table 8. Comparison of solver performance across datasets. Gurobi achieves the best Dice scores consistently, while open-source alternatives (e.g., OR-Tools) provide competitive results.

Threshold	CoNSep (Dice)	MoNuSeg (Dice)	TNBC (Dice)
0.1	74.30	77.90	81.54
0.2	76.30	77.90	81.54
0.3	79.26	77.90	81.27
0.4	73.20	77.89	81.86
0.5	71.20	78.19	82.75
0.6	70.30	78.94	82.08
0.7	71.20	79.72	82.01
0.8	68.90	79.73	80.11
0.9	69.02	79.77	80.12

Table 9. Segmentation Dice scores across datasets at different YOLOv8-m detector thresholds.

IDF into the ITD-ITS framework yields an improvement of 1–2 points, while SCF contributes a gain of 0.1–0.7 points—particularly beneficial in scenarios with high foreground–background contrast. The combination of all four components results in the best overall performance.

Solver alternatives

We evaluated BoxCell with multiple alternatives to Gurobi, including open-source ILP solvers (CBC, OR-Tools), approximate inference (α -expansion), and a sparse graph formulation. Table 8 report average time per image runtime, and Dice scores across datasets. Across datasets, we find that open-source solvers (OR-Tools and CBC) and approximate methods (α -expansion, sparse graph) achieve accuracy close to or within 1–2 Dice of the Gurobi baseline, while offering different trade-offs in runtime. The sparse graph formulation is especially effective, reducing runtime significantly while maintaining competitive Dice. These results demonstrate that BoxCell is not tied to a commercial solver: open-source or approximate alternatives can also be used.

Impact of detector threshold

To analyze the impact of detector threshold, we varied the YOLOv8-m detection score threshold (0.1–0.9) and measured the resulting segmentation dice scores across datasets (see Table 9). Segmentation quality is relatively stable across a wide range of detector thresholds, with only fluctuations in CoNSep at high threshold. Dice peaks at threshold of 0.3 for CoNSep, at 0.7 for MoNuSeg, and at 0.5 for TNBC. This indicates that BoxCell is not overly sensitive to the detector operating point: once bounding boxes are reasonably accurate, the ILP-based reconciliation mitigates detection errors.

Effect of grid size K

In our method, the grid size $K \times K$ is used to partition each image for local RGB-GMM modeling. In all our experiments we used $K = 5$. To assess sensitivity, we varied K in the range $\{3, 5, 10, 15, 20\}$. Table 10 provides

<i>K</i>	ConSep (Dice)	MoNuSeg (Dice)	TNBC (Dice)
3	81.38	81.07	85.30
5	81.39	81.74	85.01
10	81.36	80.34	84.62
15	81.38	80.59	84.23
20	81.23	80.76	84.52

Table 10. Sensitivity of BoxCell performance (Dice score) to grid size *K*.

Method	Train → Test	Dice
SAM-BBTP++	C → M	35.24
BoxCell (ours)	C → M	57.87
SAM-BBTP++	C → T	36.59
BoxCell (ours)	C → T	62.58
SAM-BBTP++	M → C	32.13
BoxCell (ours)	M → C	57.11
SAM-BBTP++	M → T	35.75
BoxCell (ours)	M → T	50.58
SAM-BBTP++	T → C	26.42
BoxCell (ours)	T → C	49.67
SAM-BBTP++	T → M	27.71
BoxCell (ours)	T → M	54.78

Table 11. Cross-domain performance (train → test).

dice score for varied value of *K* across the three datasets. As shown in Table 10, the performance remains stable across a wide range of *K*, with only marginal differences (within ± 1 Dice). This indicates that the method is not highly sensitive to the choice of grid size.

Cross-domain generalization and stain robustness

We evaluate BoxCell against the strongest baseline in cross-domain settings (Table 11), reporting train–test performance across datasets (C: CoNSeP, M: MoNuSeg, T: TNBC). BoxCell consistently outperforms SAM-BBTP++, demonstrating superior domain generalization. While performance decreases compared to within-domain results (81.39, 81.74, and 85.01 for CoNSeP, MoNuSeg, and TNBC, respectively), BoxCell still surpasses the closest baseline by effectively reconciling ITD and ITS. Training on one dataset and testing on another yields substantial Dice score gains (e.g., C → M: +22.63, C → T: +26.00, M → C: +24.98), further confirming better robustness to domain shifts.

Stain Robustness: To evaluate stain robustness, we generated stain-varied test sets for each dataset. Figure 4 illustrates the original samples (row 1) and their stain-varied counterparts (row 2). Table 12 shows that BoxCell demonstrates markedly greater robustness to stain variation compared to the baseline SAM-BBTP++. This indicates that BoxCell’s reconciliation of ITD and ITS effectively handles staining variability, making it a more reliable approach for real-world scenarios.

Compute capacity

Table 13 provide parameters and time comparison for BoxCell and closest baseline. Both models have comparable parameter counts, ensuring similar capacity. The longer runtime of BoxCell arises mainly from the ILP optimization step, not from model size. While BoxCell is slower due to the ILP solver, it achieves substantially higher Dice scores across datasets. Another variant of BoxCell with a sparse solver (see row 2 in Table 13), offers faster computation (15sec/image) while maintaining strong segmentation performance.

Error analysis

On analyzing the masks qualitatively, we find that ITS benefits from a global perspective, leveraging the full image view for producing masks. As a result, ITS has a better shape understanding. Conversely, in ITD, SAM’s inference is localized to the region defined by the box prompt. In scenarios where cell boundaries are ambiguous, ITS tends to overshoot the segmentation. In such cases, ITD often performs better, due to the localization provided by the box prompt. Another notable difference is the compounding of errors due to the pipelined nature of ITD. ITD is not able to recover from the mistakes of the object detector; if the box is a false positive, SAM generally outputs a false mask, and if a cell is missed by the detector, SAM can never output a mask for it. This is not an issue for ITS as it does not have a multi-step pipeline. With the ILP, BoxCell mitigates issues of both component models. See Fig. 5 for an illustration. BoxCell demonstrates a reduced dependence on the size of bounding boxes – it can make mask predictions outside the bounding boxes (A, Row 1). It can generate

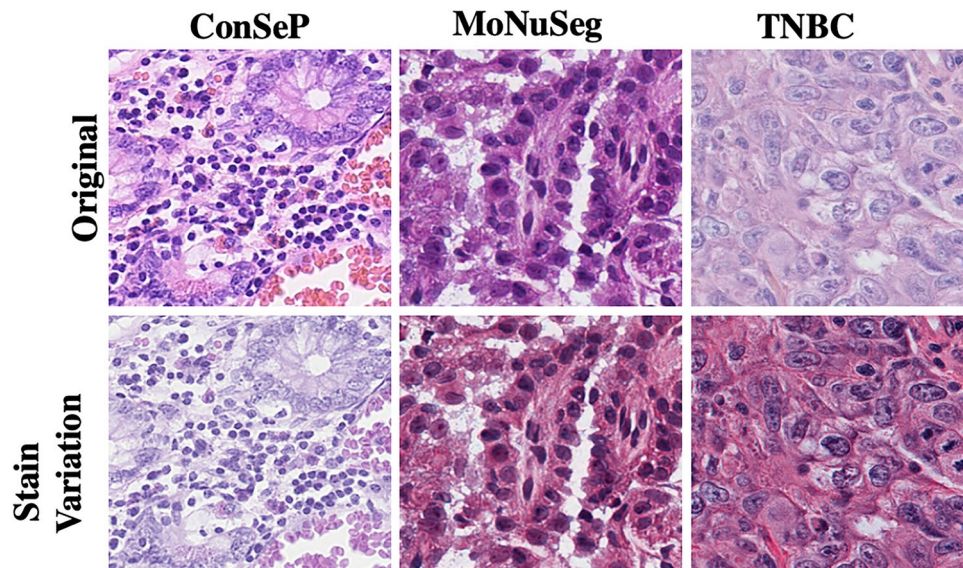


Fig. 4. Original and stain variation images across datasets.

Method		CoNSeP	MoNuSeg	TNBC
SAM-BBTP++	With Stain Variation	63.20	65.59	62.72
SAM-BBTP++	No Stain Variation	74.64	73.58	79.64
BoxCell	With Stain Variation	73.80	78.20	80.75
BoxCell	No Stain Variation	81.39	81.74	85.01

Table 12. Performance comparison across datasets under stain variation and no stain variation conditions.

Model	Parameters (M)	Time (sec/image)	ConSeP	MoNuSeg	TNBC
SAM-BBTP	15.2	0.2	74.64	73.58	79.64
BoxCell - Sparse	15.7	15	80.15	80.21	85.02
BoxCell - Gurobi	15.7	69.02	81.39	81.74	85.01

Table 13. Compute Capacity comparison between BoxCell and SAM-BBTP.

segmentation masks even when the detection model predicts no bounding box, thus mitigating ITD's false negatives problem (B, Row 1). Finally, it yields qualitatively crisper boundaries, a characteristic only observed in BoxCell across all models (C, Row 1).

Common failure modes for BoxCell include images with low contrast between the intensity of foreground and background pixels (A in Fig 5, Row2). The human ability to detect such cells relies on the shape of these faintly different regions of intensities. The method fails to detect and mitigate false positives in such cases (B in Fig 5, Row2). It tends to overlap instances of different cells due to no direct box supervision. It also tends to segment regions with intensity variation despite them not being cells. The detector's performance bottlenecks the performance of all detector-based models because of the high dependence on the box prompts. For the best-performing detection model, we still retain a lot of false positives that generate segmentation masks even when no actual cell is present. False negatives do not get segmented if the detector fails to detect them. Although SAM-ILP tries to mitigate these effects, they are still persistent in some cases.

Effect on annotation efficiency

We observe that BoxCell substantially reduces the annotation time required by pathologists. To quantify this, we conducted an annotation-efficiency study with two pathologists, each annotating 25 randomly selected cells (total $n = 50$). Manual polygon annotation, performed using LabelMe, required an average of 17.82 seconds per cell (95% CI: 17.39–18.25, SD 1.5), including identifying cell boundaries, drawing polygons, and assigning class labels. In contrast, BoxCell requires only bounding box generation, taking 5.73 seconds per cell, followed by 4.20 seconds for verifying and refining the generated instance masks, resulting in a total of 9.92 seconds per cell (95% CI: 8.93–10.91, SD 3.54). This represents a 7.9-second reduction, corresponding to a 44.4% improvement in annotation efficiency.

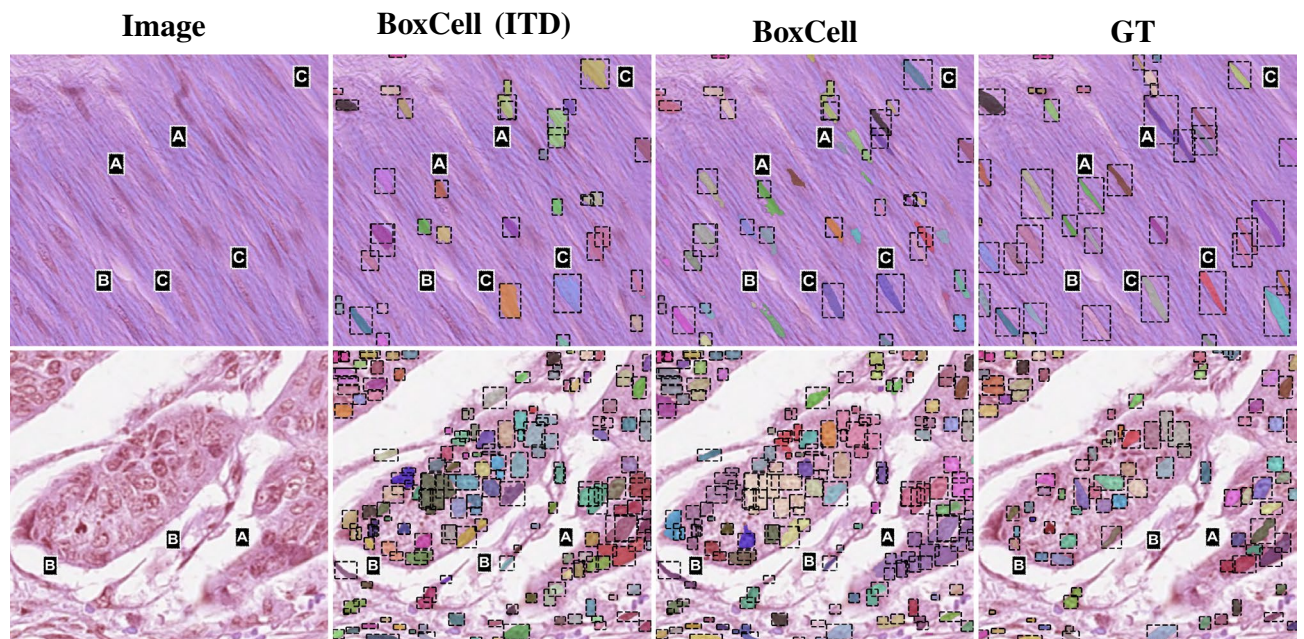


Fig. 5. BoxCell with only ITD does not predict foreground outside box-prompts. BoxCell can do so, reducing the number of false negatives and improving the mask quality for wrongly sized boxes (A and B). BoxCell produces finer segmentation masks (C). BoxCell performs less effectively for images with low contrast in *f/g* and *b/g* (A row 2) where its capability to mitigate false positives is limited (B in row 2).

Conclusion

We present BoxCell, the first approach to use SAM-based segmentation over histopathological images when only bounding box supervision is available. It computes two segmentation masks using SAM at train and test times; and reconciles them via a novel ILP that balances pixel likelihood and neighborhood objectives. Our experiments over three benchmark datasets show that our proposed approaches consistently beat the current weak supervision methods by up to 10 dice pts. Additionally, we compare with mask ensembling and refinement method and show the effectiveness of BoxCell. Our work opens up new possibilities for leveraging SAM in weak-supervision settings and using constrained optimization strategies to post-process segmentation masks.

Data availability

The datasets used in this study are publicly available. The code for BoxCell implementation is available at <https://github.com/dair-iitd/BoxCell>. All experimental data and weights are available upon reasonable request to the corresponding author.

Received: 23 May 2025; Accepted: 30 October 2025

Published online: 22 November 2025

References

- Lu, C. et al. Nuclear shape and orientation features from h & e images predict survival in early-stage estrogen receptor-positive breast cancers. *Lab. Invest.* **98**, 1438–1448 (2018).
- Wählby, C., Sintorn, I.-M., Erlandsson, F., Borgefors, G. & Bengtsson, E. Combining intensity, edge and shape information for 2d and 3d segmentation of cell nuclei in tissue sections. *J. Microsc.* **215**, 67–76 (2004).
- Tyagi, A. K. et al. Degpr: Deep guided posterior regularization for multi-class cell detection and counting. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23913–23923 (2023).
- Ahn, J., Cho, S. & Kwak, S. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proc. IEEE/CVF conference on computer vision and pattern recognition*, 2209–2218 (2019).
- Lin, D., Dai, J., Jia, J., He, K. & Sun, J. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proc. IEEE conference on computer vision and pattern recognition*, 3159–3167 (2016).
- Bearman, A., Russakovsky, O., Ferrari, V. & Fei-Fei, L. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, 549–565 (Springer, 2016).
- Cheng, T., Wang, X., Chen, S., Zhang, Q. & Liu, W. Boxteacher: Exploring high-quality pseudo labels for weakly supervised instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3145–3154 (2023).
- Wang, J. & Xia, B. Bounding box tightness prior for weakly supervised image segmentation. In *International conference on medical image computing and computer-assisted intervention*, 526–536 (Springer, 2021).
- Tian, Z., Shen, C., Wang, X. & Chen, H. Boxinst: High-performance instance segmentation with box annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5443–5452 (2021).
- Mahani, G. K. et al. Bounding box based weakly supervised deep convolutional neural network for medical image segmentation using an uncertainty guided and spatially constrained loss. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, 1–5 (IEEE, 2022).

11. Yang, R., Song, L., Ge, Y. & Li, X. Boxsnake: Polygonal instance segmentation with box supervision. *arXiv preprint arXiv:2303.11630* (2023).
12. Liu, Y. et al. Box2seg: Learning semantics of 3d point clouds with box-level supervision. *arXiv preprint arXiv:2201.02963* (2022).
13. Kirillov, A. et al. Segment anything. *arXiv preprint arXiv:2304.02643* (2023).
14. Lou, A., Guan, S., Ko, H. & Loew, M. H. Caranet: Context axial reverse attention network for segmentation of small medical objects. In *Medical Imaging 2022: Image Processing*, vol. 12032, 81–92 (SPIE, 2022).
15. Jocher, G., Chaurasia, A. & Qiu, J. YOLO by Ultralytics (2023).
16. Graham, S. et al. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Med. Image Anal.* **58**, 101563 (2019).
17. Verma, R. et al. Multi-organ nuclei segmentation and classification challenge 2020. *IEEE Trans. Med. Imaging* **39**, 8 (2020).
18. Naylor, P., Laé, M., Reyat, F. & Walter, T. Segmentation of nuclei in histopathology images by deep regression of the distance map. *IEEE Trans. Med. Imaging* **38**, 448–459 (2018).
19. Ji, W. et al. Segment anything is not always perfect: An investigation of sam on different real-world applications (2024).
20. He, C. et al. Weakly-supervised concealed object segmentation with sam-based pseudo labeling and multi-scale feature grouping. *arXiv preprint arXiv:2305.11003* (2023).
21. Qiao, Y. et al. Robustness of sam: Segment anything under corruptions and beyond. *arXiv preprint arXiv:2306.07713* (2023).
22. Hu, C. & Li, X. When sam meets medical images: An investigation of segment anything model (sam) on multi-phase liver tumor segmentation. *arXiv preprint arXiv:2304.08506* (2023).
23. Mohapatra, S., Gosai, A. & Schlaug, G. Sam vs bet: A comparative study for brain extraction and segmentation of magnetic resonance images using deep learning. *arXiv preprint arXiv:2304.047382*, 4 (2023).
24. R. D. et al. Segment anything model (sam) for digital pathology: Assess zero-shot segmentation on whole slide imaging. In *IS & T International Symposium on Electronic Imaging*, vol. 37, COIMG–132 (2025).
25. Na, S., Guo, Y., Jiang, F., Ma, H. & Huang, J. Segment any cell: A sam-based auto-prompting fine-tuning framework for nuclei segmentation. *arXiv preprint arXiv:2401.13220* (2024).
26. Israel, U. et al. A foundation model for cell segmentation. *bioRxiv* 2023–11 (2024).
27. Mazurowski, M. A. et al. Segment anything model for medical image analysis: an experimental study. *Med. Image Anal.* **89**, 102918 (2023).
28. Archit, A. et al. Segment anything for microscopy. *bioRxiv* 2023–08 (2023).
29. Chen, J., Huang, W., Zhang, J., Debattista, K. & Han, J. Addressing inconsistent labeling with cross image matching for scribble-based medical image segmentation. *IEEE Transactions on Image Processing* (2025).
30. Chen, J., Li, W., Li, H. & Zhang, J. Deep class-specific affinity-guided convolutional network for multimodal unpaired image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 187–196 (Springer, 2020).
31. Chen, J. et al. Dynamic contrastive learning guided by class confidence and confusion degree for medical image segmentation. *Pattern Recogn.* **145**, 109881 (2024).
32. Chen, J., Zhang, J., Debattista, K. & Han, J. Semi-supervised unpaired medical image segmentation through task-affinity consistency. *IEEE Trans. Med. Imaging* **42**, 594–605 (2022).
33. Kervadec, H., Dolz, J., Wang, S., Granger, E. & Ayed, I. B. Bounding boxes for weakly supervised segmentation: Global constraints get close to full supervision. In *Medical imaging with deep learning*, 365–381 (PMLR, 2020).
34. Liu, W., He, Q. & He, X. Weakly supervised nuclei segmentation via instance learning. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, 1–5 (IEEE, 2022).
35. Chen, Z., Tian, Z., Zhu, J., Li, C. & Du, S. C-cam: Causal cam for weakly supervised semantic segmentation on medical image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11676–11685 (2022).
36. Ma, T. et al. Ensembling low precision models for binary biomedical image segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 325–334 (2021).
37. Dai, H. et al. Samaug: Point prompt augmentation for segment anything model. *arXiv preprint arXiv:2307.01187* (2023).
38. Alush, A. & Goldberger, J. Ensemble segmentation using efficient integer linear programming. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**, 1966–1977 (2012).
39. Rother, C., Kolmogorov, V. & Blake, A. “Grabcut” interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. (TOG)* **23**, 309–314 (2004).
40. Triggs, B. & Verbeek, J. Scene segmentation with crfs learned from partially labeled images. *Advances in neural information processing systems* **20** (2007).
41. Plath, N., Toussaint, M. & Nakajima, S. Multi-class image segmentation using conditional random fields and global classification. In *Proceedings of the 26th annual international conference on machine learning*, 817–824 (2009).
42. Krähenbühl, P. & Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems* **24** (2011).
43. Kumar, N. et al. A multi-organ nucleus segmentation challenge. *IEEE Trans. Med. Imaging* **39**, 1380–1391 (2019).
44. Csurka, G., Larlus, D., Perronnin, F. & Meylan, F. What is a good evaluation measure for semantic segmentation?. In *Bmvc*, vol. 27, 10–5244 (Bristol, 2013).
45. Xie, E. et al. Segformer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural. Inf. Process. Syst.* **34**, 12077–12090 (2021).
46. Das, A., Das Choudhury, S., Das, A. K., Samal, A. & Awada, T. Emergenet: A novel deep-learning based ensemble segmentation model for emergence timing detection of coleoptile. *Front. Plant Sci.* **14**, 1084778 (2023).
47. Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual (2023).
48. Hsu, C.-C., Hsu, K.-J., Tsai, C.-C., Lin, Y.-Y. & Chuang, Y.-Y. Weakly supervised instance segmentation using the bounding box tightness prior. *Advances in Neural Information Processing Systems* **32** (2019).
49. Ravi, N. et al. Sam 2: Segment anything in images and videos. *arXiv preprint* (2024).
50. Ma, J. et al. Segment anything in medical images. *Nat. Commun.* **15**, 654 (2024).
51. Ma, J. & Wang, B. Segment anything in medical images. *arXiv preprint arXiv:2304.12306* (2023).
52. Archit, A. et al. Segment anything for microscopy. *Nat. Methods* **22**, 579–591 (2025).

Acknowledgements

We thank Indian Institute of Technology, Delhi (IIT-D) HPC facility for computational resources. This work is supported by Jai Gupta chair fellowship by IIT Delhi, Yardi School of AI (ScAI) - IIT-Delhi, and Prime Minister Research Fellowship (PMRF).

Author contributions

Aayush: Writing—original draft, conceptualization, experiments, implementation; Vaibhav: Baseline implementation, conceptualization, design; Prathosh A.P.: Writing—review and editing, conceptualization, design, supervision; Mausam: Writing—review and editing, conceptualization, design, methodology, supervision.

Funding

Project is funded by Indian Council of Medical Research (ICMR), Prime Minister Research Fellowship (PMRF) and Jai Gupta Chair.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.K.T.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2025